TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF PRODUCTION ENGINEERING AND MANAGEMENT

Diploma Thesis

"Reconstitution of the administrative process of printer cartridges procurement by using a Business Process Management System (BPMS)"

by Benioudaki Maria

Thesis Committee

Prof. Nikolaos Matsatsinis (Supervisor)
Prof. Evangelos Grigoroudis
Dr. Nikolaos Spanoudakis

May 2019

ABSTRACT

It is known that the Greek public sector lacks automation for the processes which take place within the public institutions, leading the employees to work inefficiently and to waste time in repetitive, sometimes even unnecessary, procedures. One of the mentioned procedures is the procurement of the compatible and genuine ink cartridges for the Technical University of Crete. The present thesis aims to automate the process of procurement by using Business Process Management.

According to the people in charge of the procurements, the present process is outdated, since the users order their ink cartridges either by sending an e-mail or by making a phone call. This practice is time-consuming, because most of the times one email or phone call is not enough, while many times the users cannot correctly specify their printer's model (or ink cartridge) leading to confusion and mistakes.

For the present thesis, at first, the processes that take place in the procurement process were mapped and represented by using Business Process Model and Notation (BPMN). Afterwards, the designed processes were configured and executed by using an open-source Business Process Management System (BPMS). Furthermore, by using this BPMS a platform (referred as Ink BManager) was created for both the administrators and the users. Within the platform users can log in and choose products from an extensive list of available printers, while the system automatically matches the compatible ink cartridges, whilst in case that a product is not within this list, the platform offers the ability to the users to add it. Also, the platform empowers the administrator to view the ordered products and to approve or not a given order. Since mistakes can occur while placing one order the system enables the administrator to reopen an order so that users can edit their already placed order. For administrator's assistance the platform also generates an aggregate list of all the ordered products, and it enables him/her to estimate the cost of the order, as well as to save the prices given from the supplier. Additionally, the platform gives the ability to the administrator to view the past orders and to keep track of the expenses based on a procurement period, a department or a user.

Finally, in order to make sure that the platform would meet the expectations as a real-world solution, a pilot test was conducted between the members of the Technical University of Crete, and a survey was made in order to receive the users' feedback.

ΠΕΡΙΛΗΨΗ

Κύριο χαρακτηριστικό της ελληνικής πραγματικότητας είναι η ελλιπής αυτοματοποίηση των διαδικασιών του δημοσίου τομέα, που αναγκάζει τους υπαλλήλους να εκτελούν εργασίες με τρόπο χρονοβόρο και μη αποδοτικό. Σκοπός της παρούσας διπλωματικής εργασίας είναι η αυτοματοποίηση του τρόπου διενέργειας διαγωνισμών για την προμήθεια συμβατών και γνήσιων μελανοδοχείων για τους εκτυπωτές όλων των τμημάτων του Πολυτεχνείου Κρήτης με χρήση της μεθόδου Διαχείρισης Επιχειρηματικών Διαδικασιών (BPM - Business Process Management).

Αρχικά η διαδικασία των προμηθειών αναπαραστάθηκε μέσω της Γλώσσας Μοντελοποίησης Επιχειρηματικών Διαδικασιών (BPMN – Business Process Model and Notation) και στην συνέχεια υλοποιήθηκε μέσω ενός Συστήματος Διαχείρισης Επιχειρηματικών Διαδικασιών (BPMS – Business Process Management System). Μέσω του συστήματος BPMS, αναπτύχθηκε μια πλατφόρμα (αναφέρεται ως Ink BManager) που απευθύνεται τόσο στους διαχειριστές (μέλη του Τμήματος Διοικητικής Υπολογιστικής Υποδομής του Πολυτεχνείου Κρήτης) όσο και στους απλούς χρήστες. Το σύστημα λειτουργεί με ηλεκτρονικές εγγραφές και ηλεκτρονικό ταχυδρομείο και αποσκοπεί στην βελτίωση της επικοινωνίας όλων των εμπλεκομένων (χρήστες των εκτυπωτών, υπαλλήλους του τμήματος προμηθειών και προμηθευτές), καθώς και την καταγραφή του ιστορικού των διαγωνισμών για την καλύτερη και αποτελεσματικότερη εποπτεία της διαδικασίας.

Η πλατφόρμα επιτρέπει στους χρήστες να συνδέονται και στην συνέχεια να επιλέγουν το μοντέλο του εκτυπωτή τους, ενώ το σύστημα αυτομάτως αντιστοιχεί τα συμβατά μελανοδοχεία. Εφόσον το σύστημα καταγράφει αυτόματα τα αιτήματα των χρηστών, βοηθάει στην βελτίωση της ταχύτηταw εκτέλεσης των παραγγελιών, ενώ ταυτόχρονα επιτυγχάνεται πλήρης εποπτεία των διαγωνισμών μέσω της καταγραφής του ιστορικού (κόστος, ποσότητες, χρήστες κτλ.).

Για την βοήθεια του διαχειριστή το σύστημα επιτρέπει την αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου (μαζικώς ή μεμονωμένα), την επιλογή χρηστών που θα εκτελέσουν την παραγγελία τους, την καταγραφή κάθε παραγγελίας, την έγκριση ή απόρριψη μιας παραγγελίας από τον διαχειριστή, καθώς και την επεξεργασία μίας παραγγελίας από τον χρήστη. Ακόμα το σύστημα δίνει την δυνατότητα κατάρτισης προϋπολογισμού βάσει έρευνας αγοράς, την αποθήκευση των τιμών όπως διαμορφώθηκαν με το πέρας του διαγωνισμού και την προβολή του ιστορικού ώστε να υπάρχει πλήρης εποπτεία των εξόδων ανά περίοδο προμηθειών, ανά τμήμα ή ανά χρήστη.

Τέλος για να μπορέσει να ελεγχθεί η σωστή λειτουργία του συστήματος σε πραγματικές συνθήκες, εκτελέστηκε μία πιλοτική δοκιμή μεταξύ μελών του Πολυτεχνείου Κρήτης, τα αποτελέσματα της οποίας παρουσιάζονται μέσω μιας έρευνας ικανοποίησης χρηστών που διεξάχθηκε ώστε να μπορέσει να αξιολογηθεί η εμπειρία των χρηστών.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Nikolaos Matsatsinis. Also, I would like to thank Dr. Nikolaos Spanoudakis for his continuous guidance and for our excellent cooperation during this thesis. From the Department of Administrative Computer Infrastructure I would like to personally thank for her help and her time Mrs. Konstantina Glymidaki and George Klados for his help regarding some technical issues that arose during this thesis.

Finally, I would like to express my deepest gratitude to my parents Yianna and Manolis, for their constant support and encouragement throughout my academic life. Equally, I would like to thank my sister Angeliki who puts up with me every day and supports me like no one. Lastly, I would like to thank all those friends who stood by me, and especially those who were always available for me during the development of this thesis.

1 CONTENTS

1 LIST OF ABBREVIATIONS

AMP (LAMP/WAMP)	(Windows/Linux) Apache, MySQL, and PHP
API	Application programming interface
BAM	Business Activity Monitoring
BPM	Business Process Management
BPMN	Business Process Model and Notation
BPMS	Business Process Management System
PAIS	Process Aware Information System
REST	Representational State Transfer
SOA	Service Oriented Architecture
WfM	Workflow Management
WfMS	Workflow Management System

2 LIST OF FIGURES

3 INTRODUCTION

With the emergence of the internet, businesses have to compete in an environment that changes rapidly. In this highly competitive context businesses must find ways to instantly adapt to changes and handle their work in the best possible way. An aid to achieve that is Business Process Management (BPM), which enables businesses to view their work as processes, i.e. as a sequence of tasks that have to be completed in order to produce value. A crucial part of BPM is to map the processes and define how the work is distributed within an organization and what resources are used for every task. With this way businesses can have a full insight of how the work is handled and detect any abnormality, leading to process improvement, which will affect the overall productivity.

In a world ruled by the internet, Business Process Management must take advantage of the technology in order to move from theory to practice. To achieve that, Business Process Management Systems (BPMS) are developed in a need to automate processes. This kind of systems is driven by processes, meaning that the enactment is based on a business process model, and this is the feature that makes those systems so agile and the best alternative in such a competitive environment. More specifically, BPMSs enable the designer to make changes to the developed application just by alternating the business process model. Another advantage of BPMSs is the division of work in processes, where instead of configuring a system that handles every single piece of work in a unified way, processes are configured separately so a potential change does not affect the whole system, making changes easy to apply and enabling businesses to directly adapt to a varying environment.

The purpose of this thesis is to benefit from all the above and attain business process improvement by using BPM techniques and BPMS. More specifically, according to the people in charge for the procurement of ink cartridges in Technical University of Crete, the procurement process is conducted with a counterproductive way, since users make their order either by phone calls or emails, many times without being able to specify the right printer or product, leading to a confusion that is time consuming. So in order to improve the procurement process, an open source Business Process Management System is used to develop a web-based platform where users can log in and order their products from an extensive list of printers and their corresponding ink cartridges. The system also offers to the admin some tools that facilitate the process of procurements. One of the most important objectives of this platform is to reduce the time spent in the whole process by the people in charge. A way to achieve that, is by limiting the admin's engagement with the users by sending mass emails to the users, that include detailed instructions on how to place their order online. Also, another important objective is to lessen the time spent in confusions regarding the models of the ink cartridges, so the order forms are designed in a way that the probability of mistake is minimized. Lastly, in the context of financial management, the platform offers the ability to the admin to view past procurements in a way that the expenses can be tracked.

In the present thesis, the theoretical background of Business Process Management, Business Process Systems, business process modelling and any notion related to BPM will be introduced in Section 4. Following, in Section 5, the open source Business Process Management System will be presented, and the technical background will be given. Section 6, refers to the implementation of the project using the selected BPMS, while in Section 7 the results of this project will be displayed, as they were extracted from the pilot testing of the platform and the satisfaction survey that was conducted in order to obtain the users' feedback. Lastly, the conclusion of this thesis and the lessons learned during the implementation can be found in Section 8 and 9, respectively.

4 BACKGROUND

4.1 THE IMPORTANCE OF BUSINESS PROCESS MANAGEMENT

Before diving into theory and definitions, it is vital to understand what Business Process Management (BPM) is in general and why it is important for the businesses. At first, BPM is a discipline or a "theory in practice", used to *identify* and *design* business processes. Since the processes are identified, they are *executed* and *monitored*. While monitoring the processes, by *analyzing* the results, businesses can find areas to improve. And this is the main objective of BPM, i.e. to improve business processes so that the whole business performance will be improved and objectives will be met with greater agility, leading to value creation. In a few words, BPM enables to view business as a set of processes, and by improving these processes the business's overall performance will be increased.

A main example of how inefficient processes can lead to great loss is the Barings Bank bankruptcy in 1995. Barings Bank was founded in 1762 and it was considered to be one of the most powerful Banks in England, while even the Queen of England was among its clients. The bank went bankrupt after the losses of 1.4 billion pounds that were the fault of just one person, Nicholas Leeson. Leeson made unauthorized speculative trades that at first made huge contributions for Barings, but soon after he started losing large amounts of money while conducting those trades, leading to the complete bankruptcy. Where the importance of BPM comes, is the fact that not only Leeson short-circuited normal accounting and internal control/audit safeguards and processes, but also his actions were not detected by the account regulation procedures even though there were hints that should have caught one's eye. This shows that insufficient process controls and business process failures existed in Barings Bank without anyone taking care of them. With BPM such situations can be prevented since it offers increased visibility and knowledge of the company's activities. By knowing exactly all the activities which take place within the business it is easy to spot the bottlenecks or "abnormalities" and to improve them accordingly. Also, given the Barings Bank example, BPM offers the background to detect any fraud and ensure that all employees comply with the regulations.

Another factor that causes loss for one business is the waiting times. We live in a world where orders are made faster than ever, and in this context goods and accurate information should be transferred without any further delay. There are many examples in the industry where lead times can be reduced either by just rightly assigning job and roles or by recognizing the points where time is wasted. Better lead times mean better customer service, with less cost. Lastly, BPM is of a great importance for businesses because it offers the means to easily adapt to changes. In a worldwide Market, with the omnipresence of internet and e-commerce, it is obvious that businesses need to adapt their strategies straightaway, in order to compete with the competitors. But to do so, many times it is not enough only to know the business processes or to have the optimal processes. In a world ruled by technology, businesses must find a way to make the most of it in order to maintain their presence in the markets. A valuable tool to achieve that, is the Business Process Management systems (BMS) that support the whole BPM process, since a BPMS is *"a generic software system that is driven by explicit process designs to enact and manage operational business processes"* [5].

4.2 BPM AND ITS LIFE CYCLE

Throughout the years, many definitions of Business Process Management have emerged. One of the most concise definitions is the one given from van der Aalst et al. where BPM is defined as

"Supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information." [5]

On the other hand, a more extensive but yet helpful definition to understand what exactly BPM is, is the one given by the Association of Business Process Management Professionals where BPM is defined as a

"disciplined approach to identify, design, execute, document, measure, monitor, and control both automated and non-automated business processes to achieve consistent, targeted results aligned with an organization's strategic goals. BPM involves the deliberate, collaborative and increasingly technology-aided definition, improvement, innovation, and management of end-to-end business processes that drive business results, create value, and enable an organization to meet its business objectives with more agility. BPM enables an enterprise to align its business processes to its business strategy, leading to effective overall company performance through improvements of specific work activities either within a specific department, across the enterprise, or between organizations." [6]

One term that is usually used in the literature but many times is not given enough attention is the "end-to-end business processes". BPM allows to view processes from start to finish, including every task and every person that takes place, in order to monitor and evaluate everything that a business does to achieve a certain result. Only if there is a view of a process as a whole, there can be an improvement that would really affect the performance of a business.

Complementary to the definition, there is the so-called BPM Life Cycle. There are many variations of this cycle but only van der Aalst et al.'s will be mentioned since this is the starting point of every other variation.

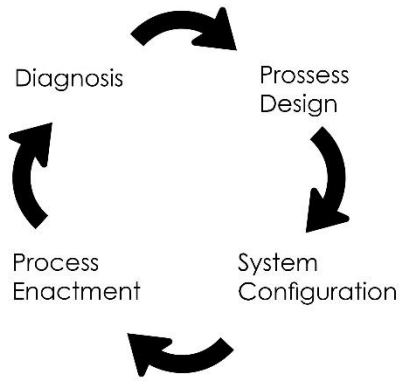


Figure 1 BPM Life cycle

•Process Design: It corresponds to either existing or not existing processes. The goal of this phase is to represent the workflow of the given process using a process model. For example in process design, tasks are defined, the sequence of how the work is done is described, user assignments are made etc.

•System Configuration: This is the phase where the process model retrieved from the designing process, is passed into a Process-aware Information System (PAIS) (see 3.4.1) by "translating" it into a notation that the software can understand (e.g. BPMN, EPC). Then all the parameters are configured. This can be very challenging since this kind of systems usually are not standalone so the connections with web services and other applications have to be configured.

•Process Enactment: After the configuration of the business process model within a Process-aware Information System, the process is deployed and executed. While a process is running it is referred as a process instance. The process instances generate variable values and messages.

Here it should be noted that the enactment of a business process can be manual, automatic or interchangeably manual and automatic.

•Diagnosis: In this phase, monitoring tools (e.g. dashboards) can track the process performance and display it in a manner that is easily understood by the managers. At this phase, any defeasibility and bottleneck will be revealed.

The point of diagnosis phase is to use the new information and feedback –given from the software, as well as the users- in order understand the weaknesses of the running process and re-design it in order to achieve better results.

4.3 Processes and Process Models

4.3.1 Business process definition

In every aspect of everyday life there are activities that are repeated in order to produce value. An example of an everyday process is given in Figure 2. In the same way, within a business there are activities that are repeated and include many people from the organization or third parties (e.g. ordering goods from a supplier, customers' order handling etc.). Since this kind of activities are executed more than once it is crucial to map them and define them, in order to standardize and optimize the business procedures or as they are mentioned -the business processes.

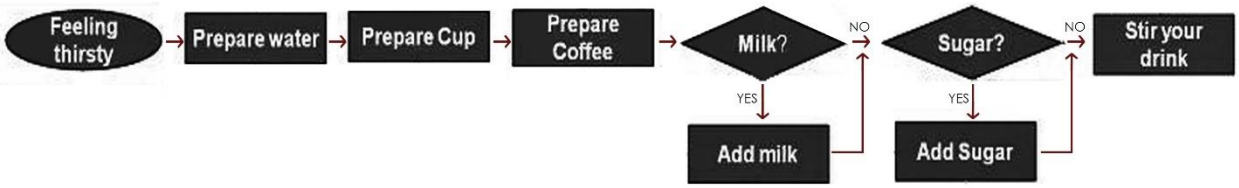


Figure 2 Example of the everyday process of coffee making Adapted from "https://www.business-online-learning.com/"

The first definition of business process was given by Hammer and Champy [8] where according to them a business process is defined as *"a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer. A business process has a goal and is affected by events occurring in the external world or in other processes."*

This definition may be influential and widely used, but it does not include the actors, i.e. the people or the groups who will carry out each activity. To fill this gap in the definition, Van der Aalst et al. and R.K. Ko gave their alternative definitions.

Van der Aalst et al. define a business process as follows: *"By process we mean the way an organization arranges their work and resources, for instance the order in which tasks are performed and which group of people are allowed to perform specific tasks."* [1]

R.K. Ko defines a business process as *"a series or network of value-added activities, performed by their relevant roles or collaborators, to purposefully achieve the common business goal."* [9]

From the mentioned definitions one must keep in mind that business processes include a finite set of well-defined activities/tasks, executed by one or more originators (including people from different departments/organizations) in order to produce an output while following defined rules. The output may be a document, a service, an entry in a database or even a chance of a status.

4.3.2 Business process model

A business process model is the representation of the business process in a way that can be understood by every stakeholder of the Business Process Management by using notations that produce rigorous and unambiguous models (e.g. by using BPMN, UML, EPC). A business process model can depict every activity of a process, including the work and information flow, as well as the decision logic. According to van der Aalst [7] a business process must include information in regard to the following perspectives:

•Resources, i.e. all the requirements and responsibilities e.g. modelling roles, organizational units, authorizations, etc.

•Data, i.e. the way that all the information is interconnected e.g. variables that change values during the enactment, forms, documents, tables etc.

•Time, e.g. deadlines, modelling durations etc.

•Function, i.e. describing the activities and the related applications.

In general, a business process model is not only intended to be executed by software. The business process model may be used by the organization in order to perform various analysis (e.g. "what if" analysis) and evaluate possible outcomes while simulating the business process. Therefore, an aspect that is not often emphasized is the "educational" aspect, where just by mapping and discussing about the modeled business process, valuable knowledge can be obtained.

So given that not every process model is meant to be executed, there are the descriptive models that represent either an existing process or an optimized one, the executable models that can be passed to a software without misconceptions or information loss and the normative models that are governed by rules that limit the way that activities are carried out [7].

4.3.3 Business process modelling languages

In order to represent a business process, a business process modelling language is needed. In most cases, the mentioned languages are graph-based, which means that they use nodes to represent tasks, and arcs to represent the relations between them, i.e. arcs indicate the sequence of the tasks, while there are features to add more logic, like rules and constraints. There are three kinds of such languages, the formal languages, the conceptual languages and the execution languages.

- Formal languages are based on theory, hence they can give well-structured and unambiguous models, but yet the obtained models cannot be easily understood by business people. Examples of such languages are the Petri Nets, Abstract State Machine (ASM), Pi-Calculus and Logic, where their main trait is the formal semantics that allow process analysis.

- Conceptual languages are languages with less formal semantics, that are however easier to be understood by people with no theoretical knowledge. Examples of these languages are the Business Process Model and Notation (BPMN), the Event Driven Process Chains (EPC) and Unified Modeling Language (UML)

- Execution Languages are the ones used to enact the process. These languages will execute the process as defined from the conceptual languages, but in order to prevent any information loss other standards that translate the graphical model into technical code are used. Examples of execution languages are the BPEL, BPML, XPDL.

The systems that support BPM use conceptual languages (by offering Graphical User Interfaces) where the designer can depict the process with the most user-friendly manner and execution languages to

"translate" the mentioned ones. Nevertheless, no matter the kind of the language used, each language support the task structure (control flow) and the data flow. They also support the definition of the way that those tasks will be executed, for example if tasks will be executed in parallel, or if the execution will be determined by a decision. In that way, exceptions can also be handled, for example the way that one process will continue in case of failure can be indicated. Lastly, every language can support time definitions such as duration of tasks, time events like "if a certain amount of time is passed execute these steps" etc.

4.3.4 Business Process Model and Notation (BPMN)

One of the most widely used process modelling language, used by the majority of the BPMS vendors (also used for the implementation of this thesis) is the Business Process Model and Notation (BPMN). As a notation it was developed by Business Process Management Initiative (BPMI), and since 2005 it is maintained by the Object Management Group (OMG). The main reason for BPMN's development was the need for a business process language that will be understood by both business people and technical developers, without any misconceptions, while enabling the modelling of complex processes. Also since there are many executions standards such as BPEL (Business Process Execution Language), BPML (Business Process Modeling Language), XML Process Definition Language (XPDL) etc. there is a need for a unified way to visually represent the processes in every case.

There are four main groups of elements used in BPMN, the flow objects, the connecting objects, the swim lanes and the artifacts.

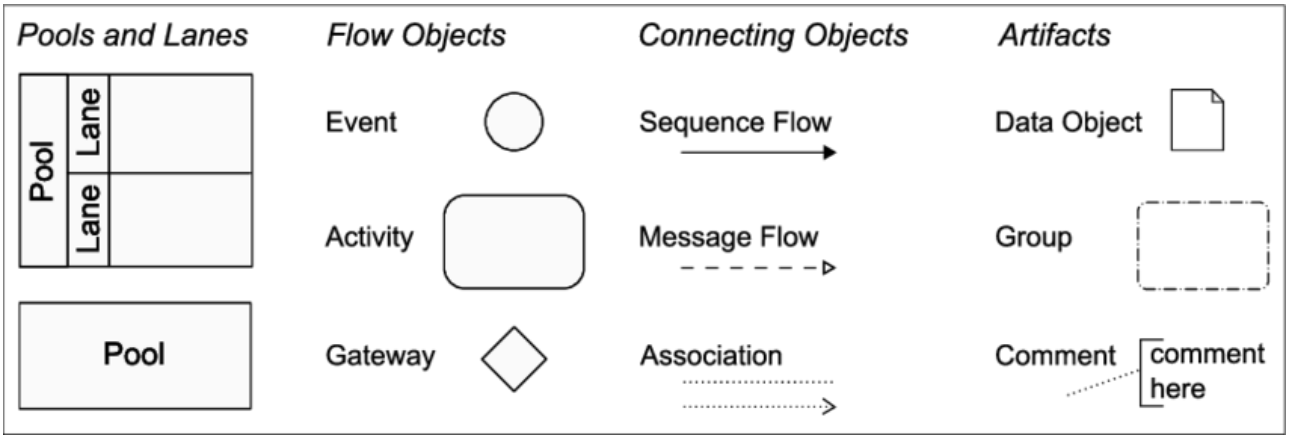


Figure 3 The core elements of BPMN. Adapted from" BPMN for healthcare processes" [14]

As shown in Figure 3 flow objects can be events, activities or gateways.

Types of events:

Events are used in business models in order to represent something that occurs during a process and affects its outcome.

•Start event: Represented with a circle with a narrow single border. With start events the processes can be initiated. The process can be initiated either manually or automatically e.g. if a message is received or if a certain amount of time has passed (timer start event).

•Intermediate event: Represented with a circle with a double border. It depicts something that happens between the start and end events. It may be a message that is received/sent or a timer intermediate event that represents a delay, e.g. wait two days for the customer to place the money, otherwise end the order.

•End event: Represented with a circle with a single thick border. These events terminate the process and they may not be restarted. Also, a message can be sent as soon as the process is finished.

Types of activities:

Activities represent the work that has to be completed within a business. As shown in Figure 3, activities can be tasks, sub-processes or call activities.

•Task: Represented with a rounded-corner rectangle. It represents a single piece of work that has to be done, and it cannot be broken further down. It may be a receive message task, a send message task, a user task, a service task, a manual task or a business rules task.

•Sub-process: Represented with a rounded-corner rectangle with a "+" sign. A sub-process is a separate process that is embedded in another process with its own start and end event.

•Call activity: Represented with a thick rounded-corner rectangle. A call activity is used in order to reuse processes within the new process, since it calls a process that is external to the designed process.

Types of gateways:

A gateway is represented by a rhombus and it is used to direct the flow when two or more tasks are present. There are many supported types of gateways such as event-based, parallel, inclusive, exclusive event-based, complex or parallel event based. The mainly used gateways are the parallel, the exclusive and the inclusive gateways.

•Parallel gateway: It is represented with a "+" in a rhombus and it is used to split the process flow into multiple parallel paths. This gateway may also be used to merge multiple parallel paths into one path but in this case the flow will move on only after each task belonging in the mentioned paths is completed.

•Exclusive gateway: It is represented with a "x" in a rhombus and it is used to split the flow in multiple paths, where only one will be followed based on a condition that will be evaluated as TRUE. It may be also used to merge paths without any limitation.

•Inclusive Gateway: It is represented with a "o" in a rhombus and it is used to split the process flow into one or more parallel paths, based on conditions that are evaluated as TRUE.

Moving on, connecting objects can be sequence flow, message flow and associations. A sequence flow is represented by a solid line and an arrowhead that defines the sequence with which the tasks will be completed in a process. The message flow is represented with a dashed line, an open circle at the start, and an open arrowhead at the end and indicates how the messages will be distributed between the lanes. Lastly, an association is represented with a dotted line and it is used to link artifacts (e.g. shapes of databases, texts etch.) with the process, without affecting the flow in the process.

Artifacts are used to add extra information in a process model, while not affecting in any way the process. They are mostly used as "comments" to indicate data objects and the way that data are exchanged within a process. Also, artifacts can be used to group tasks together without affecting the flow (the grouped tasks placed in rounded-corner rectangle with dashed lines) or to add text in order to provide the reader with extra explanations.

A pool is a rectangle in which the processes are designed. They are used in order to represent an organization and by adding lanes -that are sub-partitions within a pool- all the participants can be represented. In other words, pools and lanes are used to describe who will execute which task.

4.4 Business Process Management Systems (BPMS)

4.4.1 BPMS Components

Business Process Management systems are process aware in-formation systems (PAISs). This means that they (PAISs) can *"manage and execute operational processes involving people, applications, and/or information sources on the basis of process models [4]".* The characteristic that makes this kind of information systems to stand out, is that the execution is done based on a business model which is expressed in a graphic way using languages such as Petri-nets or BPMN. Being a process-aware information system a BPMS can manage and monitor the state of activities defined by the business process model. The main advantage of such systems is the fact that they are *"driven by models rather than code"* [4] enabling faster changes, since those may be applied just by changing the business process model. Also, the presence of a process model helps productivity, as no misconceptions can occur regarding the workflow. Last but not least feature of PAISs, is the ability to monitor the process with data logs, making the bottlenecks of the processes visible.

According to the reference model of the Workflow Management Coalition (WfMC) [10] every BPM system consists of the following components. The main and most important component is the *workflow enactment service* which executes and manages all the activities based on the designed flow by using a runtime environment. In order to do so, one or more *workflow engines* are required, which unlike other execution engines, they execute processes based on a workflow. This flow is designed by using the *process definition tools*, which provide a graphical environment where the designer can depict the business processes, and define how the tasks will be distributed within the business. Also in most cases there are tools to simulate, verify and debug the designed processes. These tools may be used to design slightly different processes and compare them with the existing process, so that better alternatives can be spotted. Since the process is designed and then executed, the end user will engage with the BPM system by using the *workflow client application*, where tasks will be assigned to him/her and will wait to get executed. The last component is the *administration and monitoring tools* where reports and dashboards give information of the processes' status, meaning that they can detect and display the bottlenecks, the trends and can give the overview of the executed process.

Apart from the mentioned components, BPM systems should embed a system for storing the used data, the documents, the files, the business processes etc. Furthermore, given that most of the times this kind of systems are not used as standalone, there should be integration tools that will allow the communication of the system with third-party applications used within the business, or even within other businesses.

4.4.2 Service-oriented architecture and Web Services for the BPM implementation

Service-oriented architecture (SOA) and web-services are important technologies for creating and establishing a BPM environment [2]. What seem to be a reality in large companies is the fact that different applications are used to support different business processes, while there is no direct communication between them, leading to problems regarding data loss, duplicating data, different departments getting mismatched data about the same case etc. Given that, there is a great need to fill the gap between different systems used within an organization, so that accurate data are sent in each department, as well as to provide ways to better serve the customers using all the potentials of the internet. To achieve that, BPM sets the management approach to implement and maintain efficient and integrated business processes,

while SOA and Web services offer the technological background for the integration of all the systems that support those processes. With SOA and SOA tools disparate systems can be integrated, allowing existing systems to communicate. SOA is the best candidate to implement BPM because it is a style of software design focused in business processes, since each business process is broken down into components, and services are built around each process, rather than developing a software that will be responsible for every possible task. This way of software design offers enough independence to modify each service on its own, giving great agility in adapting to changes, since changes do not have to apply to a software that accommodates all the functionalities of the business and a potential change does not affect any other part of the BPMS.

On the other hand, Web services provide the background to implement a Service-oriented architecture (SOA). Web services are reusable software components, that are distributed and programmatically accessible over standard Internet protocols, and they are used to provide certain functionality [13]. Bearing in mind that a web service allows a program to communicate with a web page (eg. by using RESTful APIs[1]), not only communication between internal systems can be realized, but multiple organizations can interact too. Web Services are independent of platform, programming language, and vendor so they can be easily integrated into every BPMS in order to facilitate tasks like ordering materials by communicating with the supplier, or receiving orders placed online by the costumers.

To sum up, with SOA as a "philosophy" and Web services as a tool, business processes or tasks within business processes can be viewed as application pieces accessible through the internet that will be glued together in order to improve the productivity and increase the customer satisfaction.

5 System Architecture

5.1 Problem statement

According to the people in charge of the procurements for the Technical University of Crete there are some processes that need to be automated in order to save time. In an attempt to identify some those processes, people of the Department of Administrative Computer Infrastructure noted that the procurement of ink cartridges is done with a counterproductive way, which is time-consuming and leads them to waste time in a process that can be automated, while there are other more important processes that need to be handled. The current way of carrying out the procurements includes phone calls and emails, where the admin contacts the users and asks them if they wish to order ink cartridges. Most of the times more than one phone call or email is needed, since most of the users cannot properly specify the desired ink cartridge leading to confusion. This can be translated in more time spent in order to resolve the confusion, and the purchase of ink cartridges that may never be used.

For this reason, the main focus of this thesis is to automate this process, allowing the administrators to spent less time in the particular procurement, while engaging less with the users and while offering a solution that would decrease the probability of mistakes when specifying the needed products. This could be achieved with the use of an open-source Process Management System, which with no cost can offer an interface for both users and admin where the procurement of ink cartridges can be carried out. This

[1] "The REpresentational State Transfer (REST) used by browsers can be thought as the language of the internet. An Application Program Interface (API) for a website is code that allows two software programs to communicate with each another. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application. So REST is a logical choice for building APIs that allow users to connect and interact with cloud services." [17]

interface should be a web-based platform where the users can log in and place their order, while the admin can handle every task related to the procurement.

More specifically, the platform should enable the admin to send emails to every user at once, making the contact with the users easier. Then, the users should log in in the platform in order to place their orders. The platform should be as user-friendly as possible, so that the users could place their orders easily without contacting the admin form further directions. In the admin's assistance, the system should automatically collect the placed orders and issue an aggregate list of all the products. In regard to the products, the system should have a database where printers' models will be available, and their compatible inks should be linked, so that when a user selects a printer, then automatically the system would match its corresponding ink cartridges. Also, the system should predict that not every ink cartridge model will be available in the database, so there should be a tool where the admin would be able to add new printers and products without entering the database, but only by using the developed web-based platform.

In this context, the platform should offer solutions to the admin so that he does not have to enter directly in the database. For this reason, there should be a prediction for which values the admin may wish to change, and accordingly to offer the tools with which the changes would be directly applied through the platform. At this point, there should be checks, in order to make sure that no duplicated data would be entered. Lastly, since there should be no engagement with the database, the platform should offer the tools to view stored data e.g. displaying past orders for keeping track of the expenses.

5.2 SELECTING THE BPM SOFTWARE

Before choosing the BPM tool with which the mentioned platform would be carried out, six different open source BPMSs were tested. The tests were conducted by an undergraduate with good knowledge of C and a basic background in programming. With that background, the comparison of the BPM tools is done in order to find the most flexible and easy to handle tool that can run projects designed by a person with no prior experience in developing such projects. The factors that affected decision were the ease of the installation, the ease of designing and running a very basic project and the user interface –having in mind that this interface will be used by users that may not be familiar with such environments.

5.2.1 ProcessMaker 3.2.1

After running a basic process using each of the following software, ProcessMaker by Bitnami was chosen for the implementation of this project. The particular software was chosen because it was easier to understand by a developer with no extended experience, it was the most user-friendly and it was the only alternative where the basic process was fully functional.

The reason why the basic process was fully functional is the fact that ProcessMaker has the architecture of an AMP (LAMP/WAMP) stack, since the Apache Web Server is installed together with the MySQL Database, which is handled through the installed phpMyAdmin, while PHP is used the programming language. Apache Web Server gives access to the application where the admin and the users can log in through a web browser and start new cases. Admin's page is divided in four sections, the Home section where the cases can be executed and tracked, the Designer section where processes are designed, the Dashboards where Business Activity Monitoring (BAM)[2] tools are offered (fully functional only for the enterprise edition) and the Admins section where all the other parameters are configured (e.g. email servers, new users,

[2] BAM tools use real-time logged data that provide information about the status and the results of the executed processes. BAM tools are part of the Business Process Analysis (BPA) where discovery, documentation and analysis of business processes is conducted.

user groups etc.). On the other hand, the user's page has limited options and offers the ability either to start a new case or to execute a task depending on the user assignments.

With the designer tool the developer can describe the processes using an embedded BPMN2.0 modeller. The forms are created with a user-friendly drag and drop widget tool, where the available items offer many alternatives (see section 5.3.1) to achieve any layout. In order to increase the functionality of a form, ProcessMaker enables the developer to manipulate each element through Javascript and jQuery. Also, many form elements can be linked with the database using only SQL statements in order to retrieve information and display it to the user.

The programming of each process is done with pieces of PHP code called *triggers*. Through PHP, the variables can be saved or retrieved from a Database, emails can be sent, variables can be manipulated, and in general through PHP triggers the developer can achieve anything he wishes in terms of process programming. Hence, an advantage of the ProcessMaker is that all the programming is done by solely using the tools offered in the web-based Application.

ProcessMaker was also chosen because there was a variety of helpful content online. The enterprise offers extended documentation –the most complete in comparison with the alternatives- and there are many video tutorials. In addition, there is an active forum where almost every question is answered. Here it should be noted, that while using the community edition of ProcessMaker every question that was submitted in the forum, was answered within a few hours by a person in charge.

Overall, ProcessMaker is quite an easy software for a new user to adapt. As previously said, the programming started with the user only knowing how to code in C, but the good documentation made easy to achieve any task. For the project it was required to learn to code in PHP and Javascript, but the layout of the whole platform, where the programming is done in pieces, made it easy to handle for a user taking his first steps to these programming languages. Also, the offered detailed instructions were very helpful for someone who first encounters the software and needs to make functional processes. Lastly, the main advantage of ProcessMaker is that it can be used as standalone, without the need to be intergraded in another application (as needed in many of the following alternatives).

5.2.2 Camunda BPM 7.8.0

Camunda BPM is an open-source workflow and decision automation platform created by Camunda, used by major companies worldwide. It is a lightweight Java-based framework that can be used as a standalone process engine server or embedded inside custom Java application [15].

As far as the installation is concerned, even though the installation itself is easy, there are some prerequirements that make it more complex. Starting, since it is a Java-based framework a Java JDK is required, whereas other programs are needed too. For example, in order to design a BPMN model the user has to separately install the Camunda's BPMN modeller which complies with the BPMN 2.0 standards or to use the BPMN tool of Eclipse. Since the installation is completed, the creation of a project has to be done. In order to do so, a Maven Project has to be created through Eclipse and the extracted file has to be deployed by the Apache server. This means that the creation of a project is not done by solely using Camunda's platform, but the Eclipse IDE is required too. Also, designing projects through Eclipse means that the forms have to be created either by using HTML or by using the BPMN modeller. On the one hand, using HTML enables the developer to achieve any layout, but evidently that requires the extra knowledge of HTML, and on the other hand, the BPMN modeller is more user-friendly and does not require any extra knowledge, but it is not so flexible.

With that said, it is made clear that in order to work with Camunda, programming in Java is needed. Also for the deployment of complex projects, technical knowledge is needed and the platform is targeted mainly to software developers. The fact that as a platform it is targeted mainly to software developers can be spotted in the short documentation that takes as granted that the developer has knowledge of Java and Javascript. Also, there is not a variety of helpful content in order for a new user to easily adapt to the platform.

So given the fact that the platform requires excessive knowledge of Java, and doesn't provide an interface for the less experienced developers, in addition to the limited documentation, Camunda had to be rejected for this particular project.

5.2.3 Bonita BPM 7.6.3

Bonita is an open-source business process management and workflow application platform created in 2001 by BonitaSoft. It uses the Bonita BPM Engine which is a Java application that executes process definitions created with Bonita Studio. The Bonita studio has an embedded BPMN modeller (BPMN 2.0 compatible) where the processes are set up and can be used to connect a process with other information systems [16].

The installation of the Bonita BPM is quite easy, and by the finish of the installation two different parts will be available, the Bonita BPM studio, where processes are designed and configured, and the Bonita BPM platform where the applications are executed and the users get involved through a web browser. Within the Bonita Studio the processes can be designed using BPMN and the platform can be configured to communicate with third-party systems such as email servers, external databases, ERP and CRM systems. Apart from the studio, Bonita comes with the so-called "UI Designer" which enables the developer to create forms in a very user-friendly manner, using drag and drop widgets, meaning that there is not a requirement for HTML knowledge. The variables' handling within the forms is done by using APIs or JSON, so knowledge of web services and APIs is needed, while the main programming is done by using Java.

What can be noted as an advantage of Bonita BPM, is the fact that there is extended documentation and a variety of online tutorials, making the platform easy for a new user. Here it should be mentioned that as a platform is not so easy to understand for a developer with limited Java knowledge and that may be a problem, since there is not such an active community and the company gives more attention to the enterprise edition. Overall Bonita was considered as a good alternative of the chosen BPM tool, but the fact that it was not so easily understood by a new user had an impact on the final choice.

5.2.4 Activiti 7.0

Activity is a light-weight Java-based engine created by Alfresco, used to execute business processes. Activiti runs its own Apache server so before installing, a Java JDK is needed and Eclipse may be required depending on the complexity of the process that has to be designed. For non-basic projects, Eclipse is required. The installation of the platform is easy, and Activiti comes with a built-in BPMN modeller which complies with the BPMN 2.0 standards. This modeller is easy to use, but in some cases the developer may want to work with the BPMN plug-in of Eclipse. The forms can be created either by using the BPMN modeller or by using Eclipse, where in both cases the supported elements are limited. As a Java-based platform REST calls are used for the communication with the databases and APIs can be used for the communication of the designed project with other applications, whereas the programming of the processes is done by using Java and Javascript.

Activiti has a solid presence as an open-source BPM tool and for that reason there is good documentation and many online tutorials making the first basic project easy to carry out. The reason why the platform was rejected for the given project is that for the actual implementation of a real-world project, a solid

technical background is required. Additionally, the user interface is pretty basic and may be difficult for a user who encounters the platform for the first time, so the platform has to be embedded in other applications.

5.2.5 jBPM 6.4

JBPM is a platform created by Red Hat, used to design and execute business processes described in BPMN. It is a process engine written in pure Java and offers a toolkit that allows the developer to build his applications either as standalone Java applications or integrated into any other Java service.

The installation of the platform was the most difficult in comparison to all the alternatives and required the installation of other Red Hat's software, but this can be overlooked since the installed platform offers a variety of tools. More specifically, after the installation one can access the web-based Business Central Application, where modelling, deployment, execution and monitoring tools are available for any process. Only by using this Central Application, the developer can design his own processes, business rules and data models.

The platform also offers an embedded BPMN 2.0 compatible modeller, where all the processes can be designed. Also, there is a Form Modeller, where the forms can be created by using a user-friendly drag and drop widget menu. Here it must be noted, that even though this modeller is user-friendly, there are alternatives that offer better similar drag and drop elements.

To use jBPM one must know how to program in Java, since the philosophy of building a project is the same as the Java philosophy. Furthermore, as stated earlier, Java APIs and REST APIs can be used to connect the projects with other Java or web-based applications. No matter what the project is, the good documentation will help any developer, and the online tutorials will enable a new user to easily adapt to the platform.

Although jBPM is a powerful platform, the excessive use of Java logic, made it difficult to understand by a user with the skills mentioned earlier and this is the reason why it was rejected for this project.

5.2.6 YAWL 4.2

YAWL (Yet Another Workflow Language) is a workflow language created by researchers at Eindhoven University of Technology and Queensland University of Technology, who intended to create a workflow language that would handle any possible workflow pattern. To achieve that, researchers extended Petri Nets -which were lacking some possible patterns- by adding some new workflow patterns.

In order to support this language, YAWL BPM system was developed. This system comes with the execution engine, a worklist handler and a graphical editor where the processes are designed using YAWL. Nonetheless, there is the BPMN2YAWL component, where a BPMN diagram can be translated into YAWL. For the data definitions the platform uses XML and the data are handled by using XPath and XQuery. Furthermore, Java programming and REST APIs are supported in order to integrate a project into an already existed application.

Overall, even though YAWL is quite easy to understand and handle, it has some drawbacks that could not be overlooked and led to the rejection of the platform. The main drawback is the outdated graphics and the complicated user interface. It is nonfunctional to use a platform with such graphics as a standalone application, since it reminds of another decade. Also, there is not good documentation, the online tutorials are few and recorded many years ago, and any reference to the software is mainly focused on the theoretical background of YAWL as language.

Following there is an aggregate table where the features of the mentioned open-source BPM tools are extensively described.

Criteria	ProcessMaker BPM	Camunda BPM	Bonita OS
I. Supported Languages & Standards			
I.1 Supported BP Modeling Languages	BPMN 2.0/Not all	BPMN 2.0/All elements	BPMN 2.0/Not all
I.2 Interoperability	Import/Export BPMN	Supported/BPMN 2.0 Roundtrip	Import : BPMN 2.0, XPDL 1.0, jBPM 3.2/Export BPMN 2.0, image files
I.3 Supported Perspectives	Not Supported		
I.4 Modeling of Business Rules	Subscription	Supported	Partially Supported
I.5 Generate Process Documentation	Supported/PDF, DOC	Supported/HTML, JavaScript	Subscription/PDF, DOC, HTML, RTF, PPT
I.6 Reuse BP Models	Supported	Not Supported	Not supported in the community edition
I.7 Modeling of PPI	Subscription	Not Supported	Subscription
II. Process Design Tools			
II.1 Supported Programming Languages	PHP & JavaScript	Java & JavaScript	Java, Groovy scripts
II.2 Designing User Interfaces	Supported/runs in a browser & starts a web server	Supports both	Supported/runs in a browser & starts a web server
II.3 Support for importing organizational structure	LDAP & Active Directory	LDAP & Active Directory	Subscription
II.4 Support for assigning roles to users	Supported/Manual	Supported/Manual	Supported/Manual
II.5 Support for adding SLA & KPIs	Subscription	Subscription	Subscription
II.6 Translation into executable models	Partially Supported	Supported	Supported
II.7 Supported BP Execution Languages	ProcessMaker	Not specified	Not supported
II.8 Managing user access level	Supported	Supported	Supported
II.9 Support Exception Handling	Not Supported	Supported	Supported
III. Deployment Criteria			
III.1 Support for distributed execution	Not Supported	Supported/Clustering	Subscription/Clustering
III.2 Supported Server OS	Linux, Windows / Unix, PHP	Windows, Mac OS X, Linux	Windows, Linux
III.3 Support for integration with other systems & services	REST API & WSDL	Java & REST API	Java & REST API
IV. Execution & Operation Criteria			
IV.1 Version management of BP models	Supported	Subscription	Supported
IV.2 Support for informing users	Subscription	Supported	Subscription
IV.3 Document management	Supported	Not Supported	Supported
IV.4 Support for calendar management	Supported	Not Supported	Supported
V. Monitoring & Control Criteria			
V.1 Support for technical monitoring & control	Subscription	Not Supported	Supported
V.2 Change the role of a resource for an activity	Supported	Supported	Supported
V.3 Support for informing users in case of failures	Not Supported	Supported	Supported
V.4 Support for business activity monitoring (BAM)	Supported	Supported	Subscription
V.5 Support for changing business rules	Subscription	Subscription	Not Supported
V.6 Support for changing the workload balance	Supported	Supported	Subscription/Clustering
V.7 Support dashboards and reports	Supported	Supported	Subscription
VI. Analysis Criteria			
VI.1 Support for process verification	Subscription	Supported	Supported
VI.2 Support for process simulation	Not Supported	Not Supported	Not Supported
VI.3 Support for recording historical process execution data	Subscription	log files & DB	Subscription
VI.4 Support for BI and Process mining tools	Subscription	Supported	Subscription
VII. Other Criteria			
VII.1 Training	Supported	Supported	Supported
VII.2 Latest Release	April 15, 2019	November 30, 2018	December 6, 2018
VII.3 Fully open-source	No	No	No
VII.4 Founded year	2000	2008	2009
VII.5 Ease of installation	Easy	Difficult	Easy
VII.6 Installation prerequisites	No	JDK 7+, Apache Maven	Java version 8 update 121
VII.7 Execution Engine	Yes	Yes	Yes
VII.8 E-mail	Subscription	Yes	Yes
VII.9 Web-based	Yes	Yes	Yes
VII.10 Digital signature	Partially	Not Supported	Not Supported
VII.11 Secure Sockets Layer (SSL)	Not Supported	Not Supported	Supported

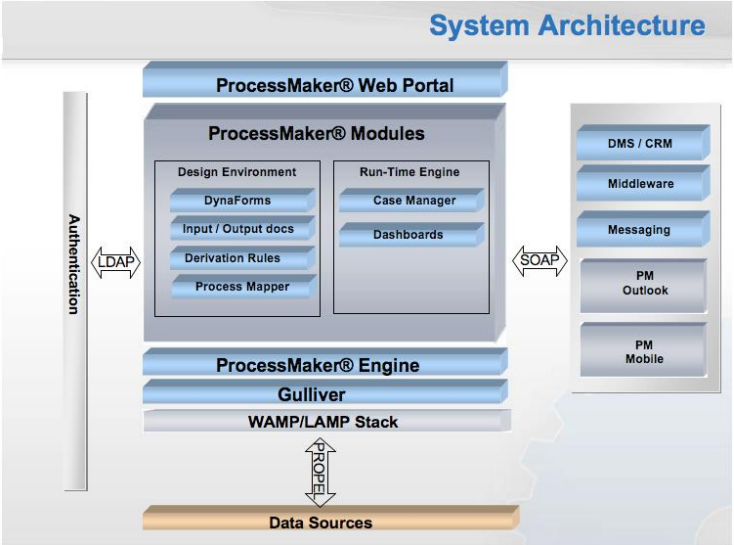
Figure 4 Comparing BPM tools (joint work with Stefanos Triantis)

Criteria	Activiti BPM	JBoss jBPM	YAWL
I. Supported Languages & Standards			
I.1 Supported BP Modeling Languages	BPMN 2.0	BPMN 2.0/Not all	YAWL
I.2 Interoperability	Import/Export BPMN 2.0	Import/Export BPMN 2.0	PNG images, CPN models
I.3 Supported Perspectives			control flow, data, resource
I.4 Modeling of Business Rules	Supported/Drools	Supported/Drools & constraints	Supported
I.5 Generate Process Documentation	Subscription/MS office, PDF	Supported (Auto)/Excel, CSV	Supported/Auto
I.6 Reuse BP Models	Supported	Supported	Supported
I.7 Modeling of PPI	Not Supported	Not Supported	Not Supported
II. Process Design Tools			
II.1 Supported Programming Languages	Java & JavaScript	Java, JavaScript, MVEL	Java
II.2 Designing User Interfaces	Both are supported		Supported/ JavaServer Faces
II.3 Support for importing organizational structure	LDAP & Active Directory	Supported	Supported
II.4 Support for assigning roles to users	Supported/Manual	Supported/Manual & Auto	Supported/Manual & Auto
II.5 Support for adding SLA & KPIs	Not Supported	Not Supported	Not Supported
II.6 Translation into executable models	Supported	Supported	Automatic
II.7 Supported BP Execution Languages	Not specified		YAWL
II.8 Managing user access level	Supported	Supported	Supported
II.9 Support Exception Handling	Supported	Supported	Supported
III. Deployment Criteria			
III.1 Support for distributed execution	Supported/Clustering	Partially Supported/VFS Clustering	Not Supported
III.2 Supported Server OS	Windows, Linux, Mac OS X	Windows, Mac OS X, Linux	Windows, Linux, Mac OS X
III.3 Support for integration with other systems & services	Java, REST API & WSDL	(REST, JMS, Java) API & WS	Web Services
IV. Execution & Operation Criteria			
IV.1 Version management of BP models	Supported	Supported	Supported
IV.2 Support for informing users	Supported	Supported	Partially Supported
IV.3 Document management	Subscription	Supported	Supported
IV.4 Support for calendar management	Not Supported	Supported	Supported
V. Monitoring & Control Criteria			
V.1 Support for technical monitoring & control	Not Supported	Not Supported	Not Supported
V.2 Change the role of a resource for an activity	Supported	Supported	Supported
V.3 Support for informing users in case of failures	Supported	Supported	
V.4 Support for business activity monitoring (BAM)	Not Supported	Supported	Partially Supported
V.5 Support for changing business rules	Not Supported	Not Supported	Partially supported
V.6 Support for changing the workload balance	Supported		Supported
V.7 Support dashboards and reports	Partially Supported	Supported	Not Supported
VI. Analysis Criteria			
VI.1 Support for process verification	Supported	Supported	Supported
VI.2 Support for process simulation	Supported	Supported	Supported
VI.3 Support for recording historical process execution data	log files & DB	log files, DB & JMS	process log files
VI.4 Support for BI and Process mining tools	Supported	Supported	Supported
VII. Other Criteria			
VII.1 Training	Partially Supported	Supported	Supported
VII.2 Latest Release	May 26, 2017	January 14, 2019	Nov 29, 2017
VII.3 Fully open-source	Yes/Alfresco	Yes/Red Hat	Yes/Rheni & Acclario
VII.4 Founded year	2010	2014	2002
VII.5 Ease of installation	Difficult	Difficult	Easy
VII.6 Installation prerequisites	JDK 6+, Eclipse Indigo/Juno, Apache Tomcat	JDK 6+, Apache Ant 1.7 +	JRE 6+
VII.7 Execution Engine	Yes	Yes	Yes
VII.8 E-mail	Yes		Yes
VII.9 Web-based	Yes	Yes	Yes
VII.10 Digital signature			Supported
VII.11 Secure Sockets Layer (SSL)			Not Supported

Figure 4 Comparing BPM tools (joint work with Stefanos Triantis)

5.3 ARCHITECTURE

As said above, the chosen solution for this thesis is ProcessMaker. ProcessMaker is an AMP stack, meaning that an Apache server is installed, along with MySQL database and phpMyAdmin for the database management, while PHP is used as the main programming language. In this thesis ProcessMaker is used as standalone, and it is no embedded in any other application or webpage, meaning the users and the admin



Adapted from "https://wiki.processmaker.com"

login in the web environment provided by the tool. The architecture of ProcessMaker can be seen in the given diagram.

Also, In order to make the platform available to the public a public IP address was needed. For this reason a virtual machine was requested from the university, so that the platform would be hosted. The requested virtual machine has Ubuntu 18.04 LTS operating system, with 60GB hard disk drive and 4GB RAM.

6 IMPLEMENTATION

The present section describes the implementation of the thesis project using ProcessMaker 3.2.1. For the purpose of the thesis, a platform was developed, which will be extensively described and for now on it will be referred as “Ink BManager”.

6.1 DEFINING PROCESSES BY USING BPMN

As stated earlier, BPMN is a notation used to graphically represent processes across businesses. In the case of ink cartridges’ procurement, nine processes had to be designed. In this project there is one process where customers are involved by placing their orders, one process where customers can contact the admin and seven processes that are used for the admin’s assistance.

ProcessMaker comes with an embedded BPMN tool that complies with the Business Process Management Notation (BPMN) 2.0 standard. This tool is very powerful and can be used to design pretty complex processes. However, not every single element of BPMN 2.0 is supported, but the given elements should be sufficient to build a fully functional process.

Below all the processes will be described.

6.1.1 Process of ordering ink cartridges

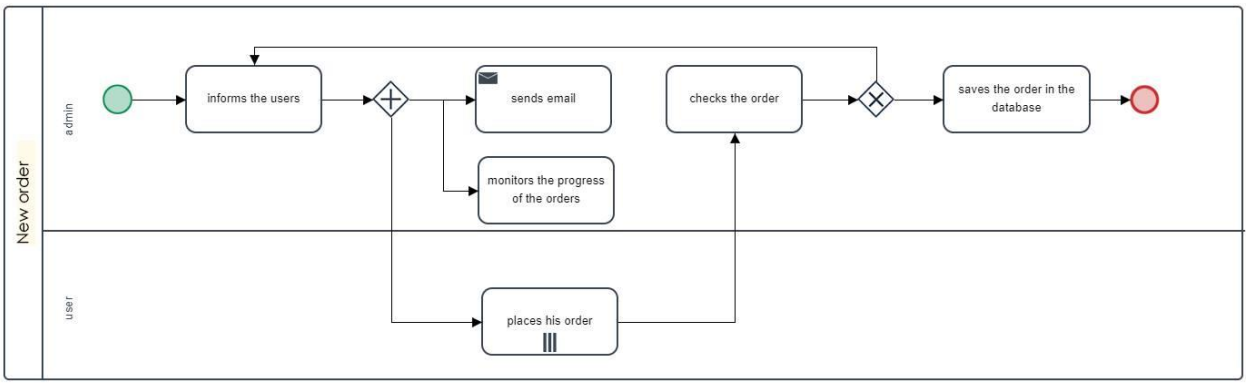


Figure 6 Process of ordering ink cartridges

This process is made in order to automate the process of ordering ink cartridges. The main purpose is to replace the previous manual process, where admin had to manually contact the users and the users placed their order via email.

The process starts with a start symbol represented by a green circle, and it is followed by a task which is used by the admin. In the task “informs the users” the admin can select which users will have to place their order. After the selection is made, there is a parallel gateway (a cross in a rhombus) which is used to simultaneously start three tasks. The first task which is completed is an email task (“sends email”), where emails are automatically sent to users, in order to be informed for the beginning of the new procurement period. The next task to be started (but not completed) is the one where admin can monitor the progress of the orders. In the task “monitors the progress of the orders” he can view which users have placed their order and which ones have an open order. Simultaneously with the above two tasks, a parallel task (“places his order”) is started, in which every user can place his order, since every user’s form has been opened. The present task is the only one that can be claimed by the users in the particular process.

At this point, a special reference has to be made in order to describe the parallel task. A parallel task in ProcessMaker and in every other system that uses BPMN notation is marked with three vertical lines. This kind of tasks is used when many users have to do the same task. In this project many users have to fill the order form. A drawback of the parallel task is the fact that the process cannot move on until every single user fills his form. To bypass this problem another process is designed and the solution will be discussed later (see section 6.5.1).

As soon as every user has filled his form (or the admin chooses to close every user’s form), the process will move on to the next task “checks the order” and the task “monitors the progress of the orders” will automatically end. In the task “checks the order”, admin can accept or deny an item and he may want to impel a user to change his order. This is the reason why the “checks the order” task is followed by an exclusive gateway (an x in a rhombus). Admin will have to choose if he wants to impel a user to fill his form again, or if he wants to proceed to the task where the final order is saved. In the first case, the process will be leaded in the beginning task (“informs the users”), where the admin will choose which users will place again their order and then the process will continue in the same way as before. In the second case, the process will move to the task where the final order is saved. In the task “saves the order in the database”, all the requested items are gathered together and they are saved, so that the admin can have a shorted list of all the items needed for the ink cartridges’ procurement. After the last save is made, the whole process ends with an end symbol which is represented by a red circle.

6.1.2 Process of budget creation after price comparison

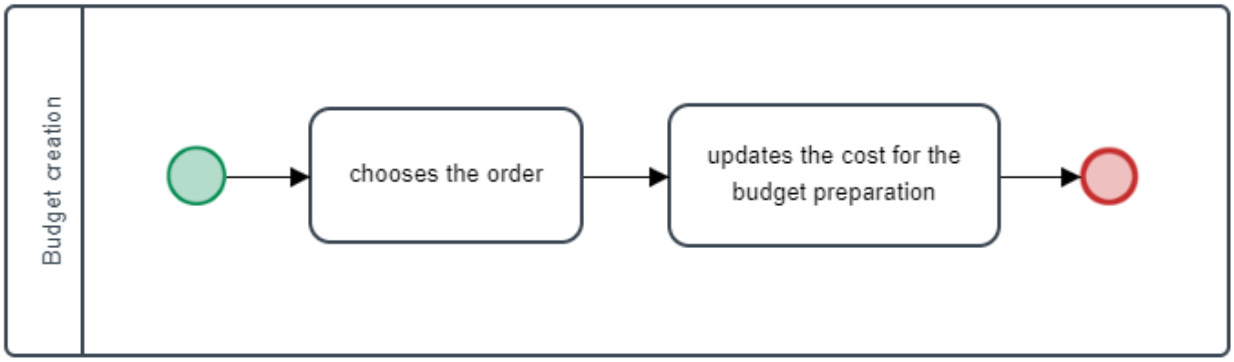


Figure 7 Process of budget creation after price comparison

The particular process is made in order to create a budget with the prices of the ordered ink cartridges. The prices are derived from the price comparison, which is conducted by the admin, and they are used for knowing the range of prices in the market, so that when the quotations are given, the admin will be able to choose the best alternative.

The process starts with the start symbol, and it is continued with the task “chooses the order”. In this task, the admin will have to choose if he wants to prepare a new budget or if he wants to view a budget as it was placed in a past procurement. Also, the admin will be capable to edit the budget of the running period.

After the choice is made, the process will move on to the task “updates the cost for the budget preparation”, where the admin will fill the prices of the products. When done, the whole process will end.

6.1.3 Process of updating the cost

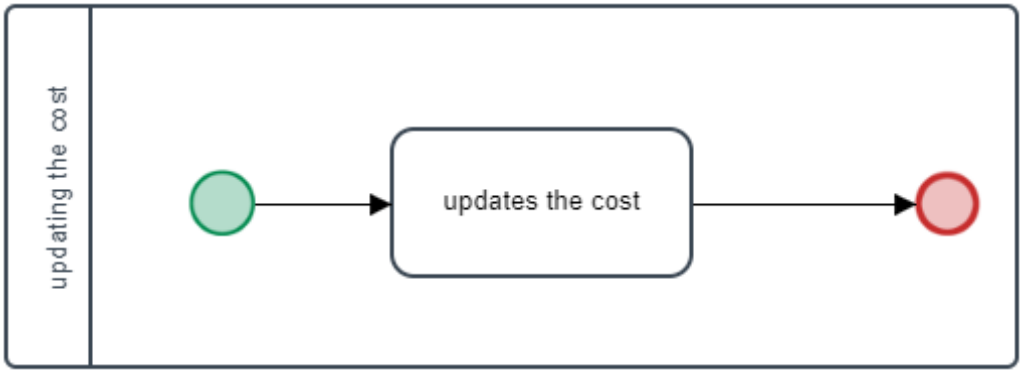


Figure 8 Process of updating the cost

As shown in Figure 8 the whole process consists of one and only task, the task “updates the cost”. This process is started when the quotations are given and it is used by the admin in order to save the prices as they were formed after the tender for supplies. In the same form, the admin will have to add the supplier’s information. After he fills the form, the process will end.

6.1.4 Process of viewing the record

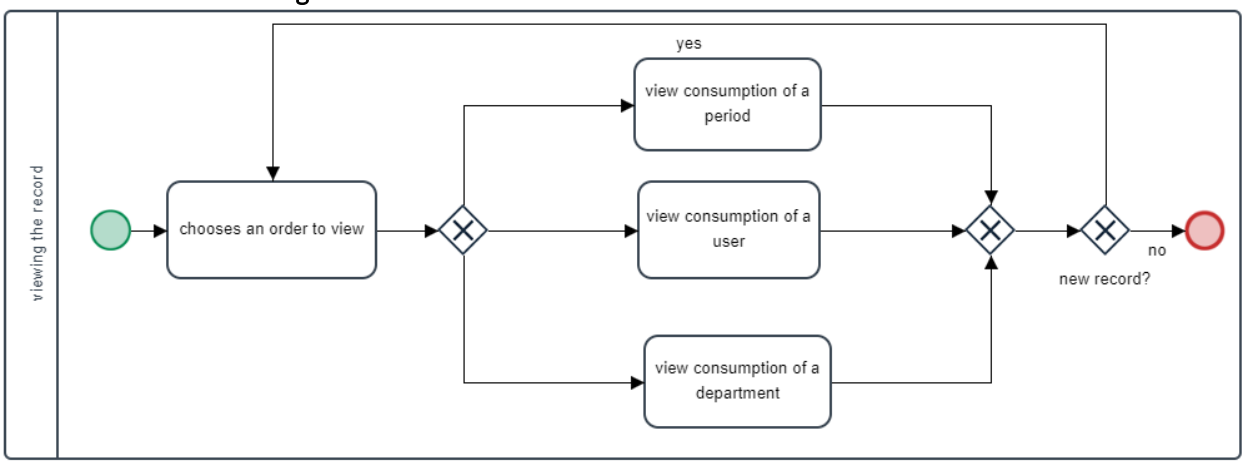


Figure 9 Process of viewing the record

This process is made so that the admin can view past orders, as well as for viewing the consumption of each user or department, in order to keep track of the expenses.

The process starts with the task "chooses an order to view", where the admin firstly selects which order he wants to view and then which information he wants to view. This task is followed by an exclusive gateway which routes the process based on admin's selection. The gateway leads to three different tasks, which will populate three different forms, where the admin will either view the order as it was placed, or based on user's selection, or based on the order of a department.

Every task is joined in an exclusive gateway since only one task can be claimed each time. After the first exclusive gateway, another exclusive gateway is placed in order to re-route the process in the case where the admin wants to view another order. In this way when the admin chooses to view another record, the process will be routed in the first task and then will follow the flow as before. The process will end when the admin will choose not to view another record.

6.1.5 Process of adding a new printer/ product or a new department/ office

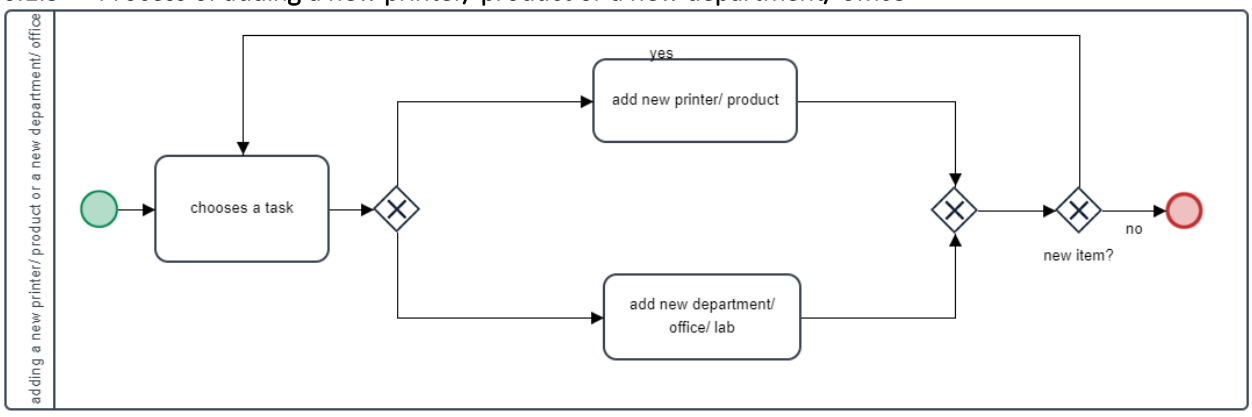


Figure 10 Process of adding a new printer/ product or a new department/ office

This process is designed to give the admin the capability to add new printers or ink cartridges, as well as new departments, offices or laboratories.

The process starts with the start symbol and continues on the first task “chooses a task”. In this task, the admin will have to choose if he wants to continue by adding a new printer/ product or by adding a new department/ office/ lab in the database. Following this task, there is an exclusive gateway which routes the flow depending on the admin’s choice. Given that, the flow splits into two other tasks, and then it is merged again in an exclusive gateway. Following this, there is another exclusive gateway which routes the flow in the beginning, in the case where the admin wants to add another printer, product, department or lab. The process ends when the admin states that he wishes to exit the process.

6.1.6 Process of reopening a case

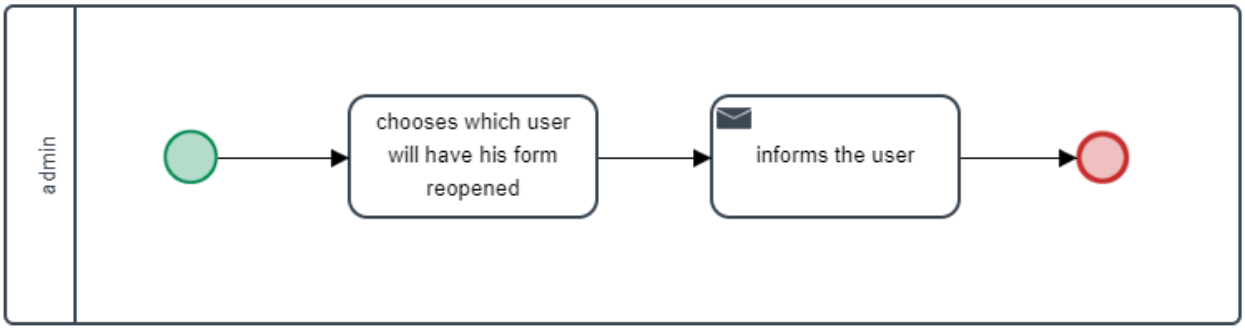


Figure 11 Process of reopening a case

The particular process is designed to reopen the order form of a user. Given that once a user places his order he doesn’t have the ability to edit it, there should be a way where, in case of a mistake, the order will be modified. Therefore, when a user wishes to change his order he will contact the admin and then the admin will start this process.

The process starts with the task “chooses which user will have his form reopened”, where as stated in the title, the admin chooses the user who needs to edit his order. Following that, there is an automatic email task “informs the user”, where an email is sent to the user in order to notify him, stating that he can now edit his order. When the email is sent, the whole process ends.

6.1.7 Process of cancelling a user's order

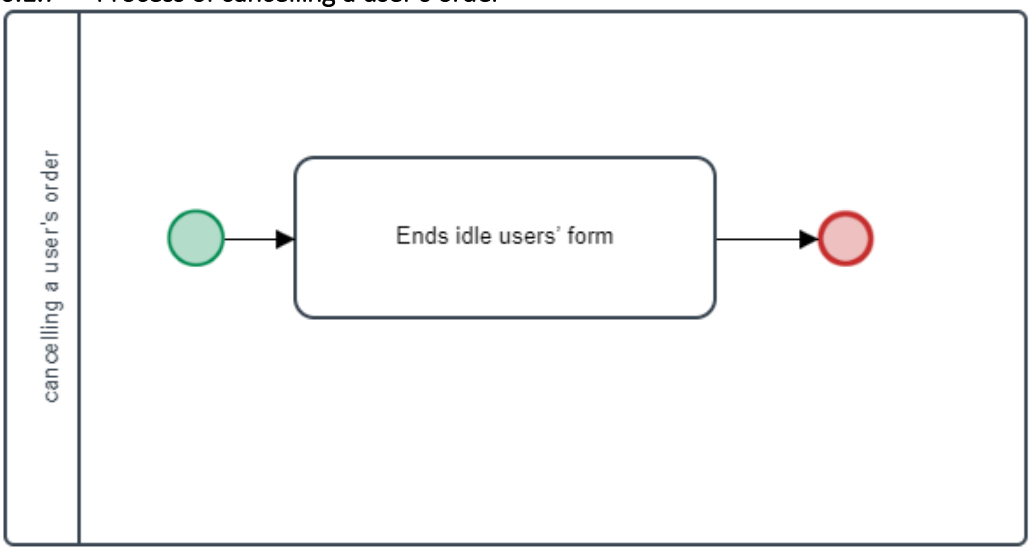


Figure 12 Process of cancelling a user's order

Earlier there was a mention of the parallel tasks. It was noted that a drawback of this kind of tasks is the fact that every single user has to claim and end his task in order for the process to move on. To bypass this problem this process was created. So when the admin decides that he can no longer wait for some of the customers to place their order, he starts this process in order to deactivate their forms. When this is done for every idle user, the "Process of ordering ink cartridges" can normally move on.

In regard to the BPMN, the process starts and then the flow is routed to the task "Ends idle users' form" which has to be claimed by the admin. When the admin is done with this task the process ends.

6.1.8 Process of contacting the admin

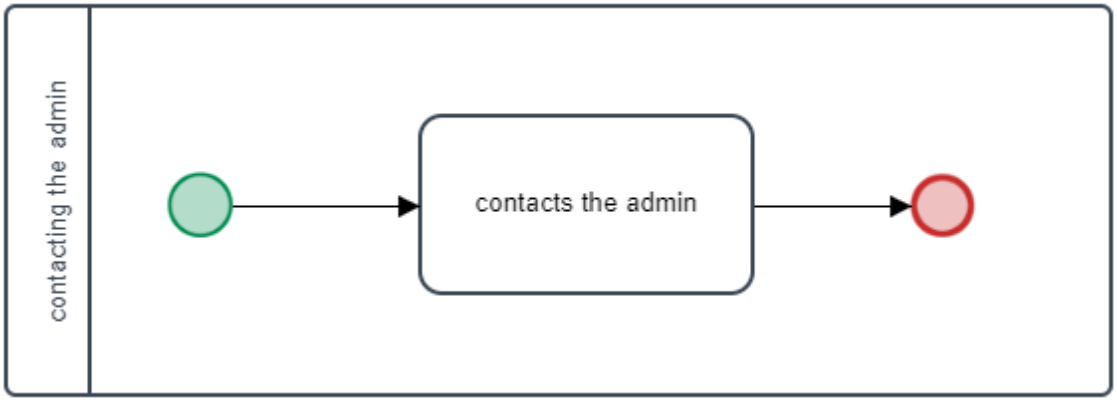


Figure 13 Process of contacting the admin

For the needs of this project, it was necessary to add a process where the users will have the ability to contact the admin. For this reason, the particular process was designed. This process can be started at any point, by any user, and enables users to send a mail to the admin directly from the platform.

As seen in Figure 13 this process consists of only one task. The process starts and when the email is sent, it ends.

6.1.9 Process of contacting the users

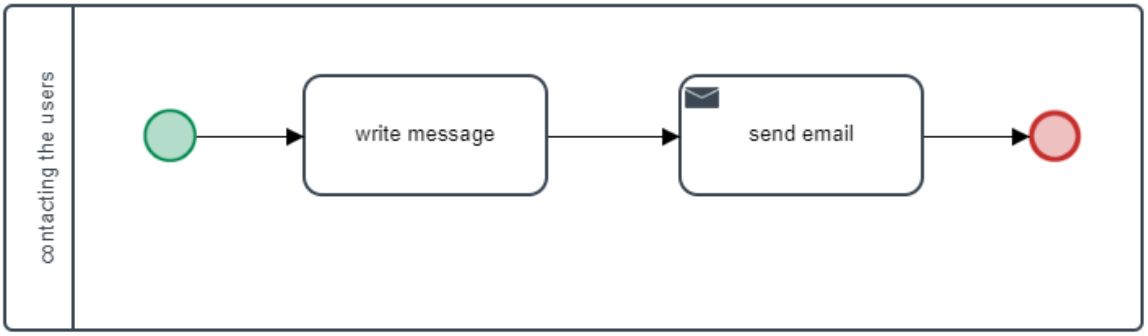


Figure 14 Process of contacting the users

The last process that was designed is the one shown in Figure 14. This process enables the admin to send emails through the platform, reaching one user at the time or many users at once. The process is started by the admin and afterwards he chooses the recipients and he composes the email. Then automatically the emails are sent and the whole process ends.

6.2 TASK ASSIGNMENT

After the tasks are defined and the BPMN models are created, the designer should determine which person will claim which task. ProcessMaker gives many options for doing that so that every case scenario will be supported.

The first supported way to assign a task to users is the “cyclical assignment”, where the task is assigned to a group of users and it follows a round-robin pattern so that every user is able to claim the task without affecting the other users. The main purpose of this kind of assignment is to spread the work evenly within a group. The second way to assign a task is the “value based assignment”. This kind of assignment is used many times in this project and the reason is that the users that will be able to claim the next task are determined by a variable. With this way, the designer has great flexibility in choosing which one will put through one task. Another dynamic way of assigning tasks to users is “self service value based assignment”, where the users who are allowed to claim one task are set by one variable.

The mentioned ways are the ones that are used in the project, however it is useful to mention all the possible ways of assigning tasks with ProcessMaker. Another way of assigning tasks to users is the “manual task”. In this case, when a user finishes his task he will determine the next person who will put through the next task. Apart from that, ProcessMaker with the “reports to” task assignment can assign tasks while taking into consideration the structure of an enterprise. In many cases a supervisor will have to review the work of his subordinate, so when a subordinate finishes his task, the next task will be automatically assigned to his manager in order to review his work. The last scenario that has to be covered is when a task is assigned to a group of users, in order to only one user to claim it. This kind of assignment can be done with the “self service task” which is made to reduce the workload, since one user can commit himself in this task.

Further down, there will be given examples of the task assignments as used in the project.

Earlier it was stated that the Ink B Manager consists of nine processes, where the seven of those are addressed to the admin. Given that, most of the tasks are assigned to the admin with a “cyclical assignment”. In order to make an assignment in ProcessMaker, one should select the task of interest, and then

by right clicking on this, will have to choose “Assignment Rules” from the shown up list, as shown in Figure 15.

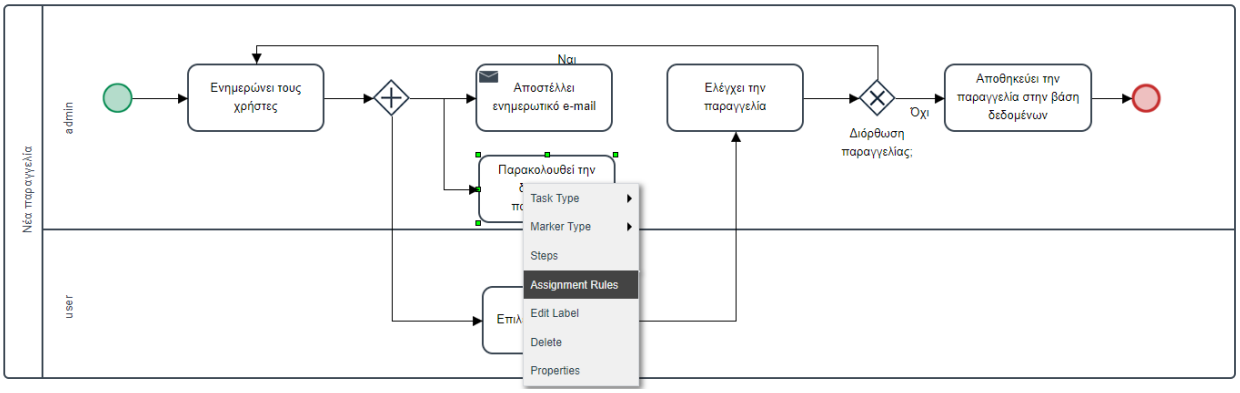


Figure 15 The way of assigning tasks to the users

When “Assignment Rules” is selected, a new window appears. In that window “Cyclical Assignment” has to be selected from the “Case assignment method” and the users that will be committed to the task have to move from the “Available users list” to the “Assigned users lists”. In the particular case, admin has to claim this task so the “Assigned users list” should only include the administrator, as shown in Figure 16.

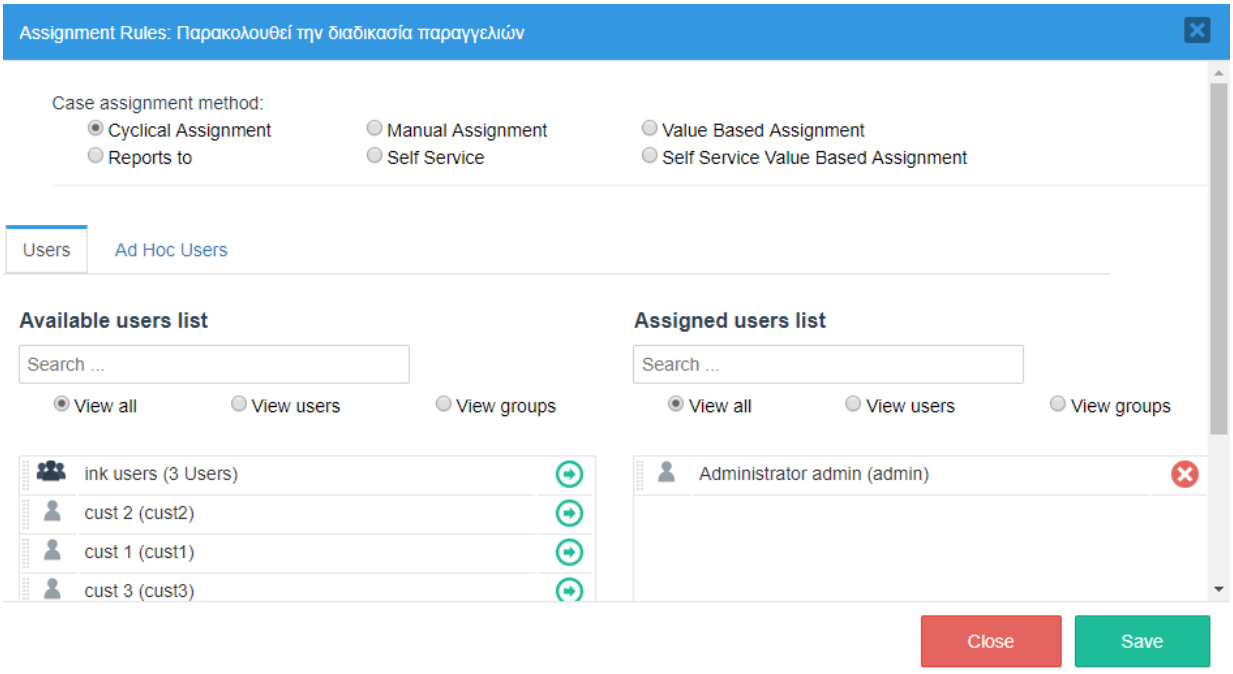


Figure 16 Assigning task to the admin

However, one task can be claimed by many users. In this case, in the “Assigned users list” will be added either a group of users or every user singly. It is recommended to assign groups of users, because with this way it is easier to delete an existed user or to add a new one in the users' list. For example, in the

project a task where the users will be able to contact the admin was created. In this case, a “cyclical assignment” to the group of users “ink users” had to be made. With this “cyclical assignment” every user, at any point can claim the task without affecting the work progress of another user.

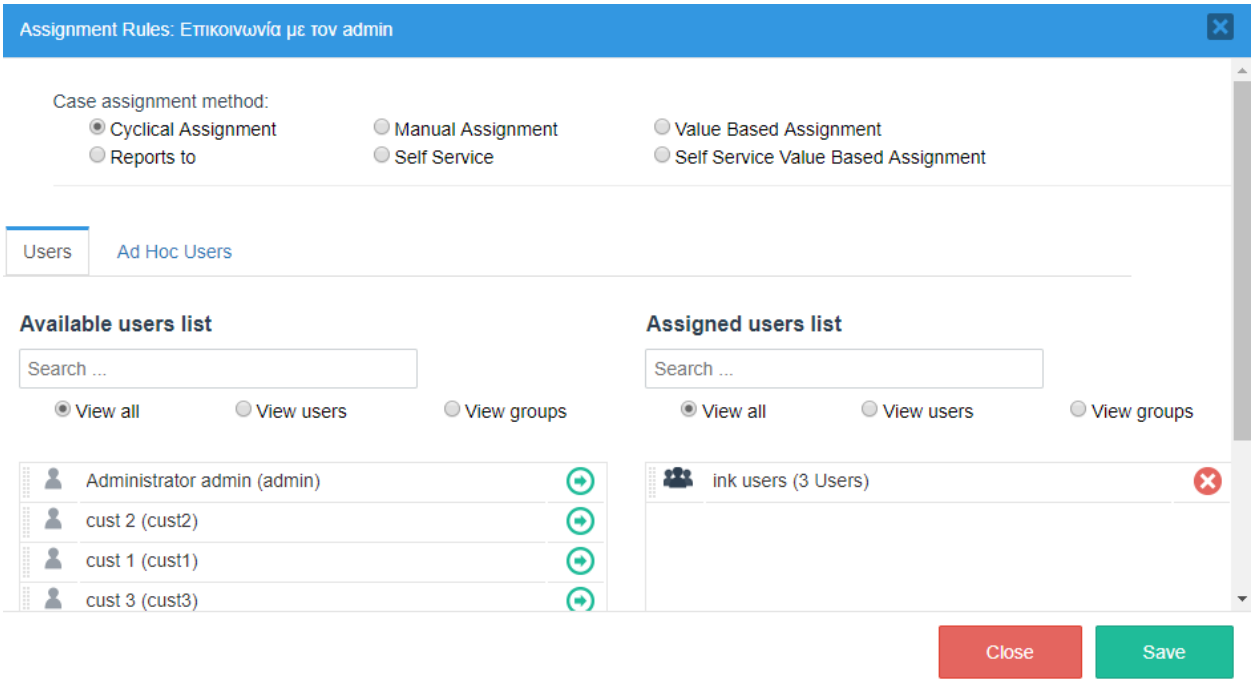


Figure 17 Assigning a task to a group of users.

Lastly, the “value based assignment” has to be explained. In the project, when the procurement process begins, the admin has to choose which users will place their orders (Figure 6). This is done by using the “allUsers” variable which quires the database so that all the available users are listed. Out of this list, the admin can select all users or some of them to continue with the process and claim the task “places his order”. For that reason, the selected users have to be passed in the variable “nextUsers” which is the variable that is used in the value based assignment. To pass the data from the “allUsers” variable to the “nextUsers” variable the following PHP code was created. After that, the “nextUsers” variable is used for the assignment as shown in Figure 19.

```
1.  @@nextUsers = array();
2.  $aUsers =  @@allUsers;
3.  for ($i = 0; $i < count($aUsers); $i++) {
4.    $user = @@allUsers[$i];
5.      if($user !== '1'){
6.        @@nextUsers[$i] = $user;
7.            $count++;
8.      }
```

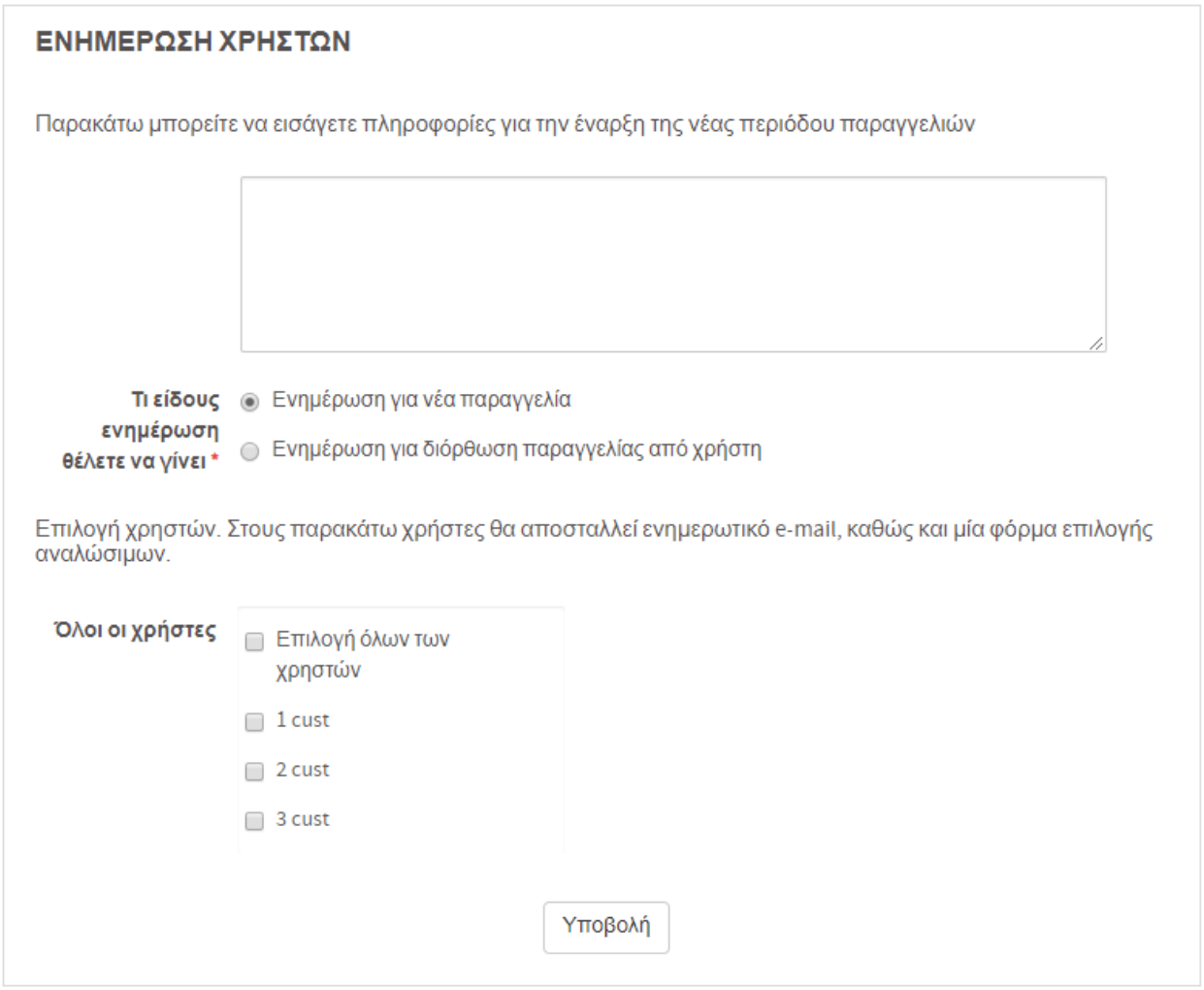


Figure 18 The admin's form where the users are chosen

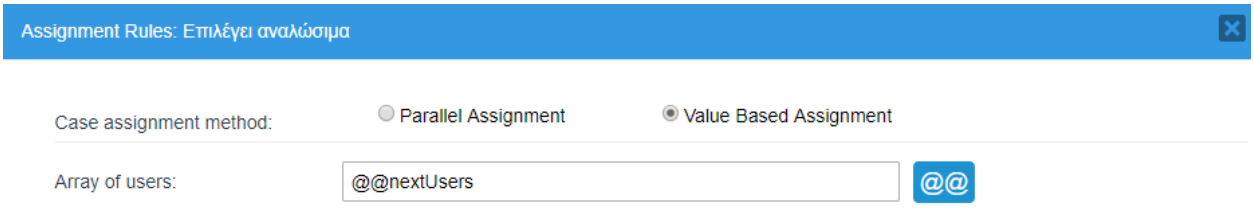


Figure 19 Value Based Assignment for the users that will place their order

The above assignment is made to enable users to place their orders. Before this, an email has to be sent, to inform the users that they may place their order. All emails are sent at once, so an assignment that will get the job done without the admin's or any other's user intervention is needed. This kind of assignment is the "self service value based assignment" (Figure 20), and it is used in the project so that the chosen users will automatically receive an informative email.

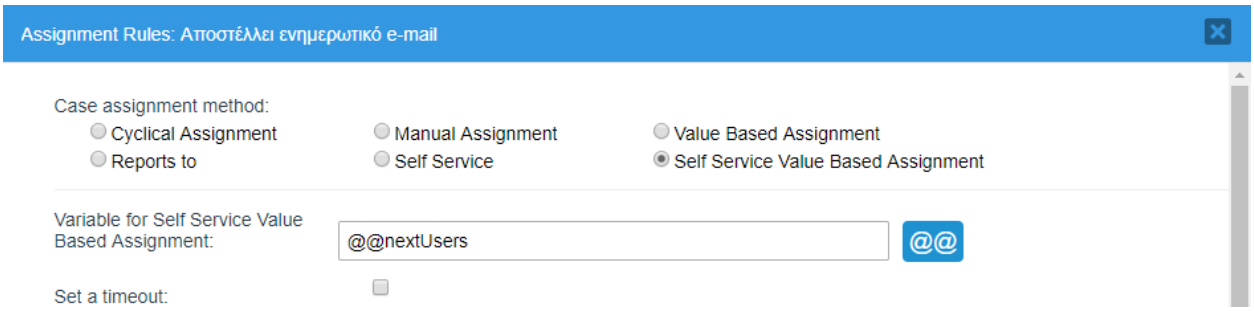


Figure 20 Self Service Based Assignment for automatically sending an email

6.3 CREATION OF A FORM

6.3.1 Available fields in forms

One of the most important things in a project like the ink BManager is the user interface. Also, it is undeniable that in projects like this, a large amount of information has to be handled. The way that this information pass from a user into the system -and vice versa- is by filling a form, or in ProcessMaker terms a ''Dynaform''.

Dynaforms enable the users to insert information into the system, or to view already stored info. This is done with the so-called "controls". ProcessMaker comes with two kinds of controls, the "web controls", where the variables are directly related to them, and all the other controls where the variables are not directly related to them and they are used to add more functionality to the forms.

Web controls may be "Textboxes" or "Textareas" where the user can enter either a text or a number, or "DateTime" where date and time are specified by the user from a pop-up calendar. Another web control is the "Dropdown menu" where many options are given to the user and he can select only one, or the "Checkgroup" where from the given options he may choose as many as he wants. Similarly to the "Dropdown menu", "Radio" buttons and "Checkboxes" can determine only one user's selection, and they are usually used to define the way that the flow of the process will move on. Furthermore, ProcessMaker supports the "Suggest" box, which is a combination of the "Dropdown menu" and the "Textbox", meaning that there is a list of possible choices and while the user enters his choice he can automatically view the options that match his insertion. The last web control that is supported is the "Hidden" control, where the hidden variable is stored so that the designer can handle information that it is not visible to the user.

Apart from the mentioned controls that handle variables, a form needs the ones that will make the interface easier and the form more attractive. For this reason, ProcessMaker provides controls such as "Title" and "Subtitle" or "Label" where the designer can give information to the users concerning a particular form. While providing this information, a link to an external site may be needed, and that is why the "Link" control is created. Also, images can be uploaded to the form with the "Image" control, and files may be attached with the "File" control.

One very useful control that was used many times in the project, is the "Grid" control. With a grid multiple data can be displayed with a table format, and also multiple data can be entered in the same variable by the user (e.g. a user may need more than one ink cartridge to be stored in the variable that handles his order). Grid is a quite flexible control since within it one may add "Textbox", "Textarea", "Dropdown", "Checkbox", "Datetime", "Suggest", "Hidden", "Link" or "File" controls, giving the designer the opportunity to achieve almost any desirable layout. Lastly, the user interface would be incomplete without the

buttons. ProcessMaker offers two kinds of buttons, the “Submit” where the form is submitted when clicked, and the “Button” control that can fire any action within a form (e.g. calculate the sum).

Below in figure 21, the fields that are used in the form where the users place their order are shown as an example. In the particular form three grids where used, there is one grid where the user can add ink cartridges in his list, one where he can insert new items that are not yet available and another one where already ordered items are viewed. In this example, almost every available control is used within the grids.

Furthermore, the grid where the user adds items that are not yet available is shown only if the user checks the “Checkbox” “You didn’t find the printer that you are looking for? Check this to add it”. In the case that this checkbox is checked, a “Textarea” control will show up and more information for the new insertion will be given. Also one can notice the two variables “resubmit” and “reopen”, those variables use “Text-box” control and they are used in a similar way as the “Hidden”, meaning that they are not visible to the user. Finally, the “Submit” button was added to enable the users to submit their form just with one click.

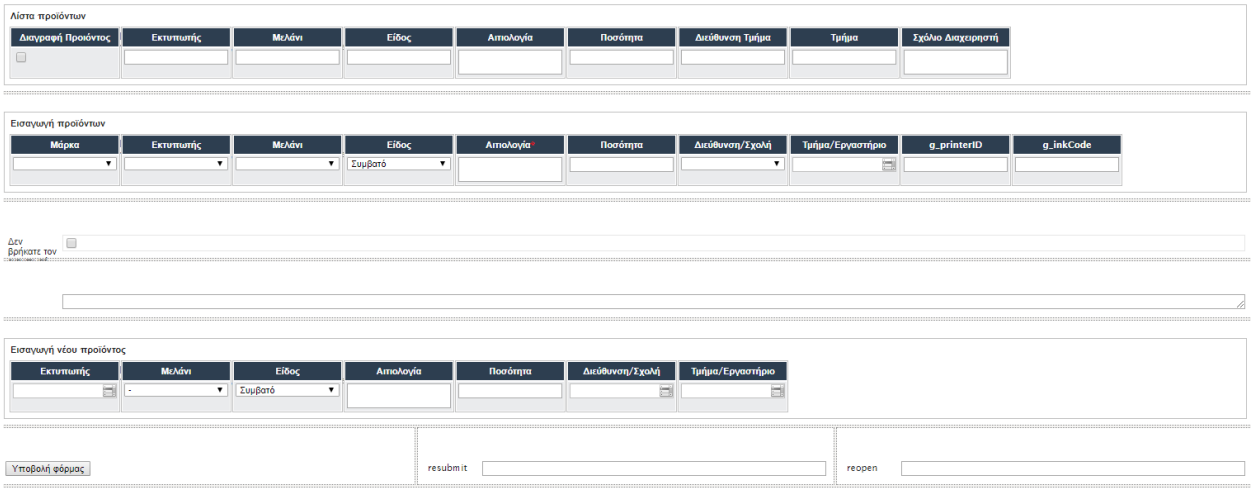


Figure 22 Example of “Grid”, “Textbox”, “Textarea”, “Checkbox” and “Submit” control (Designing Tool View)

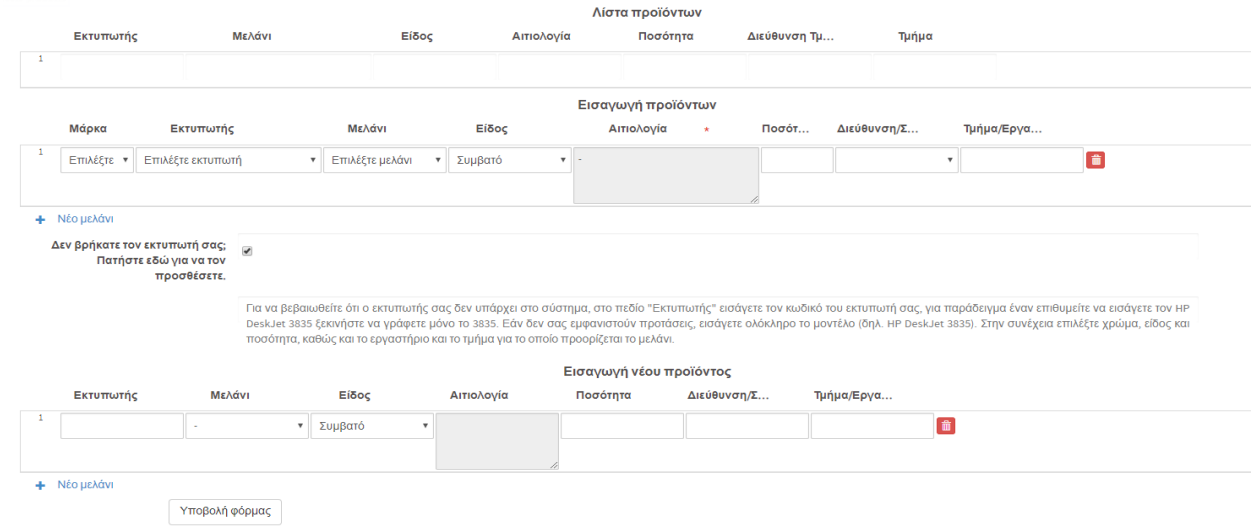


Figure 21 Example of “Grid”, “Textbox”, “Textarea”, “Checkbox” and “Submit” control (User Form View)

6.3.2 Handling variables within forms

Before discussing about handling variables within a form, it is of a major importance to clarify how ProcessMaker stores a variable. Each variable is stored as long as the process that it belongs to is running. While temporarily stored, a variable can be used any time in the process and it would be re-written if the user edits it. This way of storing restricts the creation of global variables (variables that can be used in many processes), so to overcome this problem one may store the variable in the database and retrieve it when needed.

Throughout the designing process a major endeavour is to handle the variables. ProcessMaker gives three options for doing so: the first option is to access them straight from the variable configuration field, the second option is to use JavaScript in the forms and the third one is to access the variables through triggers and PHP code. The first option will be discussed in this section, while the last two will be discussed later.

Many times a designer has to give options to the user, for example in the Ink BManager the user has to choose his printer from a "Dropdown" control. In this case, the available printers are stored in the database and they have to be extracted and displayed to the user. This is done with the SQL feature that can be found in the variable configuration field, and it is available only for Text, Textarea, Dropdown, Checkgroup, Radio, Suggest and Hidden controls.

Staying in the printers' example, the following code is used in the SQL field in order to display the available printers.

1. `SELECT ID_PRINTER, MODEL FROM `printers` WHERE BRAND = @@g_printerBrand ODER BY MODEL ASC`

One can notice the `@@g_printerBrand`, the double @ symbol in ProcessMaker is used to define a variable, so the above SQL query lists the available printers that match with a previously stored variable named `g_printerBrand`.

Aside from the SQL field, Dropdown, Checkgroup, Radio and Suggest control can display values that are given from the designer. This can be done by filling the "options" in the variable configuration field as shown in Figure 23.

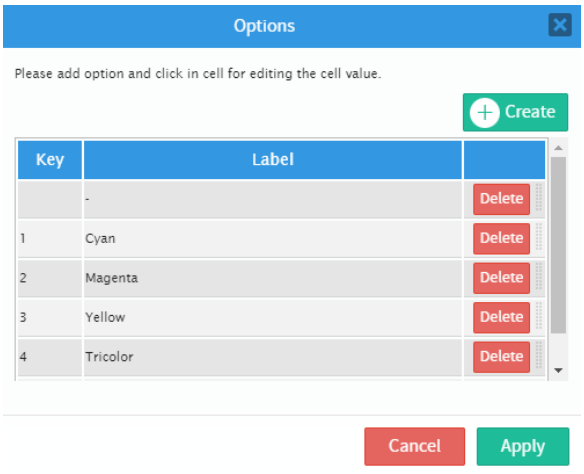


Figure 23 Populating a Dropdown control with options given by the designer

Lastly, many times the designer would have to fill a “Textbox” or a “Textarea” with a value retrieved from the database. For example in the following query that is written in the SQL field, fetches the ID of the last order that is saved in the table “orders”. Then this value is increased by one by using JavaScript, and this is one of the things that will be discussed in the next section.

```
SELECT ID_ORDER FROM orders ORDER BY ID_ORDER DESC
```

6.3.3 Using JavaScript within a form

ProcessMaker comes with an embedded JavaScript editor that can be found in every form. It is used to customize the behaviour of the Dynaform and add new features. In this section, many different examples of JavaScript usage in the Ink BManager will be given.

Two simple functions that were widely used in the project’s forms are the JQuery functions hide() and show(). These functions are used in order to dynamically hide or show any control and its label. For example the following code is used in the form where the user places his order, and reveals all the needed controls for adding a new printer when the user checks the “Checkbox” “You didn’t find the printer that you are looking for? Check this to add it”.

```
//////// HIDE/SHOW ADD ITEM GRID
$("#textareaVar001").hide();
$("#addGrid").hide();
$("#addToList").setOnchange( function(newVal, oldVal) {
    if (newVal == '"1"' || newVal == '["1"]') {
    $("#addGrid").show();
    $("#textareaVar001").show();
    }
    else {
    $("#addGrid").hide();
    $("#textareaVar001").hide();
    }
} );
```

JavaScript also can be used to edit some text or to style a field. This feature is used in the project in order to style the field where the directions are given so that a red border will be placed to draw the user’s attention. The above can be done using the .css() function as shown in the following code.

```
getFieldById("warningForReopening").$el.children().css("boder","3px solid #e8abab");
getFieldById("warningForResubmition").$el.chidren().css("boder","3px solid #e8abab");
```

Apart from the css() function, a designer may draw the user’s attention by giving directions and warnings through pop-up alerts within the browser. In Ink BManager every form has a pop-up alert which shows up when the “submit” button is clicked and waits for the user’s acceptance in order to submit the form. This is done by using the .confirm() function, but for simple warnings that do not wait for the user’s acceptance, the .alert() function can be used. Below there is an example that includes both functions.

```
var x= CheckMultipleEntries();
if  (x ===1 ) {
    alert("Έχετε επιλέξει ίδια προϊόντα. Παρακαλώ ελέγξτε την παραγγελία σας.");
    return false; //stop submit in case of multiple entries
}
if (confirm("Είστε σίγουροι ότι θέλετε να υποβάλετε την παραγγελία σας;")) {
    $("#4640808105bb7780d7c96f5044219023").submitForm();
```

```
8.   else {
9.     return false;
```

JavaScript can be used to manipulate variables within a form, giving the ability to validate a variable before the submission of a form. The following code includes a function which checks for multiple entries within a user's order. The function checks all the items that the user has inserted in his order, and if an item that is meant for the same department and address is found more than once, then an alert pops up stating that he must check his order for multiple entries.

```
1.  /////CHECK MULTIPLE ENTRIES
2.  function CheckMultipleEntries(){
3.      var ink;
4.      var type;
5.      var dieuth;
6.      var tmima;
7.      var inkArr = [];
8.      var typeArr= [];
9.      var dieuthArr = [];
10.     var tmimaArr  = [];
11.     var rows   = $("#grid").getNumberRows();
12.     for (var i=0 ; i<= rows  ; i++){
13.       ink        = $("#grid").getValue(i, 3)
14.       type       = $("#grid").getValue(i, 4)
15.       dieuth     = $("#grid").getValue(i, 7)
16.       tmima      = $("#grid").getValue(i, 8)
17.       inkArr[i]  = ink;
18.       typeArr[i] = type;
19.       dieuthArr[i] = dieuth;
20.       tmimaArr[i]= tmima;
21.     }
22.
23.     for(var i = 0; i<= rows; i++){
24.         for(var j = 0; j <= rows ; j++){
25.             if(i !== j){
26.                 if(inkArr[i] === inkArr [j] && typeArr[i] === typeArr [j] && dieuthArr[
    i]===dieuthArr[j] && tmimaArr[i]===tmimaArr[j] ){
27.                   //alert('You have selected the same ink');
28.                   return 1;
29.                 }
30.             }
31.         }
32.     }
33. }
```

Continuing, JavaScript makes possible the automatic insertion of a value into a field. The following code inserts into the variable "DEL_DELEGATE_DATE" the current date and time.

```
1.   var d = new Date();
2.   dformat = [ d.getFullYear(),
3.               d.getMonth() +1,
4.               d.getDate()
5.               ].join('-')+' '+
6.              [convertHours(),
```

```
                d.getMinutes() ,
                d.getSeconds()].join(':');
$("#DEL_DELEGATE_DATE").setValue(dformat);
```

Since there is the ability to manipulate a variable in such way, it is apparent that JavaScript offers the ability to make calculations between the variables. In the project there is a field where all the items ordered by a user are listed. That list includes the price, and the total expense has to be calculated. This is done using the following code.

```
//////// GET THE SUM OF THE INK PRICES
$("#button0000000001").click(function(){
  var totalRows = $("#consumptionGrid").getNumberRows();
  var sum = 0;
   for (var i = 1; i <= totalRows; i++) {
       var quantity    = parseFloat( $("#consumptionGrid").getValue(i, 6) );
       var price       = parseFloat( $("#consumptionGrid").getValue(i, 7) );
       sum             = sum + quantity*price ;
 }
  $("#sum").setValue(sum.toFixed(2));
} );
```

Lastly, since grids are often used in the project, there is a need to manage grid fields. One example of grid manipulation is when a user fills his order form, if he chooses original ink cartridge he has to give a justification, otherwise if he chooses compatible ink cartridge he can place his order without justifying his selection. For that reason the field “reason” can be accessed depending on the user’s selection. The following piece of code dynamically hides and shows the mentioned field.

```
//////////////// HIDE/SHOW REASON
function EnableDisableReason(gridId, gridField, row, eidos, aitiologia){
   $("#form\\["+gridId+"\\]\\[" + row + "\\]\\["+ gridField +"\\]").change(function(){
      if($('#'+gridId).getValue(row,eidos) == 0){
         $('#'+gridId).getControl(row,aitiologia).attr("disabled", false);
         $('#'+gridId).setValue('' , row, aitiologia );
      }else{
         $('#'+gridId).getControl(row,aitiologia).attr("disabled", true);
         $('#'+gridId).setValue('-', row, aitiologia );
      }

   })
}

//first row config
$("#grid").getControl(1,5).attr("disabled", true);
$("#addGrid").getControl(1,4).attr("disabled", true);
EnableDisableReason('grid', 'g_userInkState', 1, 4,5);
EnableDisableReason('addGrid','g_addInkState',1, 3,4);

//Every other row config in Grid
$("#grid").onAddRow( function(aRow, oGrid, row){
   $("#grid").getControl(row,5).attr("disabled", true);
   EnableDisableReason('grid', 'g_userInkState', row, 4,5)
});
```

```
26.
27. //Every other row config in addGrid
28. $("#addGrid").onAddRow( function(aRow, oGrid, row){
29.     $("#addGrid").getControl(row,4).attr("disabled", true);
30.     EnableDisableReason('addGrid', 'g_addInkState', row, 3,4)
31. });
```

6.4 ACCESSING THE DATABASE

One of the main advantages of ProcessMaker is the fact that phpMyAdmin is automatically installed, meaning that by the time the installation has finished the designer can easily access any database or table that is used by the system. The databases can be accessed through command window or through phpMyAdmin graphical environment.

The way that ProcessMaker and its forms communicate with the database is through triggers, using the PHP function executeQuery(). Triggers are pieces of PHP code that enable the designer to perform complex tasks and handle variables throughout the processes. Triggers can be fired before or after a step is executed, before or after a user is assigned on a task, and before or after a case is routed to the next task. In this section there will be given examples of trigger usage in the Ink BManager, covering most of the cases that a trigger can be used.

6.4.1 Storing data entered in a form

As stated earlier ProcessMaker holds a variable as long as the process that it belongs to is running. To avoid data loss and make sure that variables values can be retrieved any time, PHP triggers that save the data in the database should be fired. Those triggers can be fired “after Dynaform” or “after Routing”. In the first case, the variables that are used in a Dynaform are stored directly after the submission of a form, meaning that if the user selects to edit again his form the variable will be saved twice. That may be useful in some cases but in Ink BManager only the last user’s selection is meeded, so this kind of triggers is fired “After routing”.

Before continuing, the basic way that triggers are structured has to be explained. Firstly a database connection has to be established. This is done on the main screen of the process by choosing “Database Connections”. Then the connection has to be configured as shown in Figure 24.

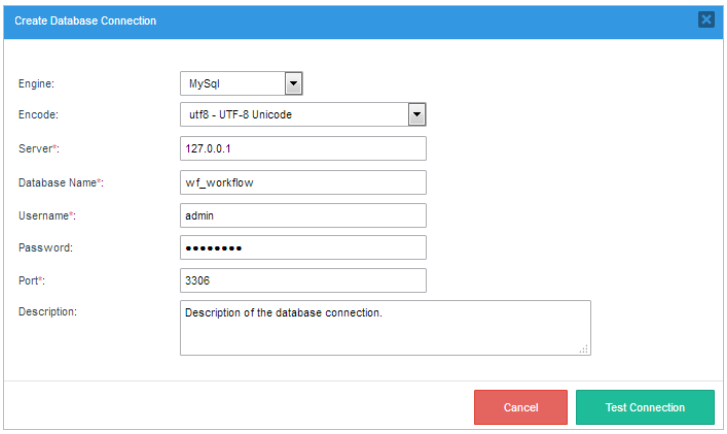


Figure 24 Configuring the database connection.

As long as the connection is established, one can find the ID of this connection. Then this ID is used in the executeQuery() function. A very simple example is the following code, where a new printer’s brand is

inserted in the table `'printers_brand'` by using the executeQuery() with first argument the wanted SQL query and second argument the ID of the database connection.

```
$dbConnection = '2857666015bbbad94e032e0030237549';
$tableName1   = 'printers_brand';
$sql1 = "INSERT INTO $tableName1 (BRAND)  VALUES ('$brand_label')";
executeQuery($sql1,  $dbConnection);
```

So in order to execute a trigger that will store any value in the database three things are needed, the ID of the database connection, the SQL query and the executeQuery() function. Bellow, a more complicated example is given. The trigger is set to fire after routing the task “places his order” in process of “Ordering ink cartridges” and is used to save the items that each user has entered in his order form.

```
$dbConnection = '5349426265ba0e8c982d332062247617'; //set to ID of the DB Connection
$tableName1 = 'user_order';

$aUser = PMFInformationUser(@@USER_LOGGED);
$username= $aUser['username'];

//write grid to the table:
foreach (@@grid as $row) {
   $id                   = $row['ID'];
   $g_userID             = $username;
   $g_tmima              = $row['g_tmima'];
   $g_tmima_label        = $row['g_tmima_label'];
   $g_userOffice         = $row['g_userOffice'];
   $g_userOffice_label   = $row['g_userOffice_label'];
   $g_userPrinter        = $row['g_userPrinter'];
   $g_userPrinter_label  = $row['g_userPrinter_label'];
   $g_userInk            = $row['g_userInk'];
   $g_inkCode            = $row['g_inkCode'];
   $g_userInkState       = $row['g_userInkState'];
   $g_userInkState_label = $row['g_userInkState_label'];
   $g_userReason         = $row['g_userReason'];
   $g_quantity           = $row['g_quantity'];

   if (empty($id) and $id !== '0' and $id !== 0) {
      $sql1 = "INSERT INTO $tableName1 (g_userID, g_userOffice, g_tmima,
   g_userPrinter, g_printerName, g_userInk, g_inkName, g_userInkState,
   g_userReason, g_quantity) ".
         "VALUES ('$g_userID', '$g_userOffice_label', '$g_tmima_label',
   '$g_userPrinter', '$g_userPrinter_label', '$g_userInk', '$g_inkCode',
   '$g_userInkState_label', '$g_userReason', '$g_quantity')";
      executeQuery($sql1,  $dbConnection);
   }
}
```

In section 5.3.2 it was shown how to use the SQL field in the variable configuration field in order to give values in a variable. The mentioned way is indeed a good way to populate a variable, but in some cases it is handier to use triggers. The basic way to populate a variable is shown below. Triggers like this are always set to fire “Before Dynaform” since the form will fetch data from the executed query.

```
$db = '9711608885bcdfcf7116767007692818';
//Populate 'grid' with products
$query = 'SELECT PRODUCT, TOTAL_sumvata, TOTAL_gnhsia FROM submited_order';
$result = executeQuery($query, $db);
if (is_array($result) and count($result) > 0)
    @=grid = $result;
```

Consequently, in order to store data in a database the “INSERT INTO” SQL statement is used, while to display data the “SELECT” statement is used.

6.5 PROCESSMAKER TROUBLESHOOTING

6.5.1 Parallel tasks

In the process of “ordering ink cartridges” there is a parallel task where users can place their order. This parallel task allows all the users to fill their forms and the flow does not move on until every user completes his order. In many cases, this way of directing the flow is useful since it forces all users to complete one task, but in the Ink BManager this is not the case. From past procurements it is known that not every user places his order even if he is constantly reminded to do so. So the fact that the flow would not move on is an issue that has to be solved. To overcome this issue the process of “Canceling a user’s order” was designed. This process can be used either to cancel a single user’s form or to cancel all open forms so that the flow can move on.

In ProcessMaker all the running and finished tasks are stored in the table “app_delegation”, where every task is represented by an entry (a row). Each entry provides information about the above tasks and one of them is the “DEL_THREAD_STATUS” where every task has a “CLOSED” or “OPEN” status. If the status is “CLOSED” it means that the task has been claimed and completed, whereas the “OPEN” status indicates that one task has to be claimed. By changing the “DEL_THREAD_STATUS” to “CLOSED” the designer can force one task to end, and this is exactly what happens when the admin selects to close one user’s form. At this point, one may think that by just forcing every user’s order form to close, the flow will move on, but things are not that simple.

Given that the “app_delegation” table has entries only for the running and finished tasks, the problem with the flow arises from the fact that the next task does not have an entry yet. In every other case, when a task gets the “CLOSED” status, the next task gets automatically an entry with an “OPEN” status allowing the user to claim it. This does not happen with the parallel tasks, so when the process has to move from the task “places his order” to the “checks the order”, the entry for the “checks the order” has to be inserted in the table with an “OPEN” status. As stated, this insertion is not automatic, so the whole process is designed in order to provide all the data for the columns needed for the manual insertion.

The manual insertion is done by using a trigger with the following PHP code, which executes an “INSERT INTO” SQL statement in order to write into the “app_delegation” table.

```
$sql = "INSERT INTO app_delegation (APP_UID,DEL_INDEX,DELEGATION_ID,APP_NUMBER,DEL_PRE-
VIOUS,DEL_LAST_INDEX,PRO_UID,TAS_UID,USR_UID,DEL_TYPE,DEL_THREAD,DEL_THREAD_STA-
TUS,DEL_PRIORITY,DEL_DELEGATE_DATE,DEL_DURATION,DEL_QUEUE_DURATION,DEL_DELAY_DURA-
TION,DEL_STARTED,DEL_FINISHED,DEL_DELAYED,DEL_DATA,APP_OVERDUE_PERCENT-
AGE,USR_ID,PRO_ID,TAS_ID) VALUES ('$APP_UID','$DEL_INDEX','$DELEGATION_ID', '$APP_NUM-
BER', '$DEL_PREVIOUS', '$DEL_LAST_IN-
DEX', '$PRO_UID', '$TAS_UID', '$USR_UID', '$DEL_TYPE', '$DEL_THREAD', '$DEL_THREAD_STA-
```

```
TUS', '$DEL_PRIORITY', '$DEL_DELEGATE_DATE', '$DEL_DURATION', '$DEL_QUEUE_DURA-
TION' , '$DEL_DELAY_DURATION' ,'$DEL_STARTED', '$DEL_FINISHED', '$DEL_DE-
LAYED', '$DEL_DATA', '$APP_OVERDUE_PERCENTAGE', '$USR_ID', '$PRO_ID', '$TAS_ID')";

if(@@close == 1 or @@close == '1'){
@@result = executeQuery($sql, $db);
```

Also the following code is used for closing one user's form. As shown, in this trigger an "UPDATE" SQL statement is executed in order to change the "OPEN" status into "CLOSED" status.

```
$db = '6825866695bb3a37647ffe4057754332';

@@close = 0;
$numOfSelectedUsers = 0;

$sql1  =  "SELECT COUNT(USR_UID)
FROM app_delegation
WHERE DEL_THREAD_STATUS = 'OPEN'
AND TAS_UID = '4054565425c9b49347849d0068107943'"; ///uid sto 'epilegei analwsima'
$result = executeQuery($sql1, $db);
$numOfAllUsers = $result[1]['COUNT(USR_UID)'];

$app_uid  = @@APP_UID;
$allUsers = @@checkgroup;
for ($i = 0; $i < count($allUsers); $i++) {
   $user = @@checkgroup[$i];
   $sql = "UPDATE app_delegation
           SET DEL_THREAD_STATUS = 'CLOSED'
           WHERE USR_UID = '$user' AND APP_UID = '$app_uid'";

    if($user !== '1'){
    executeQuery($sql, $db);
    $numOfSelectedUsers++;
   }
}

if( $numOfSelectedUsers == $numOfAllUsers){
  @@close = 1;
}
```

Another problem that arises from the usage of a parallel task, is the fact that the variables that are used in this task's form are overwritten every time a user submits his form. To avoid overwriting a variable when submitting a form, the values of the variables have to be saved in a table and then to be cleared. If the variables are not cleared, the next user who will fill his form -and consequently will use those variables- will get a form where the choices of the previous user are visible.

To avoid the above situation a trigger has to be fired "before Dynaform" in order to clear variables that are used in the particular form. For example, the next code is used to clear the variables that are used in the user's order form.

```
//populate grid
@@grid = array(
```

```
    '1' => array('g_printerBrand'=>' ', 'g_userPrinter'=>' ', 'g_userInk'=>' ', 'g_userInkState'=> array(0, 'Συμβατό'), 'g_userReason'=>'', 'g_quantity'=>'', 'g_userOffice'=> ' ', 'g_tmima'=>' ', 'g_printerID'=> '', 'g_inkCode'=>' ' )
);


@@notApprovedOrdersGrid = array(
    '1' => array('office'=>' ', 'printer'=>' ', 'ink'=>' ', 'inkState'=>' ', 'quantity'=>' ', 'reason'=>' ')
);
```

6.5.2 Reopening a user's form

Before starting designing the processes, a discussion was made with the people in charge of the procurements in the Technical University of Crete. It was noted that in the past procurements the users usually made mistakes and they had to contact them in order to change their order. Taking that into consideration, the process of "reopening a case" was created. In order to reopen a user's form, firstly the user has to send an email to the admin, stating that he wishes to change his order, and then when the admin receives the email he opens the form.

When the admin runs the process "reopening a case" he receives a form with a list of the users that are assigned to the task "places his order" of the process "ordering ink cartridges". Out of this list, he selects which user will have his form reopened. The forms are opened by using a trigger and the functions `ReactivateCurrentDelegation()` and `updateCase()`.

```
if (isset(@=caseList) and !empty(@=caseList)) {
   $reopenedCount = 0;
   foreach (@=caseList as $aCase) {
      $reopen   = $aCase[Reopen];
     if ($reopen=='1' or $reopen ==1){
      $caseId = $aCase['caseId'];
      $index  = $aCase['index'];
      $userId = $aCase['userId'];
      $c = new Cases();
      $result = $c->ReactivateCurrentDelegation($caseId, $index);

      $aCaseLoaded = $c->loadCase($caseId);
      $aCaseLoaded['APP_STATUS'] = 'TO_DO';
      $c->updateCase($caseId, $aCaseLoaded);
      $reopenedCount++;
     }

   }
  $g = new G();
  $g->SendMessageText("$reopenedCount cases were reopened.", "INFO");
}
```

When the form is reopened the user would view his form as if he wants to place a new order, but the point of the whole process is to enable the user to view and edit his past order. To make that happen, the information that a user has his form reopened, should pass from the process of "reopening a case" to the process of "ordering ink cartridges". In the section 6.3.2 it was noted that ProcessMaker does not support

variables that can be shared between processes, so to bypass this problem a new table was created in the database, where the information that one form is reopened is stored.

More specifically, in the table “reopen”, the username of the user that needs to edit his order is stored, together with a variable valued with 1 which is used as a flag. So before opening a user’s form, the system will search for this flag and if it is valued as 1 then the form will be populated with the past order. This is done with the following trigger that is set to fire before the Dynaform of the task “places his order”.

```
$db    = '5349426265ba0e8c982d332062247617';
$aUser = PMFInformationUser(@@USER_LOGGED);
$user  = $aUser['username'];

$reopen = executeQuery("SELECT * FROM reopen WHERE user = '$user'",$db);

if(!empty($reopen)){
        @@reopen = $reopen[1]['reopen'];
    $query = "SELECT g_userOffice AS office ,g_tmima AS tmima, g_printer-
Name AS printer,g_inkName AS  ink,g_userInkState AS inkState, g_userReason AS rea-
son,g_quantity AS quantity FROM user_order WHERE g_userID = '$user'
UNION
SELECT g_addOffice AS of-
fice , g_addTmima AS tmima,g_addPrinter AS printer,g_addInk AS ink,g_addInk-
State AS inkState, g_addReason AS reason,g_addQuantity AS quantity FROM addgrid
WHERE g_addUserID = '$user' ";
    $result = executeQuery($query, $db);
    if (is_array($result) and count($result) > 0)
        @=notApprovedOrdersGrid  = $result;
}
```

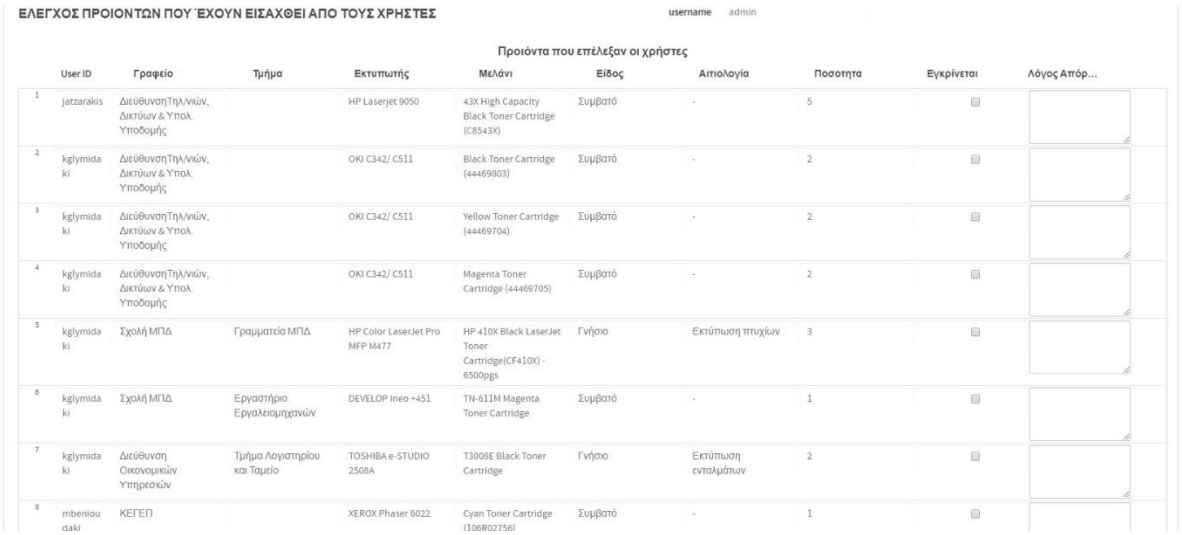
7 RESULTS AND RECOMMENDATIONS

7.1 PILOT TESTING THE INK BMANAGER

To ensure that Ink BManager will meet the expectations as a real-world project, a pilot test was conducted between the members of the Technical University of Crete. The test was conducted in collaboration with the Department of Administrative Computer Infrastructure, where the person in charge of the procurements was given the role of the admin. Also, twenty members of the university’s community participated as users. The test had a duration of ten days, where the users were able to place their orders. In addition, two meetings with the administrator were needed, one for the training and one for the testing of all the processes that do not require the users’ participation.

The main process that had to be tested is the process of “ordering ink cartridges”. Firstly the admin chose the thirty persons who would test the platform, and automatically an email was sent to them, inviting them to take part in the test. During the test, all possible scenarios where examined. Firstly, two users stated that they needed to change their order, so two users tested the process of “Contacting the admin” and since the forms of the users had to be reopened, the process of “Reopening a case” was tested too. Also, to the mentioned two users, their already placed order was displayed in their form so that they can edit it.

As expected, after ten days, not every user placed his order. In fact, only twenty users filled their form, so the admin had to close the ten open forms and move on to the task where the orders are checked. At this phase, the process of “Canceling a user’s order” was used to close the users’ forms in order to move on the flow. Once the flow moved on, the form where the admin could view the users’ orders showed up and he had to approve or not the selected users’ items. In Figure 25 there is a part of this form, while in Figure 26 there is the part where the printers that users could not find in the list are listed.



ΕΛΕΓΧΟΣ ΠΡΟΙΟΝΤΩΝ ΠΟΥ ΈΧΟΥΝ ΕΙΣΑΧΘΕΙ ΑΠΟ ΤΟΥΣ ΧΡΗΣΤΕΣ

username admin

Προιόντα που επέλεξαν οι χρήστες

	User ID	Γραφείο	Τμήμα	Εκτυπωτής	Μελάνι	Είδος	Αιτιολογία	Ποσότητα	Εγκρίνεται	Λόγος Απόρ...
1	jatzarakis	ΔιεύθυνσηΤηλ/νικών, Δικτύων & Υποδ. Υποδομής		HP Laserjet 9050	43X High Capacity Black Toner Cartridge (C8543X)	Συμβατό	-	5		
2	kglymidaki	ΔιεύθυνσηΤηλ/νικών, Δικτύων & Υποδ. Υποδομής		OKI C342/ C511	Black Toner Cartridge (44469803)	Συμβατό	-	2		
3	kglymidaki	ΔιεύθυνσηΤηλ/νικών, Δικτύων & Υποδ. Υποδομής		OKI C342/ C511	Yellow Toner Cartridge (44469704)	Συμβατό	-	2		
4	kglymidaki	ΔιεύθυνσηΤηλ/νικών, Δικτύων & Υποδ. Υποδομής		OKI C342/ C511	Magenta Toner Cartridge (44469705)	Συμβατό	-	2		
5	kglymidaki	Σχολή ΜΠΔ	Γραμματεία ΜΠΔ	HP Color LaserJet Pro MFP M477	HP 410X Black LaserJet Toner Cartridge(CF410X) - 6500pgs	Γνήσιο	Εκτύπωση πτυχίων	3		
6	kglymidaki	Σχολή ΜΠΔ	Εργαστήριο Εργαλειομηχανών	DEVELOP Ineo +451	TN-611M Magenta Toner Cartridge	Συμβατό	-	1		
7	kglymidaki	Διεύθυνση Οικονομικών Υπηρεσιών	Τμήμα Λογιστηρίου και Ταμείο	TOSHIBA e-STUDIO 2508A	T3008E Black Toner Cartridge	Γνήσιο	Εκτύπωση ενταλμάτων	2		
8	mbeniou daki	ΚΕΓΕΠ		XEROX Phaser 6022	Cyan Toner Cartridge (106R02756)	Συμβατό	-	1		

Figure 25 Users' orders that await admin's approval

Προιόντα που επέλεξαν οι χρήστες και δεν υπάρχουν στην βάση δεδομένων

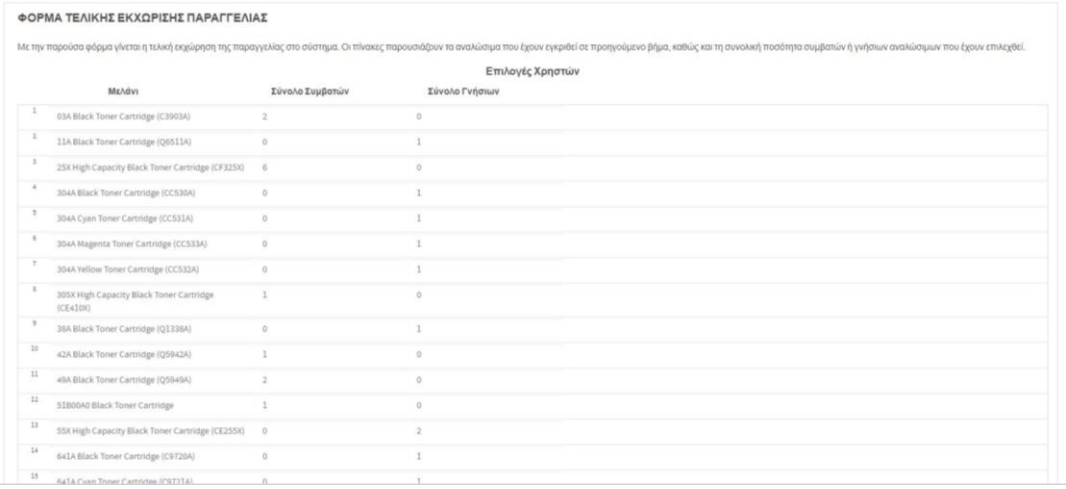
	User ID	Γραφείο	Τμήμα	Εκτυπωτής	Μελάνι	Είδος	Αιτιολογία	Ποσότητα	Εγκρίνεται	Λόγος Απόρ...
1	kglymidaki	Διεύθυνση Διοικητικών Υπηρεσιών	Τμήμα Διεκπεραίωσης και Αρχείου	HP LaserJet 4300n	Black	Συμβατό	-	4		
2	mntountounakis	ΗΜΜΥ	ΑΥΤΟΜΑΤΙΣΜΟΥ	HP Color LaserJet Pro MFP M27	Black	Συμβατό	-	1		
3	mntountounakis	Σχολή ΗΜΜΥ	ΑΥΤΟΜΑΤΙΣΜΟΥ	HP Color LaserJet Pro MFP M277	Cyan	Συμβατό		1		
4	mntountounakis	Σχολή ΗΜΜΥ	ΑΥΤΟΜΑΤΙΣΜΟΥ	HP Color LaserJet Pro MFP M277	Magenta	Συμβατό		1		
5	mntountounakis	Σχολή ΗΜΜΥ	ΑΥΤΟΜΑΤΙΣΜΟΥ	HP Color LaserJet Pro MFP M277	Yellow	Συμβατό		1		
6	mlagoudakis	Σχολή ΜΠΔ		XEPOX phaser 6100	Yellow	Συμβατό	-	1		

Figure 26 Users' orders that include items that were not found in the list of the available ink cartridges

The second scenario that was tested, was the one where the admin does not approve a user’s order. In this scenario the admin rejected some items from the order of two users. As shown in Figure 6, when an item is rejected the flow can be redirected to the first task, so that the users could edit their order. In the first task, the admin had to choose which users who have rejected items would fill their form once again. The users were selected and one email stating that they need to edit their order was sent, while their form was opened again. This time, the users’ form had a special field where the rejected items were listed, so that the users would change their order following the admin’s comments. While the admin waited for

the users to edit their orders, a user asked to edit his order so that he can add a new item. In that case, the admin executed the process of “Reopening a case” and the user edited his order. By the time this user submitted his new order, the users with the rejected items placed their orders too. Since all the users have completed their task, the flow was once again directed to the task “checks the order”, where the admin reviewed all the orders (including the new ones) and selected to move on to the final “saves the order in the database “ task.

The third scenario that was tested, was the one where the admin had to manage the items that were not in the list and the users added them manually. In this case, the process of “adding a new printer/ product or a new department/ office” was started, and the admin inserted those items in the database. Following this, the admin claimed the task “saves the order in the database“ of the process of “ordering ink cartridges” and an aggregate list of all the ordered items automatically showed up. A part of that list can be seen in Figure 27.



ΦΟΡΜΑ ΤΕΛΙΚΗΣ ΕΚΧΩΡΙΣΗΣ ΠΑΡΑΓΓΕΛΙΑΣ

Με την παρούσα φόρμα γίνεται η τελική εκχώρηση της παραγγελίας στο σύστημα. Οι πίνακες παρουσιάζουν τα αναλώσιμα που έχουν εγκριθεί σε προηγούμενο βήμα, καθώς και τη συνολική ποσότητα συμβατών ή γνήσιων αναλώσιμων που έχουν επιλεχθεί.

Επιλογές Χρηστών

	Μελάνι	Σύνολο Συμβατών	Σύνολο Γνήσιων
1	03A Black Toner Cartridge (C3903A)	2	0
2	11A Black Toner Cartridge (Q6511A)	0	1
3	25X High Capacity Black Toner Cartridge (CF325X)	6	0
4	304A Black Toner Cartridge (CC530A)	0	1
5	304A Cyan Toner Cartridge (CC531A)	0	1
6	304A Magenta Toner Cartridge (CC533A)	0	1
7	304A Yellow Toner Cartridge (CC532A)	0	1
8	305X High Capacity Black Toner Cartridge (CE410X)	1	0
9	38A Black Toner Cartridge (Q1338A)	0	1
10	42A Black Toner Cartridge (Q5942A)	1	0
11	49A Black Toner Cartridge (Q5949A)	2	0
12	51B00A0 Black Toner Cartridge	1	0
13	55X High Capacity Black Toner Cartridge (CE255X)	0	2
14	641A Black Toner Cartridge (C9720A)	0	1
15	641A Cyan Toner Cartridge (C9721A)	0	1

Figure 27 Aggregate list of all ordered items

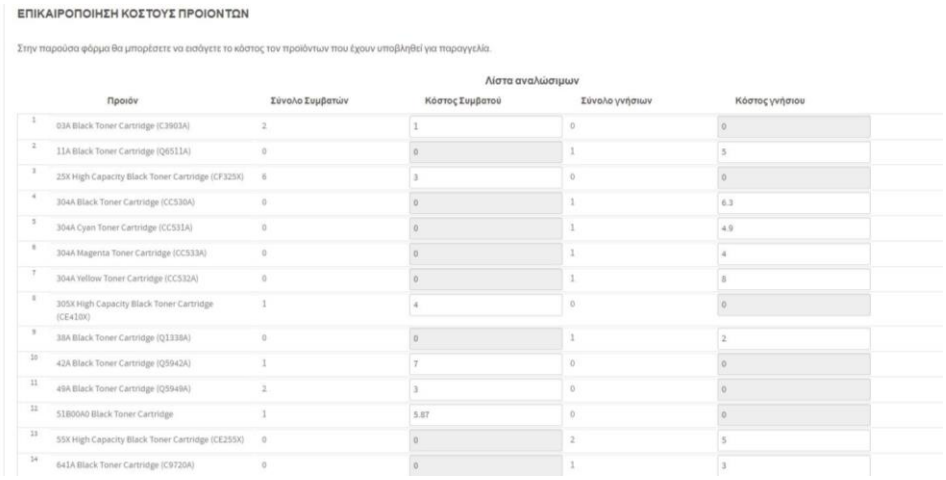
With this list, the process of “ordering ink cartridges” is completed. During the whole process four more processes were used. As described, the “cancelling a user’s order”, the “contacting the admin”, the “reopening a case” and the process of “adding a new printer/ product or a new department/ office” were also completed in order to assist this main process. But except for supporting the process where users can place their order, Ink BManager offers some processes that can assist the admin in the whole procurement process.

The first process to be completed after the extraction of the aggregate list of the ordered items is the process of “budget creation after price comparison”. In this process, the admin can enter the prices of the ordered products, as these are found in the market. Those prices will be used later, in order to compare the prices given by the suppliers. A part of the list with the products and their prices is shown in Figure 28.

Similarly to the above process, the process of “updating the cost” is used to store the prices of the ordered products. This time the prices are the ones that the supplier sets and those that would be later viewed in the record. The form that is filled in this process is the one shown in Figure 27.

Another process to be completed is the process of “viewing the record”. Before starting designing, it was made clear by the people in charge, that there is a need for a process where the past orders can be viewed. In Figure 28 the record of the order that was placed during the pilot testing is shown. The form lists all the items that the users have ordered, the corresponding prices, and the sum of the cost so that the admin can keep track of the expenses.

Lastly, the process of “contacting the users” was tested too, since the platform was used in order to send emails to the participants, asking them to fill the questionnaire that would be later used in the satisfaction survey.



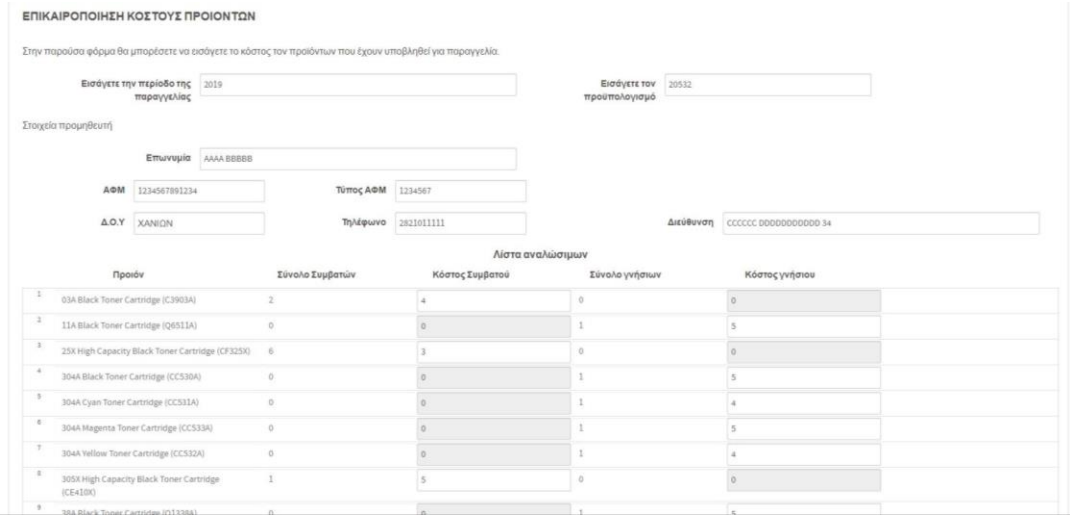
ΕΠΙΚΑΙΡΟΠΟΙΗΣΗ ΚΟΣΤΟΥΣ ΠΡΟΙΟΝΤΩΝ

Στην παρούσα φόρμα θα μπορέσετε να εισάγετε το κόστος τον προϊόντων που έχουν υποβληθεί για παραγγελία.

Λίστα αναλώσιμων

	Προϊόν	Σύνολο Συμβατών	Κόστος Συμβατού	Σύνολο γνήσιων	Κόστος γνήσιου
1	03A Black Toner Cartridge (C3903A)	2	1	0	0
2	11A Black Toner Cartridge (Q6511A)	0	0	1	5
3	25X High Capacity Black Toner Cartridge (CF325X)	6	3	0	0
4	304A Black Toner Cartridge (CC530A)	0	0	1	6.3
5	304A Cyan Toner Cartridge (CC531A)	0	0	1	4.9
6	304A Magenta Toner Cartridge (CC533A)	0	0	1	4
7	304A Yellow Toner Cartridge (CC532A)	0	0	1	8
8	305X High Capacity Black Toner Cartridge (CE410X)	1	4	0	0
9	38A Black Toner Cartridge (Q1338A)	0	0	1	2
10	42A Black Toner Cartridge (Q5942A)	1	7	0	0
11	49A Black Toner Cartridge (Q5949A)	2	3	0	0
12	51B00A0 Black Toner Cartridge	1	5.87	0	0
13	55X High Capacity Black Toner Cartridge (CE255X)	0	0	2	5
14	641A Black Toner Cartridge (C9720A)	0	0	1	3

Figure 28 Prices of the ordered products, as found in the market



ΕΠΙΚΑΙΡΟΠΟΙΗΣΗ ΚΟΣΤΟΥΣ ΠΡΟΙΟΝΤΩΝ

Στην παρούσα φόρμα θα μπορέσετε να εισάγετε το κόστος τον προϊόντων που έχουν υποβληθεί για παραγγελία.

Εισάγετε την περίοδο της παραγγελίας: 2019

Εισάγετε τον προϋπολογισμό: 20532

Στοιχεία προμηθευτή

Επωνυμία: AAAA BBBBB

ΑΦΜ: 123456789234

Τύπος ΑΦΜ: 1234567

Δ.Ο.Υ: ΧΑΝΙΩΝ

Τηλέφωνο: 2821011111

Διεύθυνση: CCCCCC DDDDDDDDDD 34

Λίστα αναλώσιμων

	Προϊόν	Σύνολο Συμβατών	Κόστος Συμβατού	Σύνολο γνήσιων	Κόστος γνήσιου
1	03A Black Toner Cartridge (C3903A)	2	4	0	0
2	11A Black Toner Cartridge (Q6511A)	0	0	1	5
3	25X High Capacity Black Toner Cartridge (CF325X)	6	3	0	0
4	304A Black Toner Cartridge (CC530A)	0	0	1	5
5	304A Cyan Toner Cartridge (CC531A)	0	0	1	4
6	304A Magenta Toner Cartridge (CC533A)	0	0	1	5
7	304A Yellow Toner Cartridge (CC532A)	0	0	1	4
8	305X High Capacity Black Toner Cartridge (CE410X)	1	5	0	0
9	38A Black Toner Cartridge (Q1338A)	0	0	1	5

Figure 29 Prices of the products as set from the supplier and supplier’s information

ΠΡΟΒΟΛΗ ΕΠΙΛΟΓΩΝ ΤΩΝ ΧΡΗΣΤΩΝ

Επιλογές χρηστών

	User name	Διεύθυνση...	Τμήμα/ Ερ...	Προϊόν	Είδος	Ποσότητα	Τιμή
1	epetraki	ΔιεύθυνσηΤηλ/νιών, Δικτύων & Υπολ. Υποδομής	Τμήμα Τηλεπικοινωνιών και Δικτύων	03A Black Toner Cartridge (C3903A)	Συμβατό	2	4
2	evountourakis	Σχολή ΜΠΔ	DSSL	11A Black Toner Cartridge (Q6511A)	Γνήσιο	1	5
3	pkontogiannis	ΔιεύθυνσηΤηλ/νιών, Δικτύων & Υπολ. Υποδομής		25X High Capacity Black Toner Cartridge (CF325X)	Συμβατό	6	3
4	sbouros	ΗΜΜΥ		304A Black Toner Cartridge (CC530A)	Γνήσιο	1	5
5	sbouros	ΗΜΜΥ		304A Cyan Toner Cartridge (CC531A)	Γνήσιο	1	4
6	sbouros	ΗΜΜΥ		304A Magenta Toner Cartridge (CC533A)	Γνήσιο	1	5
7	sbouros	ΗΜΜΥ		304A Yellow Toner Cartridge (CC532A)	Γνήσιο	1	4
8	espanaki	ΔιεύθυνσηΤηλ/νιών, Δικτύων & Υπολ. Υποδομής		305X High Capacity Black Toner Cartridge (CE410X)	Συμβατό	1	5
9	mlagoudakis	Σχολή ΜΠΔ		38A Black Toner Cartridge (Q1338A)	Γνήσιο	1	5

Figure 30 Viewing the record of the order as placed in the pilot testing

7.2 Results and Satisfaction Survey

After the end of the pilot test, it was of a major importance to obtain the users' feedback about their overall experience with the platform. For this purpose, questionnaires were sent in order to perform a satisfaction survey. The sample of this survey consists of 19 participants and in this section, the results will be discussed.

The first section of the questionnaire refers to demographics. The first question was about the sex of the participants. As shown in Figure 31, males represented 57.89% of the sample, while females 42.11%.

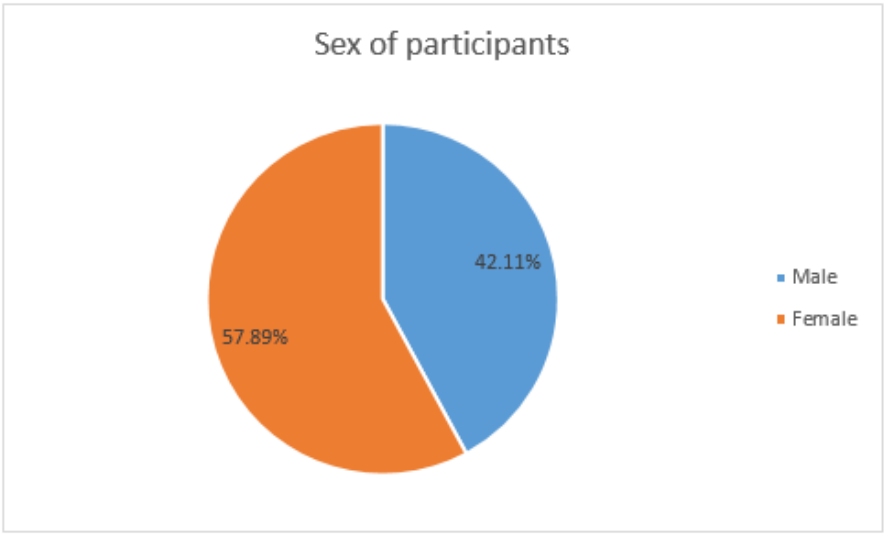


Figure 31 Sex of participants

The next question refers to the participants' age, where the majority of the participants were above 45 years old (42.11%), 31.58% were between 18-25 years old and 26.32% were between 36-45 years old (Figure 32).

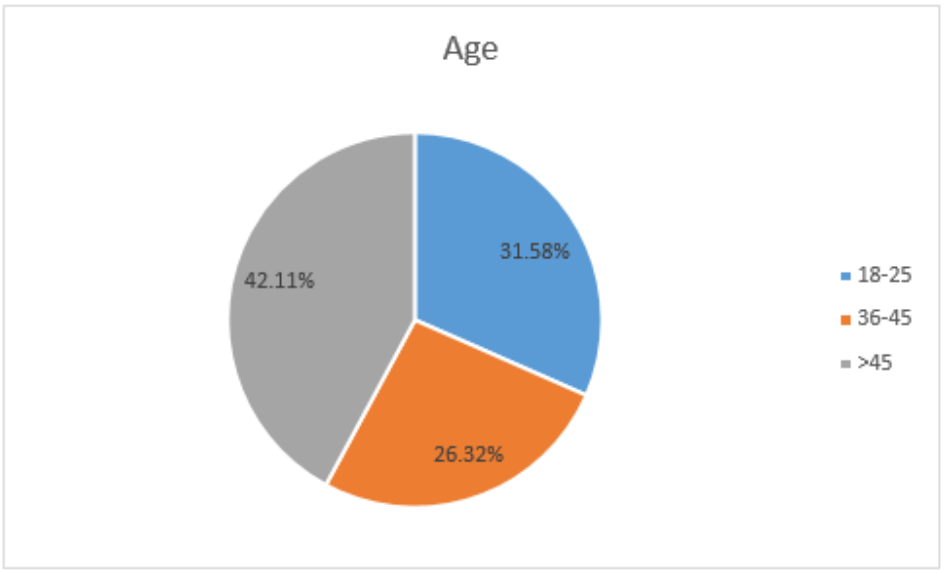


Figure 32 Age of participants

Referring to the participants' profession, the majority of the sample (36.84%) were Laboratory Teaching Staff Members (referred as EDIP), followed by the students with a percentage of 31.58%. Furthermore, 21.05% of the sample were administrative staff and only 10.53% were professors (Figure 33).

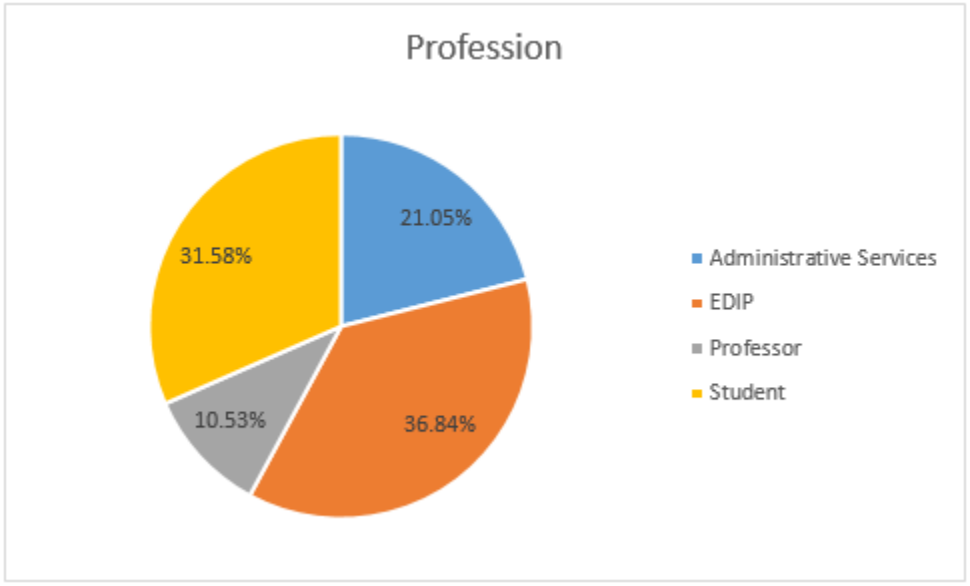


Figure 33 Profession of the participants

The following two questions refer to the time needed by the users to order ink cartridges with or without the Ink BManager. From Figure 34 one may see that most of the participants (57.89 %) have never ordered an ink cartridge before, so it is advisable not to compare the two alternatives (orders with or without Ink BManager) while taking into account the answers of the whole sample. For this reason in Figure 36, there a comparison between those two alternatives, where only the part of the sample that has placed an order without the platform is taken into consideration. More specifically, the mentioned part of the sample consisted of 10 participants, where nine of those stated that with the platform the time spent was less

than five minutes, while one person stated that needed more than fifteen minutes. At the same time, only 3 people needed less than 5 minutes for the completion of an order without Ink BManager. In this context, indeed Ink BManager limited the time spent by the users for an order.

In the question about the time needed to complete an order with Ink BManager, as shown in Figure 35, the majority of the sample (68.42%) answered that needed less than 5 minutes. Following, 26.32% needed between 6 and 10 minutes, whereas one user (5.26%) needed more than 15 minutes. Here it should be mentioned that no one needed between 11 and 15 minutes.

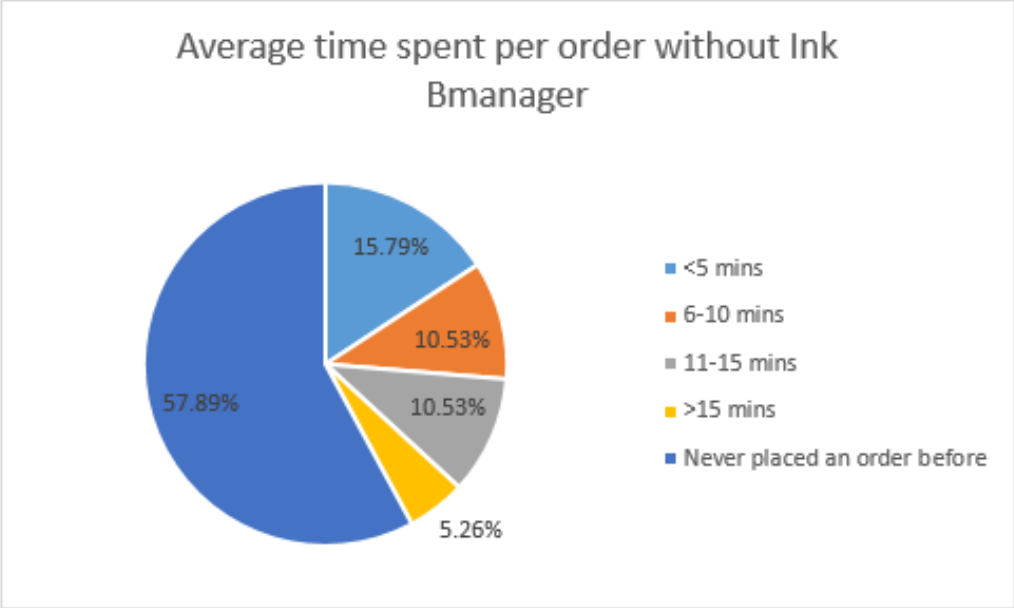


Figure 34 Average time needed to complete an order without Ink BManager

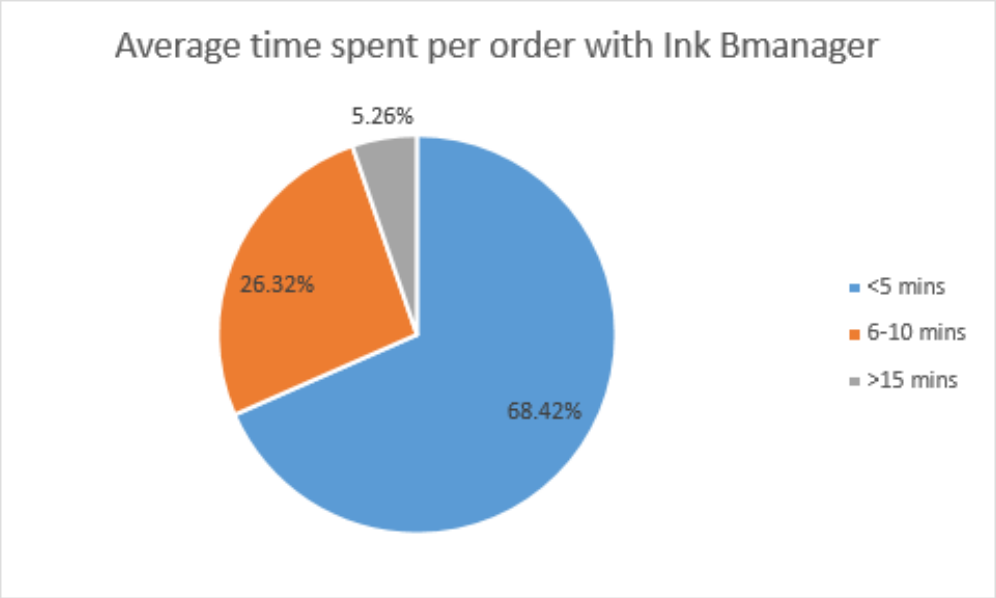


Figure 35 Average time needed to complete an order with Ink BManager

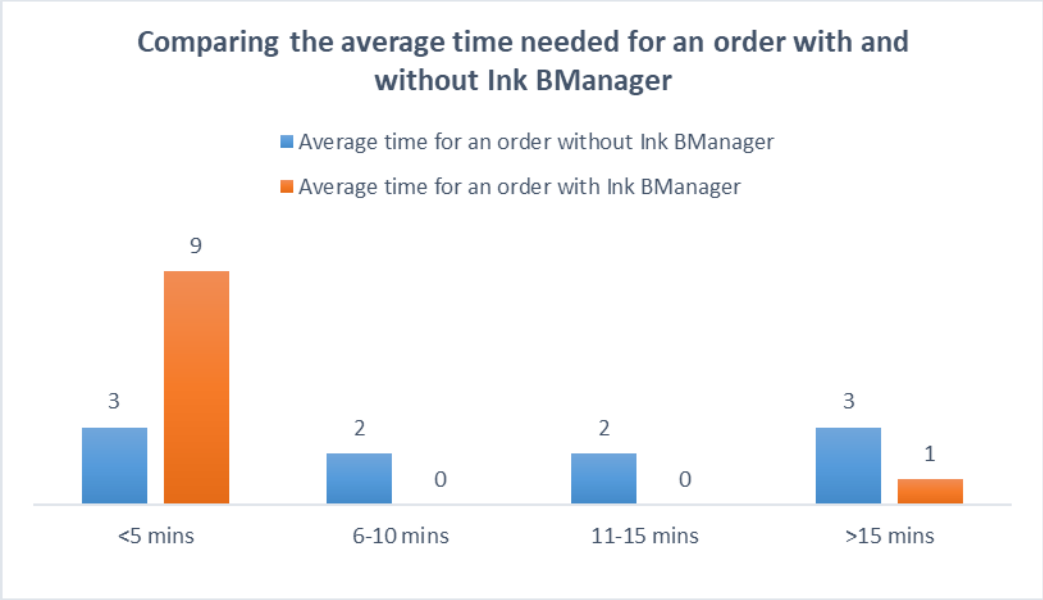


Figure 36 Comparing the average time needed for an order with and without Ink BManaer

The following question is a key one, since the users are questioned whether they believe that Ink BManager improved the procurement process or not, or if they think that the platform neither improved nor worsened the process. The vast majority with a percentage of 94.74% replied that the platform improved the process, while 5.26% (one user) stated that the platform neither improved nor worsened the whole process (Figure 37).

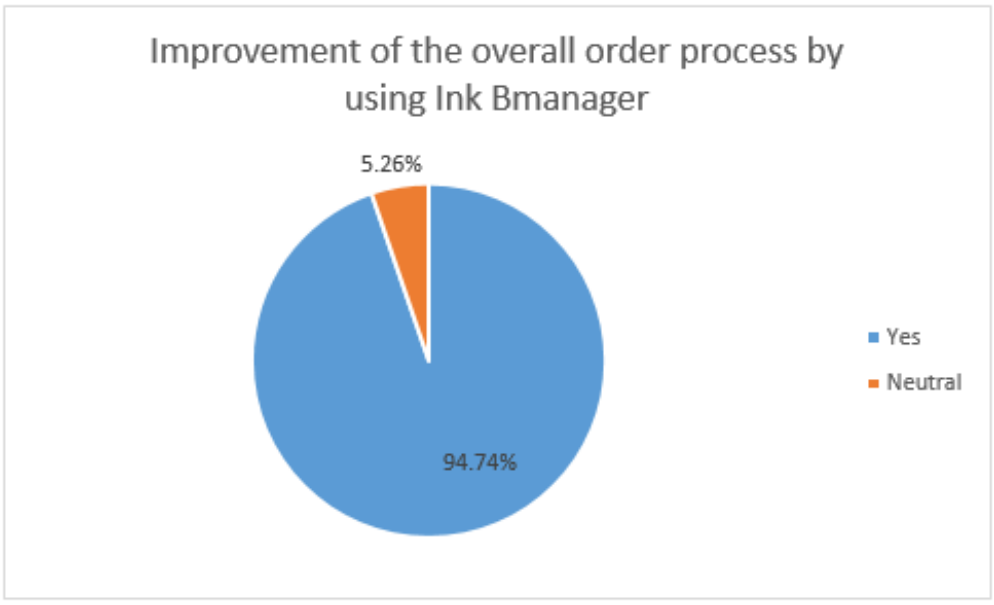


Figure 37 Improvement of the overall procurement process by using Ink BManager

The following two questions refer to the ordered products. More specifically, at first the users had to state if they placed an order for a personal printer (31.58%), a lab (31.58%), an Academic Department (5.26%), a School Faculty (10.53%) or an address (21.05%). Then, they had to state how many cartridges they ordered. As shown in Figure 39, the majority with 57.89% ordered less than 3 items, following, the users ordered 7-9 items (21.06%), while 10.53% ordered 5-6 items or more than 10 items.

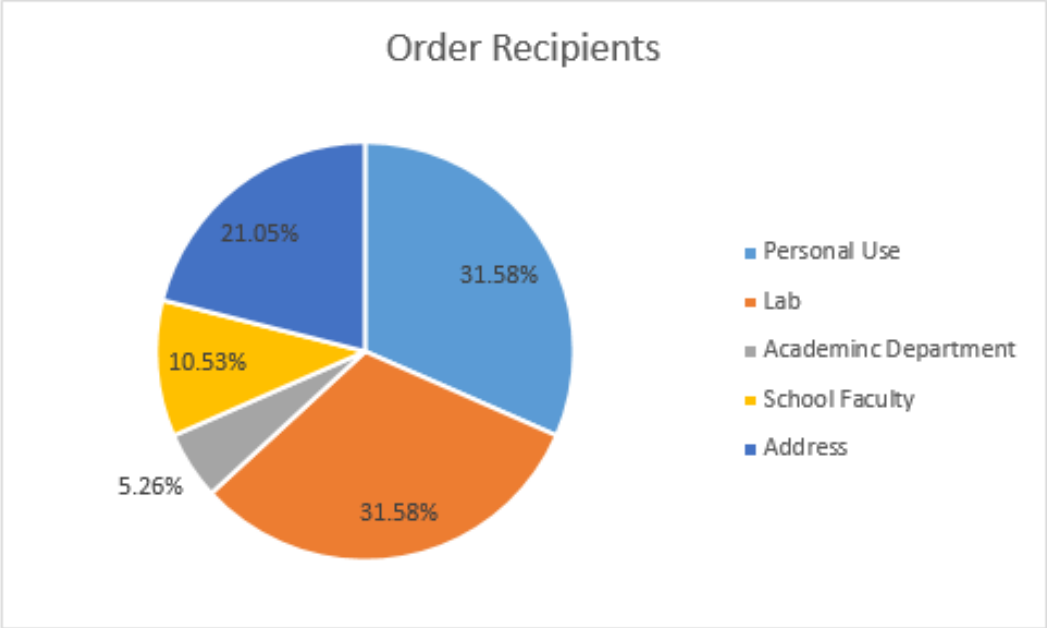


Figure 38 Improvement of the overall procurement process by using Ink BManager

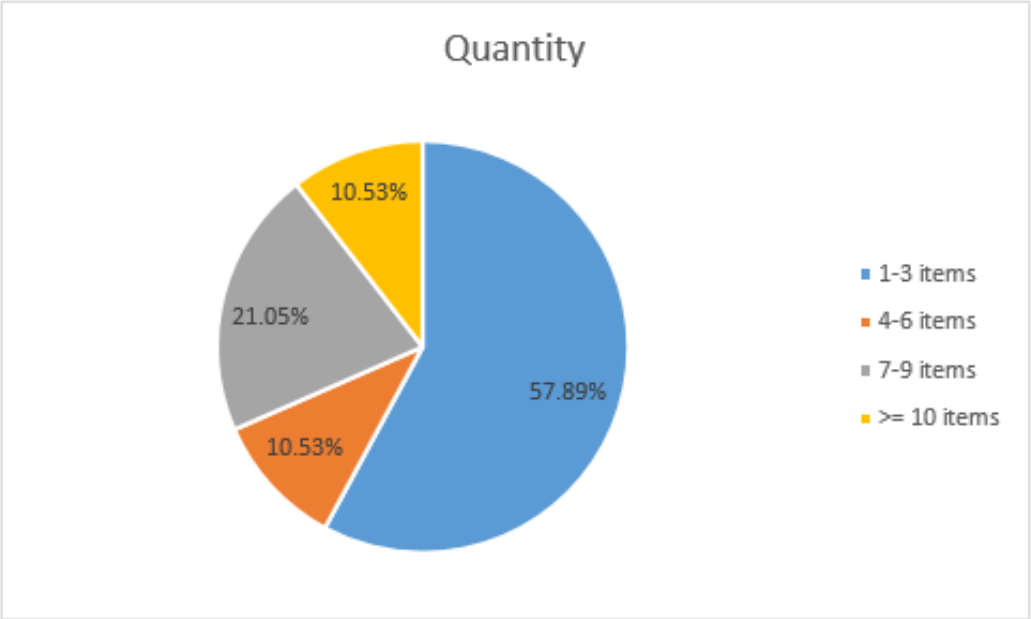


Figure 39 Quantity of ordered items

The second part of the questionnaire refers to the satisfaction of the users concerning their experience with Ink BManager. The users had to rate some features of the platform by stating if they were very satisfied, satisfied, neutral, dissatisfied or very dissatisfied. Bellow the answers of the users will be explained.

In the first question, the users had to rate how user-friendly they found the platform, taking into consideration the menu, the messages, the options etc. As shown in Figure 40 most of the users (47.37%) were very satisfied, while the satisfied users were 47.37% and the neutral ones were 15.76%.

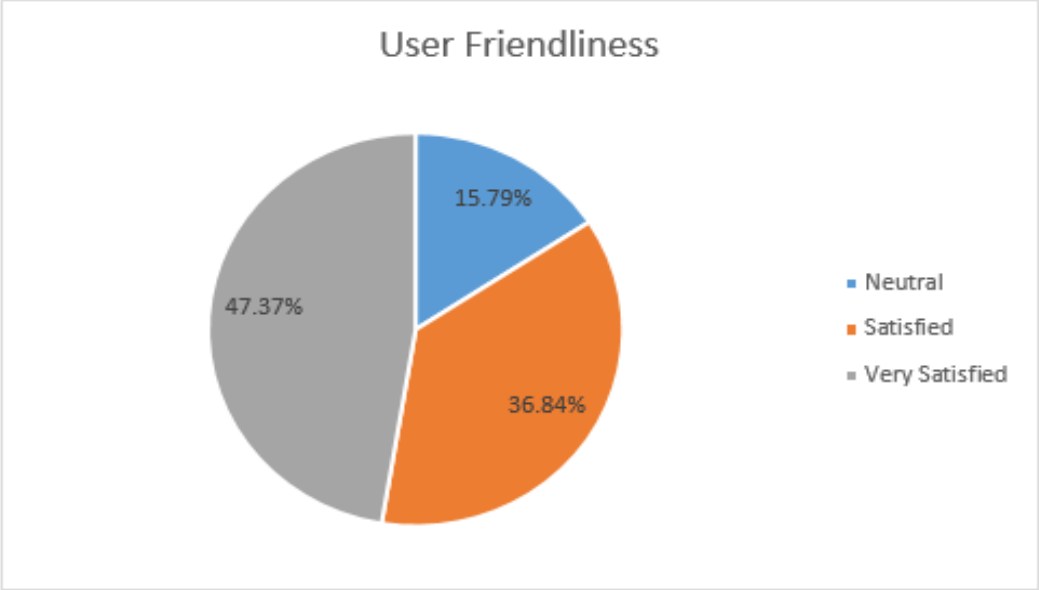


Figure 40 Satisfaction regarding user-friendliness of the platform

The next question refers to the ease of navigation within the platform (Figure 41). Even though it was pointed out from the observation and from the feedback obtained prior to this survey that many people had a slight difficulty to adapt to the new platform, the results of the survey prove that maybe some people indeed faced an initial difficulty to browse, but soon enough they found how to navigate within the platform. The results of this question show that the majority (57.89%) of the users were very satisfied, while 26.32% were satisfied and 15.79% were neutral.

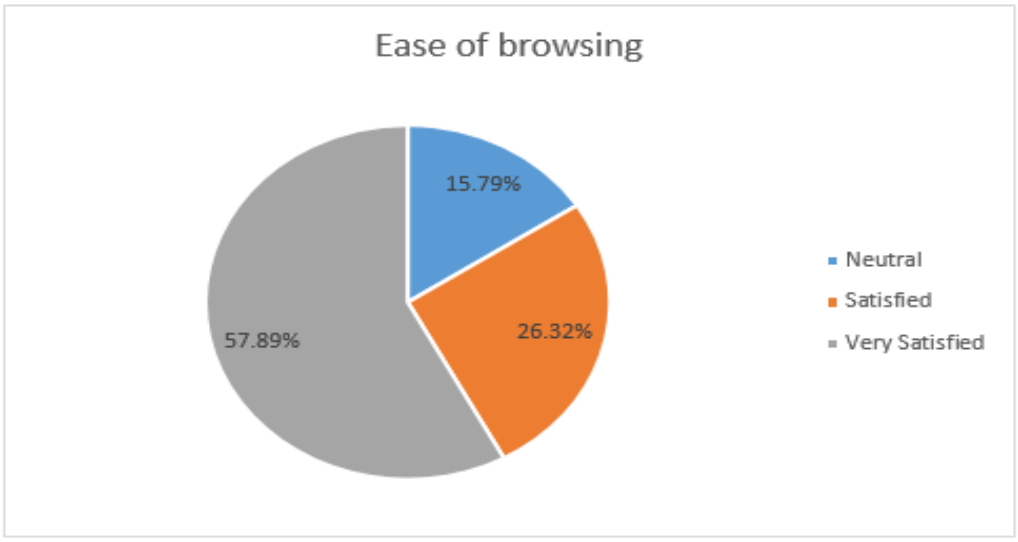


Figure 41 Satisfaction regarding the ease of navigation

The following question refers to the graphics of the platform. As shown in Figure 42, 42.11% stated that they were very satisfied, while the other 42.11% stated that they were satisfied. Also, 10.53% stated that they were neutral and one person (5.26%) stated that he/she was dissatisfied.

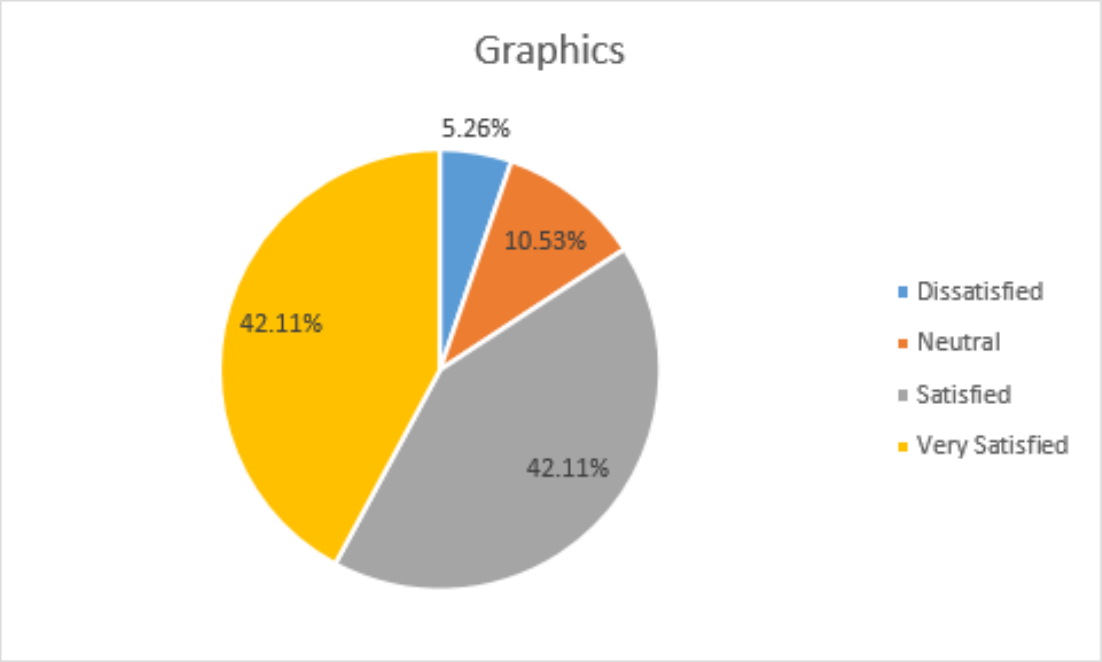


Figure 42 Satisfaction regarding the graphics of the platform

The satisfaction regarding the variety of the available printers and their corresponding products is depicted in the next question. The majority (89.47%) of the users stated that they were very satisfied with the variety, while 10.53% stated that they were satisfied (see Figure 43).

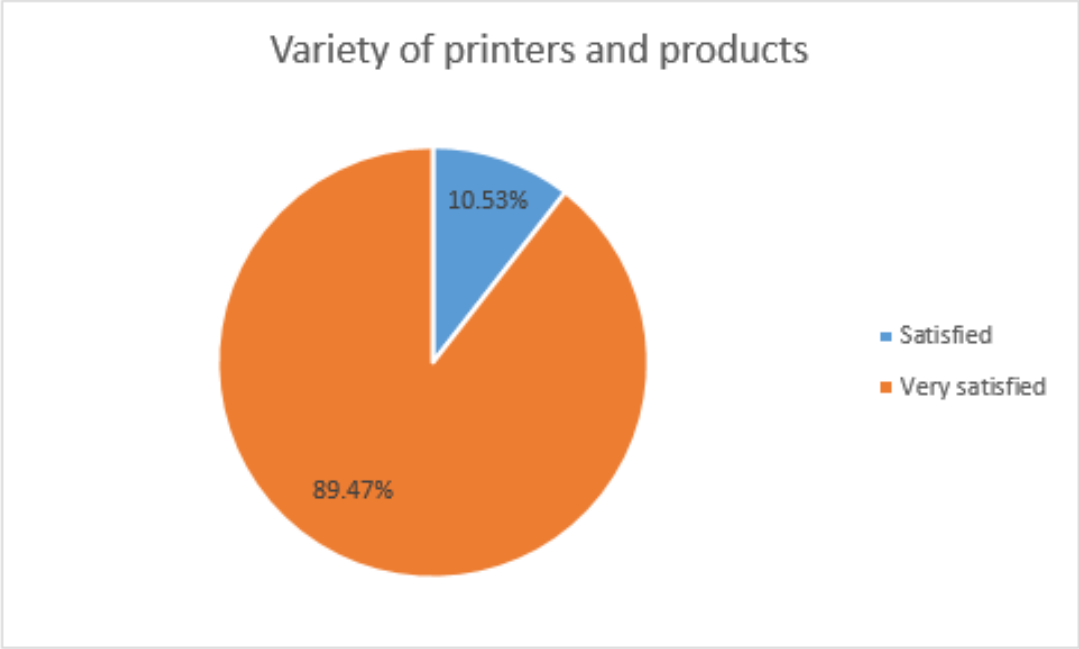


Figure 43 Satisfaction regarding the variety of the printers and their corresponding products

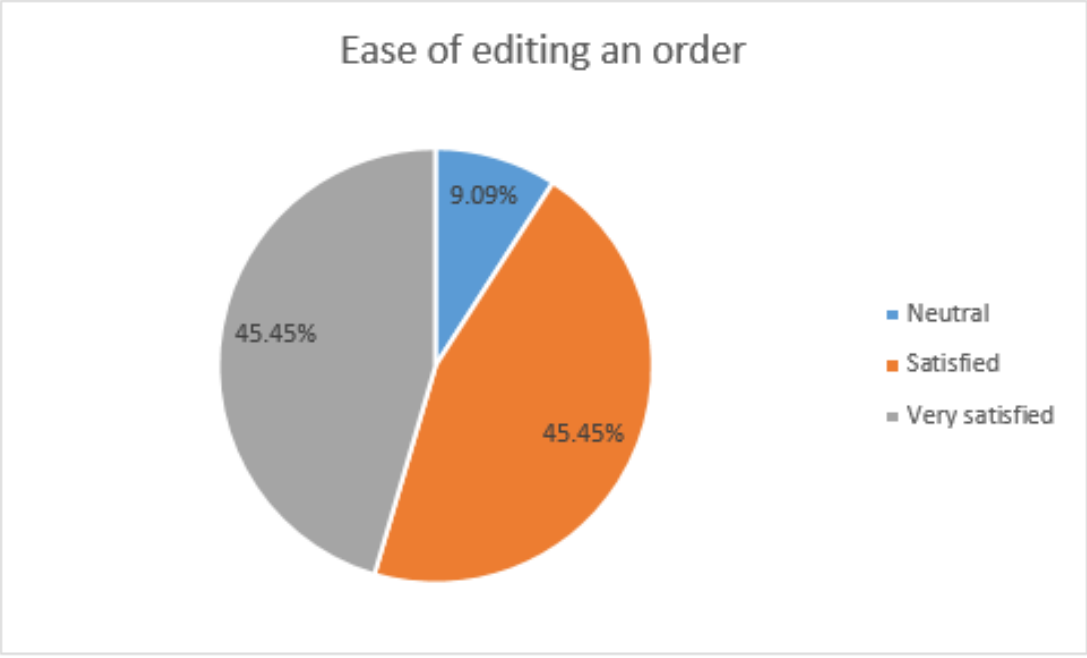


Figure 44 Satisfaction regarding the ease of editing a placed order

As shown in Figure 44, some users had to rate their experience regarding the editing of their order. The present question was crucial because in Ink BManager the way that someone can edit a placed order is not as easy as it is in an e-shop for example. More specifically, in order to edit one order, the user has to

send a message to the admin and the admin has to open the user's form. This form contains the ordered products and in order to make a change, the user has to delete the item and then order it again. Having in mind this process, the users stated that regarding the ease of editing one order, they were very satisfied in the percentage of 45.45%, satisfied in the percentage of 45.45% and neutral in the percentage of 9.09%. Here it should be noted that this question was not answered from every user, since not every user edited his order.

Continuing, the users had to rate the overall ease of filling one form. In this question, 78.95% stated that they were very satisfied, while 21.05% stated that they were just satisfied (Figure 45).

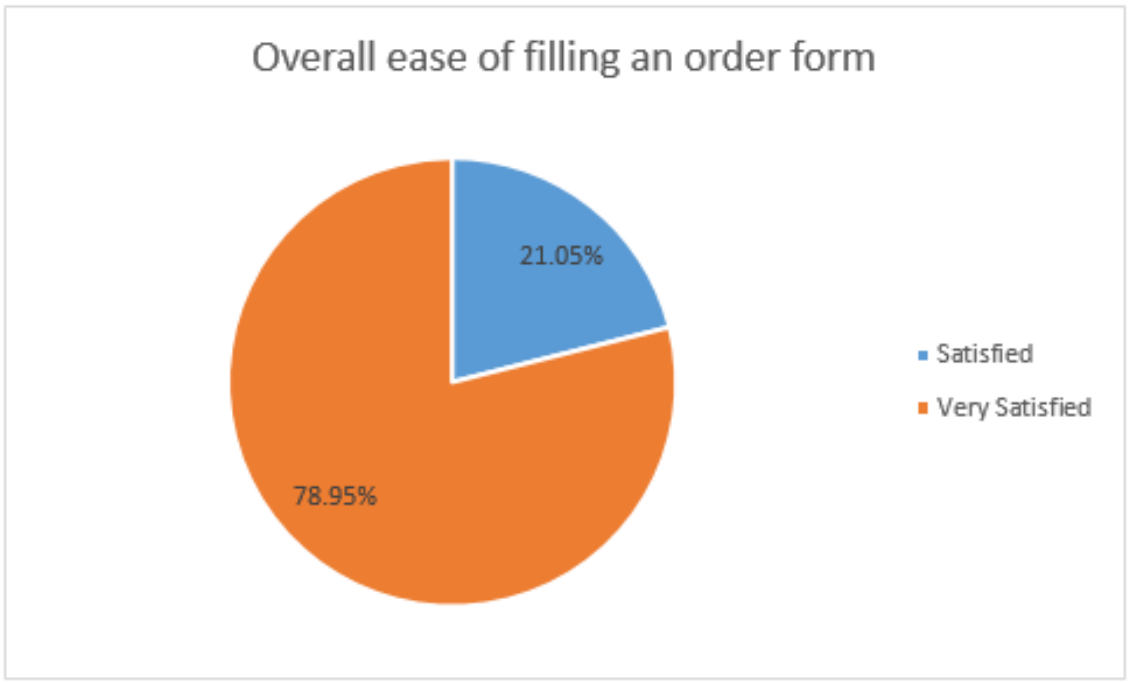


Figure 45 Satisfaction regarding the ease of filling the order form

The last question that users had to answer, refers to the overall satisfaction in using Ink BManager (Figure 46). This is another vital question, since it shows how people rate the platform in whole. More specifically, the majority with 57.89% stated that they were very satisfied, while 31.58% were satisfied and 10.53% were neutral. These results are quite encouraging, and at the same time they show that further improvement can be achieved.

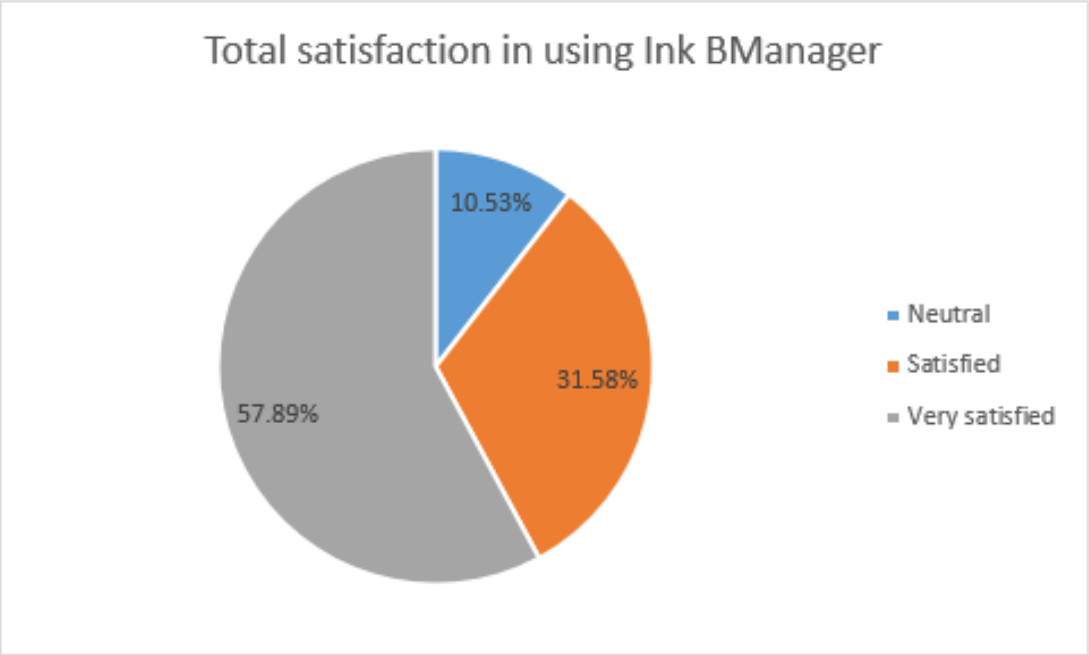


Figure 46 Overall satisfaction regarding the use of Ink BManager

After finding the frequencies that can be seen above, the correlations between the variables that affect the users' total satisfaction of Ink BManager should be also detected. For this reason, the statistical software SPSS was used in order to conduct some tests that will show which factors affect the users' total satisfaction. More specifically, the results of the Spearman correlation indicate that there is a strong positive association[3] between the total satisfaction regarding Ink BManager and user-friendliness, ($r_s[19] = .711$, $p = .001$). Also, there is a moderate positive association between the total satisfaction and the ease of browsing, ($r_s[19] = .679$, $p = .001$). Similarly, there is a moderate positive association between the total satisfaction and the graphics, ($r_s[19] = .665$, $p = .002$)[4]. The mentioned correlations should be taken into account for further improvement of the Ink BManger.

In Figure 42, the users' satisfaction regarding the graphics of the platform is presented. As someone may notice, graphics is the feature that is the most poorly rated by the users. For this reason, it is useful to understand how graphics affect other aspects of the platform. Someone may believe that graphics affect the user-friendliness, but from the conducted Spearman correlation it is shown that there is a weak positive association between the mentioned features, that is not statistically significant ($r_s[19] = .352$, $p = .137$), so a potential improvement to the graphics would not affect the user-friendliness. On the other hand, there is an expected relation between graphics and the ease of browsing, that is verified by the conducted Spearman correlation, which indicates a moderate positive correlation between the mentioned variables, that is statistically significant, ($r_s[19] = .583$, $p = .009$).

Following, while taking into consideration the present survey as well as the observation of users' behavior through the tests of the platform some recommendations will be made.

[3] Positive correlation means that when the value 1 increases the value 2 increases too.
[4] All the information extracted from SPSS can be found in the appendix.

7.3 RECOMMENDATIONS

The platform that was developed for the present thesis proved to be adequate for the automation of the procurement process of ink cartridges of the Technical University of Crete and can be used as it is in the real world. From the user's feedback, one can realize that platforms like the Ink BManager can be used to facilitate existing manual processes with public acceptance. But, as the finding from the survey suggest, complementary with the observations made during the development and testing, the present platform can be further improved.

During the tests of the platform it was obvious that some users found it hard to adapt, due to the layout that ProcessMaker offers and the fact that after the submission of their form users have to follow some more steps. For this reason, as future work it is recommended to find ways to offer a more user-friendly environment and limit the "next step" and "continue" buttons. A step further, with the right technical background the platform can be embedded in a more user-friendly web site designed for this purpose.

Furthermore, it is recommended to change the way that users edit their placed order. In this context, firstly, the platform should enable the users to edit their form without contacting the admin. Also, from the observation of the users' behaviour, some of them found it hard to edit their order. More specifically in the current version when a user states that he wants to edit his order, a form appears where the already placed order is listed. From this order, the user may select the item that needs editing and afterwards he should order this item again with the proper changes (with this way the system deletes the existing item and adds the new one). For this reason, it is recommended, to make the editing of an order similar to e-shops', meaning that since the list of the ordered items is presented, the changes may be done in that list without deleting and adding items.

Another recommendation is to use time events or catch events to link some processes, since there is already a logical sequence of how the processes are executed. For example, the admin firstly executes the process of "ordering ink cartridges", and when this process is ended and there is a list of all the products, the process of "budget creation" has to be started. When this process ends, the process of "updating the cost" should be executed too. Based on this, instead of manually starting the mentioned processes, events should be configured so that those are started automatically. Also, another thing that may be automated, is the way that the process of "adding a new printer/ product or a new department/ office" is executed interstitially in the process of "ordering ink cartridges". For future work, one may consider configuring the task to be automatically executed as long as there are new inks to be inserted to the database.

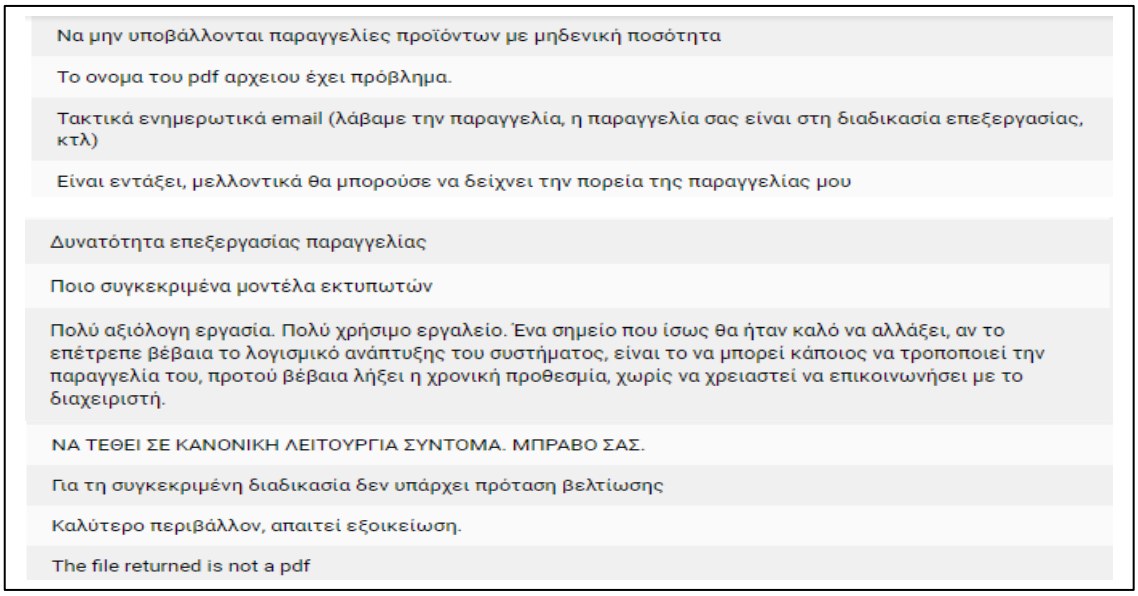
Also, as the people in charge for the procurements noted, there is a need for the users to be derived from the existing systems of the University, and the login should be done with a safe way by using the credentials that every user already has.

Lastly, the platform resides in a virtual machine hosted in a server provided by the University. During the installation, some security problems arose, so in order to avoid compromising the security of the server the platform is available only within the university's network. So in future work, there should be a focus in the security of the platform so that it can be available through the internet without any limitations

Following, the recommendations made by the users are listed.

- Ink BManager automatically sends an email to the user when an order is placed, including an attachment with a list of the ordered items. This attachment did not have the extension of .pdf so some users had to open the file by adding manually the extension. The mentioned problem was pointed out and it is resolved.

- Furthermore some people recommend to include an order's status update. If this would be implemented, it should be done while having in mind that many users do not want to receive regular emails.
- As mentioned above, users also point out the need for editing an order without contacting the admin. The reason why the present version of Ink BManager does not provide this capability, is because by giving freedom to the users one must foresee any "abnormal" behavior of the user.
- Also, one user pointed out that there is a need for a better environment. The ways to achieve that are mentioned above.
- Lastly, one user pointed that when there is zero quantity in an order this order should not be submitted. The particular problem is resolved.



8 CONCLUSION

It is known that every business seeks ways to increase its productivity in order to offer better services with less cost. Also, it is a fact that the Greek public sector is counterproductive due to the absence of defined business processes and the lack of process automation, leading to the provision of inadequate services. Business Process Management and Business Process Management Systems can be used to resolve this problem. In an attempt to find ways to increase the effectiveness of processes being carried by public institutions, like the Technical University of Crete, such solutions seem very attractive. Given that, the purpose of this thesis was to automate the process of ink cartridges' procurement by using an open source Business Process Management System and Business Process Management techniques. For this reason, the Bitnami's open source solution, ProcessMaker, was chosen among five other tested alternatives, and the web-based platform "Ink BManager" was developed in order to offer an interface where the users can place online their orders by filling a form, and where the admin can manage the whole process with the given tools specially designed to facilitate the procurement.

In this thesis, the theoretical background of Business Process Management was given, along with the explanation of every related notion. Furthermore, the way that Ink BManager was configured is described after outlining the technical background. Since Ink BManager aspires to work as a real life project, a pilot test was conducted and the users participated in a satisfaction survey, the results of which are presented in this paper. The developed platform was tested within the community of the Technical University of Crete and earned a positive feedback from the users and the people in charge of the procurements. Two main features that were appreciated by the participants were that the order form gives to a user the chance to quickly order ink cartridges by offering an extended list of the available printers and their corresponding ink cartridges, while allowing him/her to easily order a product without knowing its exact model, whilst reducing the likelihood of mistakes in the orders.

Generally, the necessity to work with greater agility within organizations, lead to the development of Ink BManager, which strives to improve the procurement process. With the use of the platform the admin can spend less time on the procurements, considering that he does not engage much with the users, since the platform automatically sends informative emails (that include directions of how to place an order) at the beginning of each period and users need to contact the admin only in special cases. Also, the system automatically issues an aggregate list of the ordered products, leading to the elimination of the time spent in gathering the orders. In regard to the financial management, Ink BManager enables the admin to view past procurements in an effort to keep archive and track the expenses.

Overall, if something has to be extracted from this thesis, is that Business Process Management is a "theory in practice", used to design business processes that will be enacted by Business Process Management Systems. With BPM and BPMSs, businesses -or public institutions- have the ability to improve their processes, affecting the whole performance of the organization. This can be achieved with the right definition of the processes, e.g. with the right division of labour, the specification of the needed resources and the definition of time limitations. Business Process Management offers a great insight into the business processes and any deficiency can easily be located, offering the opportunity to take measures for further improvement. In this thesis, there was indeed an improvement, achieved by using the technology and the BPMSs, in order to provide better services to the community of the University, while reducing the time spent on the procurement of ink cartridges from the people in charge. So the main goal of BPM was met, and one can say that by properly defining how the work should be done and with the proper use of technology through BPMSs, businesses can work with less cost, in less time, while offering better services.

8.1 LESSONS LEARNED

While developing Ink BManager three valuable lessons were learned

1. You can be hacked more easily than you think.
2. Users will act very unpredictably.
3. When developing an app intended for public use, the user interface should be as simple as possible.
4. There are ways to make a business or a public institution more efficient and on of them is BPM and BPMSs.

Starting with the first lesson learned, while hosting the platform in the virtual machine in order to make it public, the virtual machine was hacked two times. The way that it was hacked did not show an intension from the hackers' side to steal any valuable data, rather just to annoy us with their presence. More specifically, a bot was logging into the virtual machine with a great frequency, leading to the creation of huge system logs that were as big as the whole available space, meaning that the platform could not find the proper resources in order to work stably. So, for future work it is advised to give the required attention to the security when going online, because hackers will find the way to attack even applications that do not have to offer something "valuable".

Another valuable lesson, that was pointed out from the beginning and was verified with the conducted tests, it is that users will act unpredictably. So, it is the job of a developer, to predict every possible scenario in order to prevent or fix one user's "abnormal" action.

Next lesson to be learned, is that the saying "A user interface is like a joke. If you have to explain it, it's not that good" is very wise. On the one hand, some people are natural in navigating into web applications without needing any kind of directions, on the other hand there are people that find it difficult to adapt to new interfaces, even if those are slightly different from the ones that they are used to. So if I would change something in this project and in every other future project, it would be to develop an interface that would be extremely easy even for those that are not so natural with computers.

Probably the most important lesson learned from this thesis, is that there are many ways to improve the processes of a business, and from my experience I can declare that BPM together with a BPMS is one of these ways. Public institutions could surely become more efficient with tools like ProcessMaker with no cost. The only limitation of such practices, is the culture of the business/institution, i.e. whether one potential change will be welcomed by the users or the difficulty in adapting will lead to the continuing of the prior less productive processes.

9 REFERENCES

[1] Burattin, A. (2015). Introduction to Business Processes, BPM, and BPM Systems. Lecture Notes in Business Information Processing, 11–21.

[2] Møller, C., Maack, C. J., & Tan, R. D. (n.d.). What is Business Process Management: A Two Stage Literature Review of an Emerging Field.

[3] Ko, R. K., Lee, S. S., & Lee, E. W. (2009). Business process management (BPM) standards: A sur-vey. Business Process Management Journal, 15(5), 744-791.

[4] M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley & Sons, 2005.

[5] Wil M. P. van der Aalst, Hofstede, A. H., & Weske, M. (2003, June 26). Business Process Management: A Survey.

[6] "ABPMP Standards for Business Process Management (BPM)." The BPM Profession - ABPMP International, www.abpmp.org/page/BPM_Profession.

[7] Wil M. P. Van Der Aalst. (2013). Business Process Management: A Comprehensive Survey. ISRN Software Engineering, 2013, 1-37

[8] Hammer, M., & Champy, J. (n.d.). Reengineering the corporation: A manifesto for business revolution. New York, NY

[9] Ko, R. K. (2009). A computer scientist's introductory guide to business process management (BPM). Crossroads, 15(4), 11-18.

[10] Hollingsworth, D. (1995). The workflow reference model. Winchester: Workflow Management Coalition.

[11] Findings: Confusion Remains Regarding BPM Terminology. (2008, March 10). Retrieved from https://www.gartner.com/doc/619126/findings-confusion-remains-regarding-bpm

[12] Georgakopoulos, D., Hornick, M., & Sheth, A. (1995). An overview of workflow management: From process modeling to workflow automation infrastructure. Distributed and Parallel Databases, 3(2), 119-153

[13] Dongsong Zhang (2005) Web Services Composition for Process Management in E-Business, Journal of Computer Information Systems

[14] Müller, Richard & Solti, Andreas. (2011). BPMN for healthcare processes. Proceedings of the 3rd Central-European Workshop on Services and their Composition. 705. 65-72.

[15] Wikipedia contributors. (2019, March 22). Camunda. In Wikipedia, The Free Encyclopedia. Retrieved 11:53, April 3, 2019, from https://en.wikipedia.org/w/index.php?title=C munda&oldid=888966760

[16] Wikipedia contributors. (2019, February 19). Bonita BPM. In Wikipedia, The Free Encyclopedia. Retrieved 11:51, April 3, 2019, from https://en.wikipedia.org/w/index.php?title=Bonita_BPM&oldid=884066221

[17] What is RESTful API? - Definition from WhatIs.com. (n.d.). Retrieved from https://searchmicroservices.techtarget.com/definition/RESTful-API

[18] What are the differences between BPM and SOA? (2017, May 18). Retrieved from https://www.integrella.com/2015/04/21/differences-between-bpm-and-soa/

[19] Aalst, Wil M. P..(2012). Aalst, W.M.P.: Business process management: a comprehensive survey.

[20] Leymann, F., Roller, D., & Schmidt, M. (2002). Web services and business process management. IBM Systems

10 APPENDIX

10.1 ADMINS GUIDE FOR INK BMANAGER

INK BMANAGER

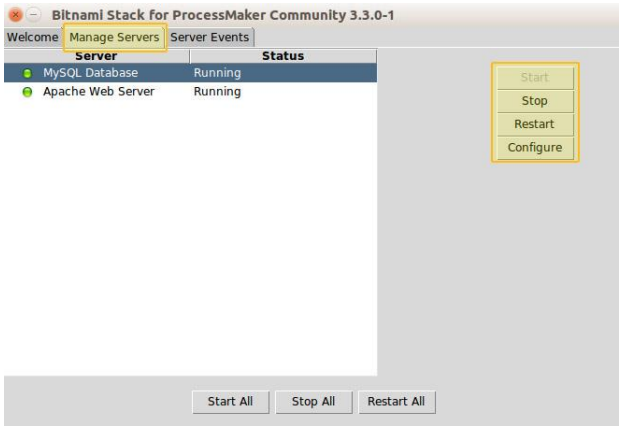
ΟΔΗΓΙΕΣ ΓΙΑ ΤΟΝ ΔΙΑΧΕΙΡΙΣΤΗ

10.1.1 Διαχείριση Server

10.1.1.1 Πρόσβαση στους Servers

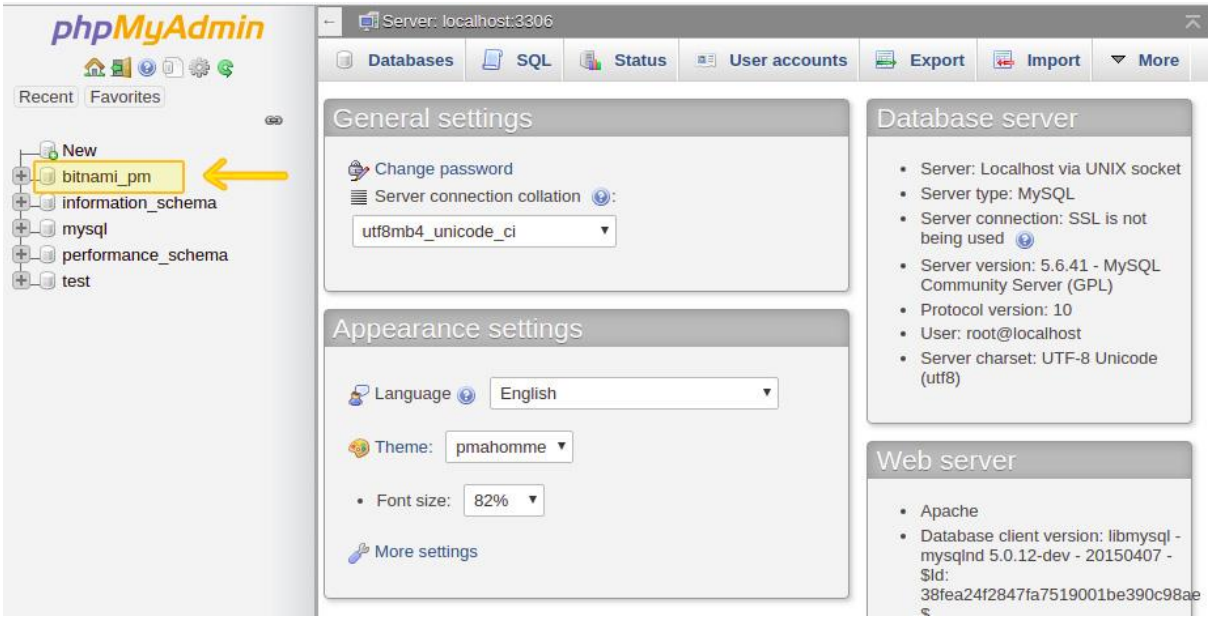
Το Processmaker λειτουργεί έχοντας δύο servers. Το mySQL Database και τον Apache Web Server. Είναι σημαντικό να μπορούμε να έχουμε πρόσβαση στους servers καθώς σε περίπτωση που το vm (virtual machinge) είτε κάνει εκκίνηση, είτε κλείσει, αυτομάτως και οι Server θα κλείσουν. Οι servers είναι προγραμματισμένοι να ανοίγουν αυτόματα με την εκκίνηση του vm, παρόλα αυτά θα πρέπει να ξέρουμε να τους διαχειριζόμαστε.

Για να διαχειριστούμε τους Server θα πρέπει:
1. Να αποκτήσουμε ακολουθήσουμε το directory του vm other locations/computer/opt/processmaker-3.3.0-1
2. Στο terminal να εκτελέσουμε την εντολή sudo ./manager-linux-x64.run
3. Από το παράθυρο Manage Server επιλέξουμε μία ενέργεια (start, restart, stop)



10.1.1.2 MySQL Database

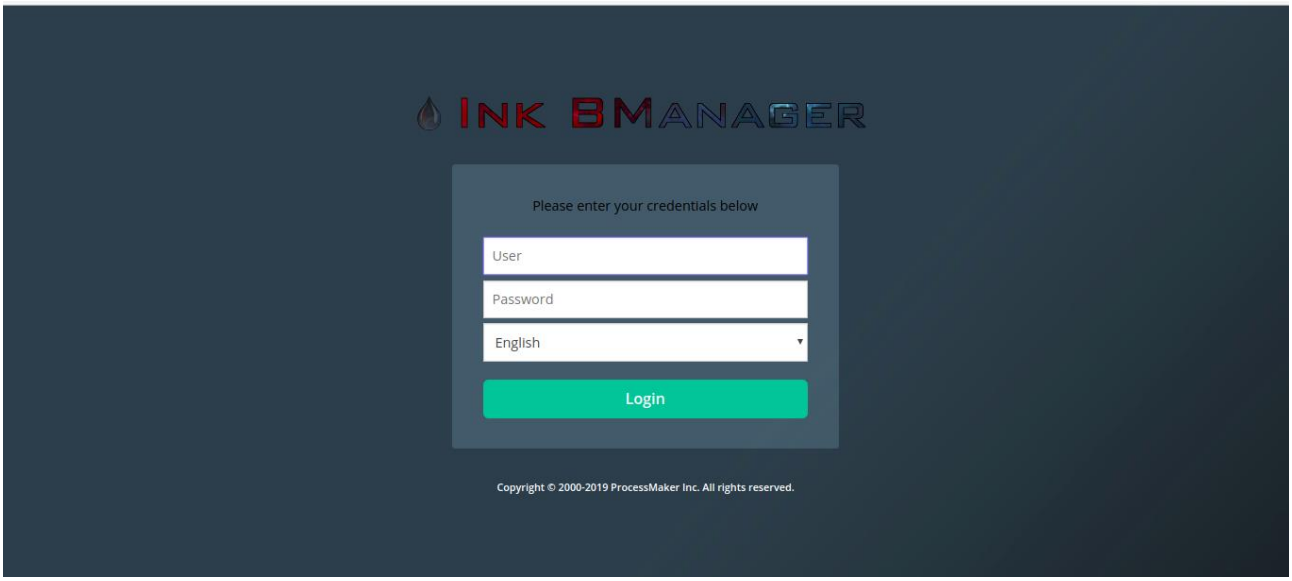
Μέσω του mySQL Database έχουμε πρόσβαση σε όλους του πίνακες που χρησιμοποιεί το ProcessMaker. Για να μπορέσουμε να επεξεργαστούμε έναν πίνακα θα πρέπει να συνδεθούμε στο vm. Εφόσον ο Server είναι ανοιχτός, ακολουθώντας τον σύνδεσμο https://147.27.70.93/phpmyadmin/ συνδεόμαστε στο phpMyAdmin. Στα αριστερά βλέπουμε τις διαθέσιμες βάσεις δεδομένων. Αυτή που μας ενδιαφέρει είναι η bitnami_pm.



10.1.2 Log in στο ProcessMaker

1. Ακολουθούμε τον σύνδεσμο https://147.27.70.93/

2. Εισάγουμε όνομα χρήστη και τον κωδικό.

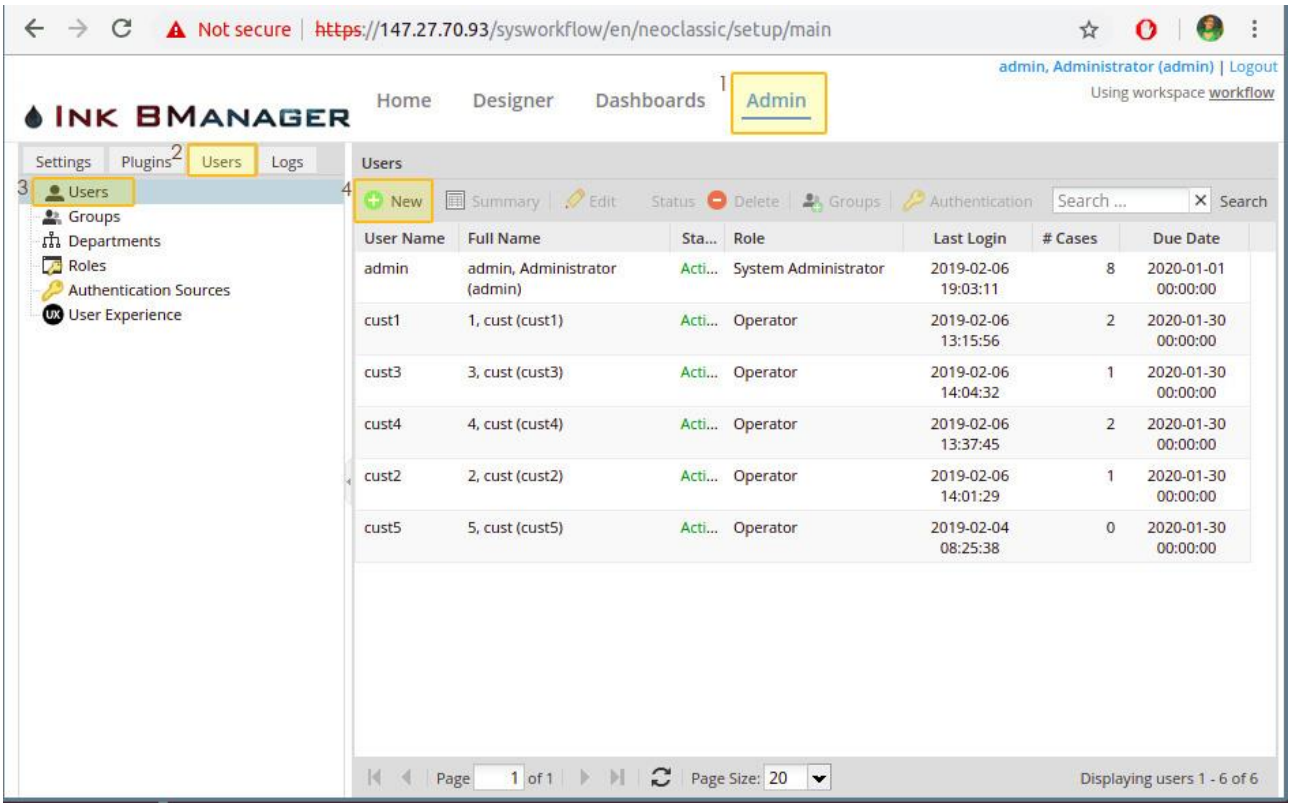


10.1.3 Δημιουργία χρηστών

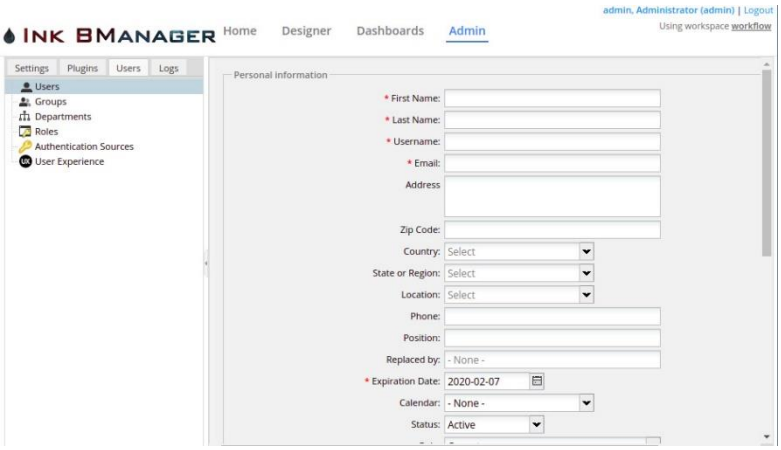
Ένα από τα δικαιώματα που έχει μόνο ο διαχειριστής είναι να μπορεί να δημιουργεί και να επεξεργάζεται νέους χρήστες.

10.1.3.1 Δημιουργία νέου χρήστη

1. Στην καρτέλα Admin στην αριστερή πλευρά επιλέγουμε users



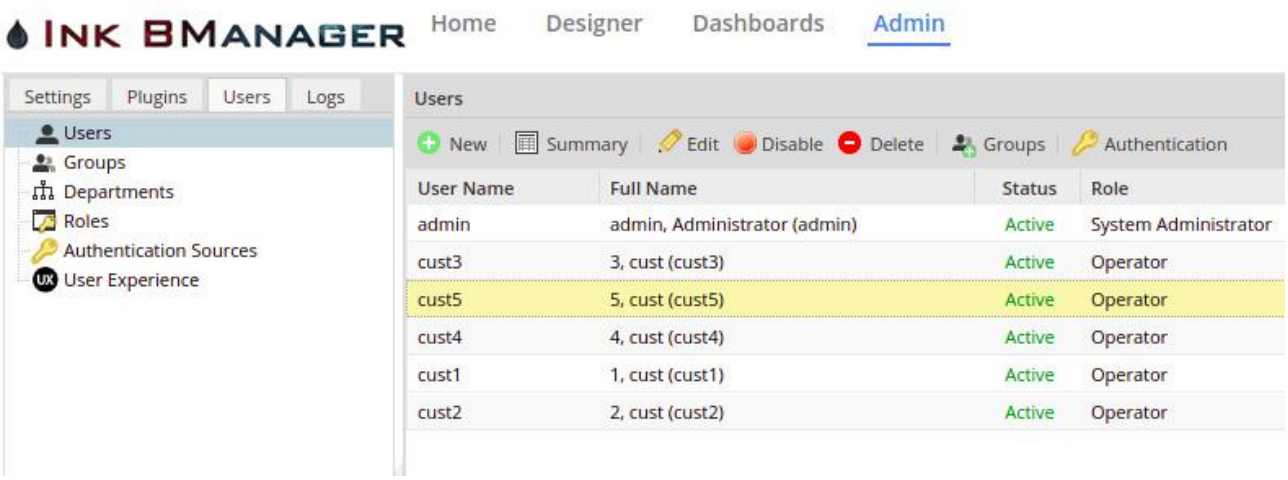
2. Επιλέγουμε new για την δημιουργία νέου χρήστη, και στην συνέχεια εισάγουμε τα στοιχεία του.



10.1.3.2 Επεξεργασία στοιχείων υπάρχοντος χρήστη

1. Από την λίστα χρηστών κάνουμε κλικ πάνω στον χρήστη που θέλουμε να επεξεργαστούμε τα στοιχεία του.

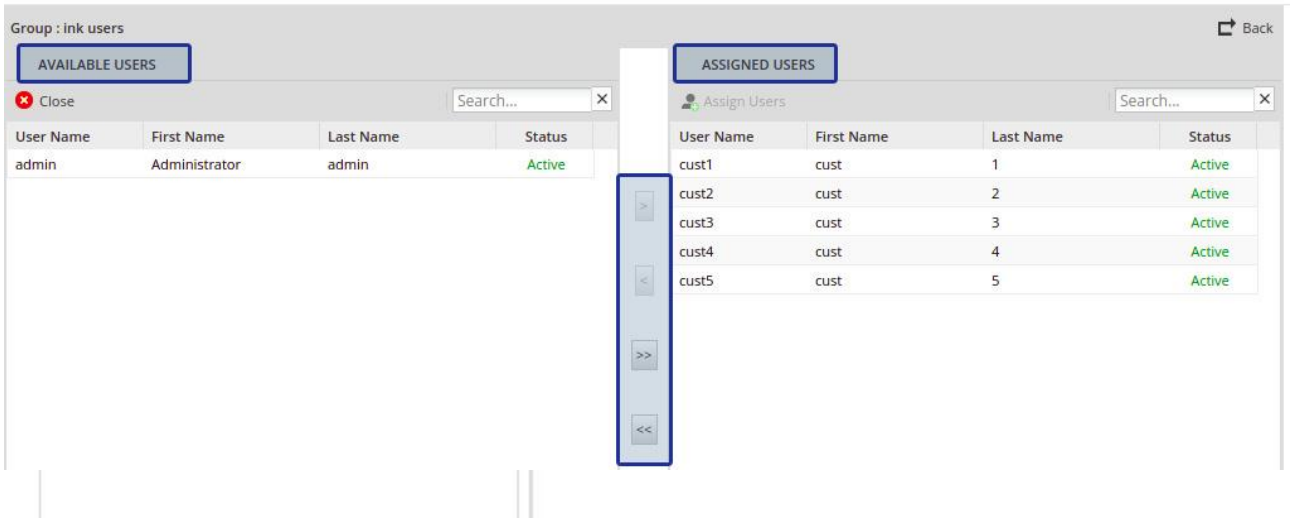
2. Πατάμε edit.



10.1.3.3 Δημιουργία Group

Υπάρχει ένα group που είναι πολύ σημαντικό να μπορούμε να διαχειριζόμαστε. Αυτό είναι το ink users. Στο group αυτό πρέπει να υπάρχουν όλοι οι χρήστες του συστήματος, εκτός από τον admin. Στην ουσία υπάρχει για να μπορεί να δίνει πρόσβαση στους χρήστες στην διαδικασία "επικοινωνία με τον admin". Εάν λείπει ένας χρήστης δεν θα μπορεί να στείλει mail στον admin μέσω του Processmaker.

1. Από την καρτέλα admin επιλέγουμε στα αριστερά την καρτέλα users
2. Από εκεί επιλέγουμε groups
3. Διαλέγουμε το ήδη υπάρχον group με όνομα ink users

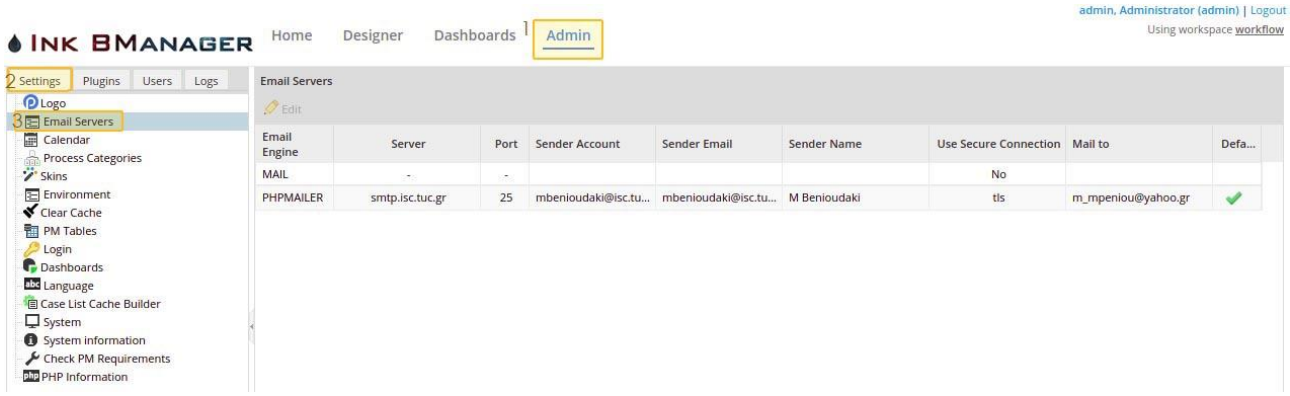


4. Εισάγουμε χρήστες στο group μέσω των βελών που βρίσκονται στο κέντρο της οθόνης. Πρέπει στο "assigned users" να υπάρχουν όλοι οι χρήστες εκτός από τον admin.

10.1.4 Email Server

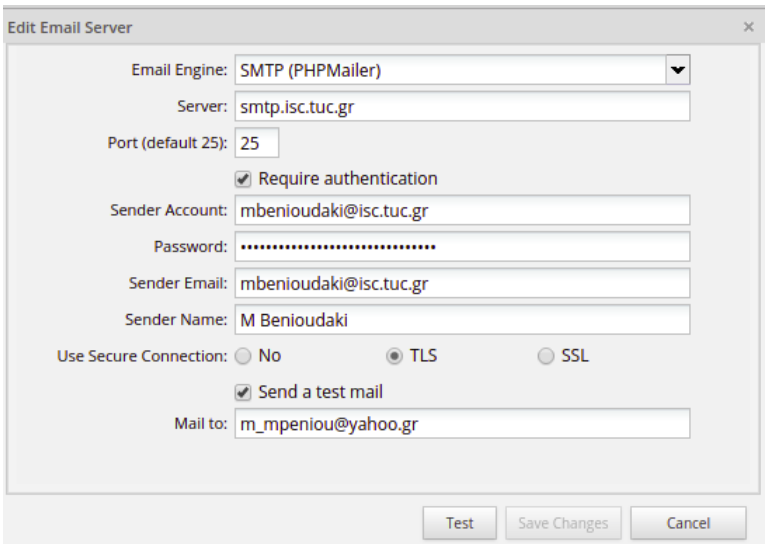
Ο email server πρέπει να ρυθμίζεται ανάλογα με το email του admin. Εάν ο admin δηλώσει mail με @gmail.com θα πρέπει να ρυθμιστεί ο server για το gmail. Εάν δηλώσει το @isc.tuc.gr θα πρέπει ο server να ρυθμιστεί για το ιδρυματικό mail. Ακολουθούν οδηγίες για την ρύθμιση του email server.

1. Στην καρτέλα admin στα αριστερά επιλέγουμε την καρτέλα settings και έπειτα email servers.



2. Επιλέγουμε το PHPMAILER και πατάμε edit.

3. Συμπληρώνουμε τα στοιχεία για πολυτεχνικό mail ή για gmail.

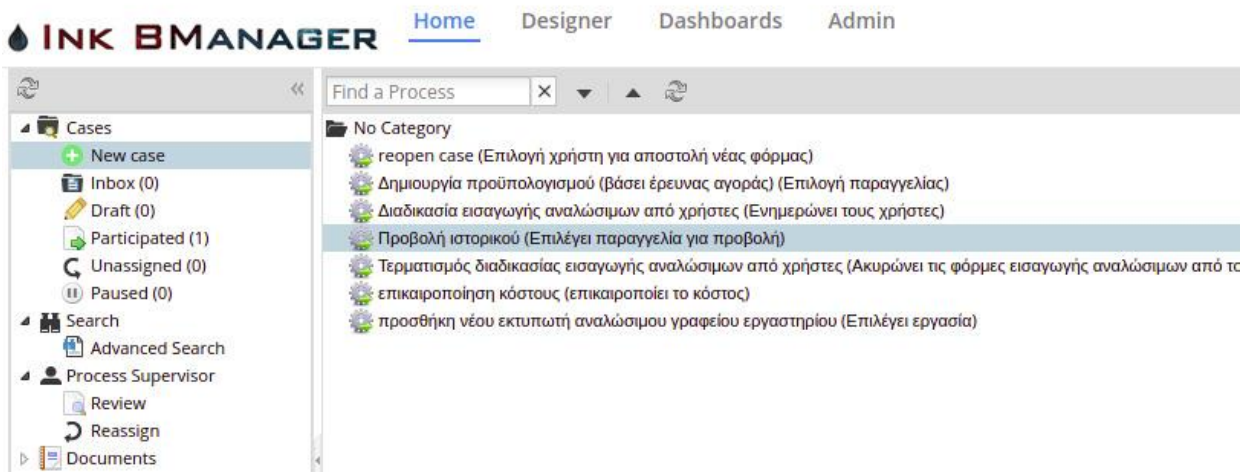


4. Πατάμε test και save changes.

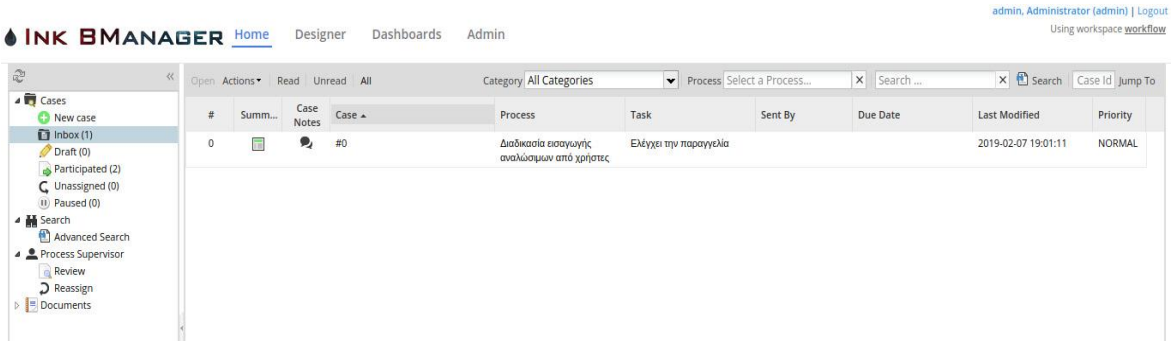
Υπάρχει περίπτωση μετά από ένα reboot του vm να μην μπορεί να συνδεθεί ο email server. Για να διορθωθεί το πρόβλημα, στο αρχείο resolv. conf που βρίσκεται στον φάκελο etc του vm αλλάζουμε το nameserver σε 147.27.18.1 .

10.1.5 Γενικές οδηγίες για την εκτέλεση των διαδικασιών

- Ξεκινάμε μια διαδικασία από την καρτέλα home. Στα αριστερά επιλέγουμε new case. Με διπλό κλικ επιλέγουμε την διαδικασία που θέλουμε να ξεκινήσουμε



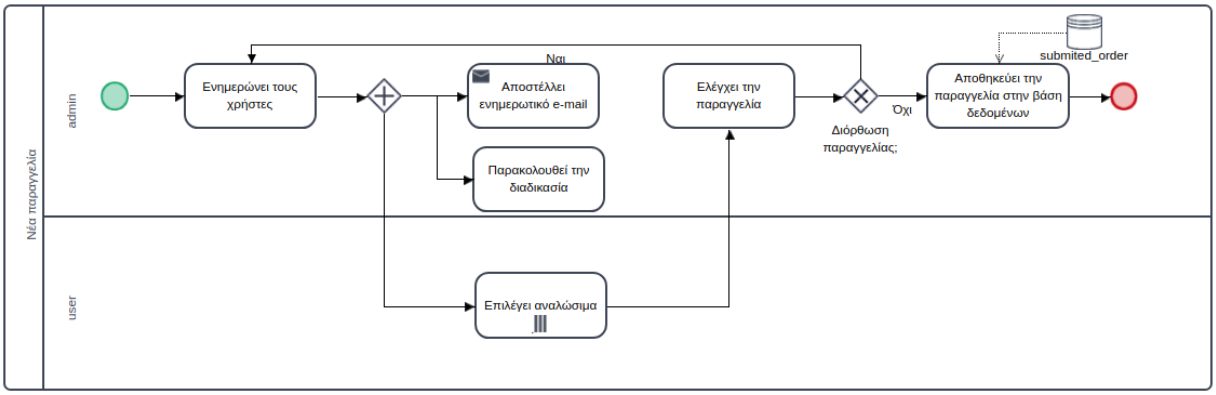
- Μία νέα εργασία που πρέπει να εκτελεστεί, εμφανίζεται στο κέντρο της αρχικής σελίδας στην καρτέλα home και στο inbox υπάρχει κάτι διαφορετικό από το (0). Για να την εκτελέσουμε κάνουμε **διπλό κλικ**.



- Φροντίζουμε σε κάθε εκτέλεση μία εργασίας να πατάμε **next step, continue ή finish** μέχρι το σύστημα να μας κατευθύνει στην αρχική οθόνη του home.
- Σε πολλές εργασίες υπάρχουν output documents. Τα αρχεία αυτά είναι .doc ή pdf και μπορούμε να τα κατεβάζουμε πατώντας στο open.
  Προσοχή πρέπει να πατάμε next step και continue ή finish μέχρι να τελειώσει η διαδικασία.

10.1.6 Διαδικασίες για την προμήθεια μελανοδοχείων

10.1.6.1 Διαδικασία εισαγωγής αναλώσιμων από χρήστες



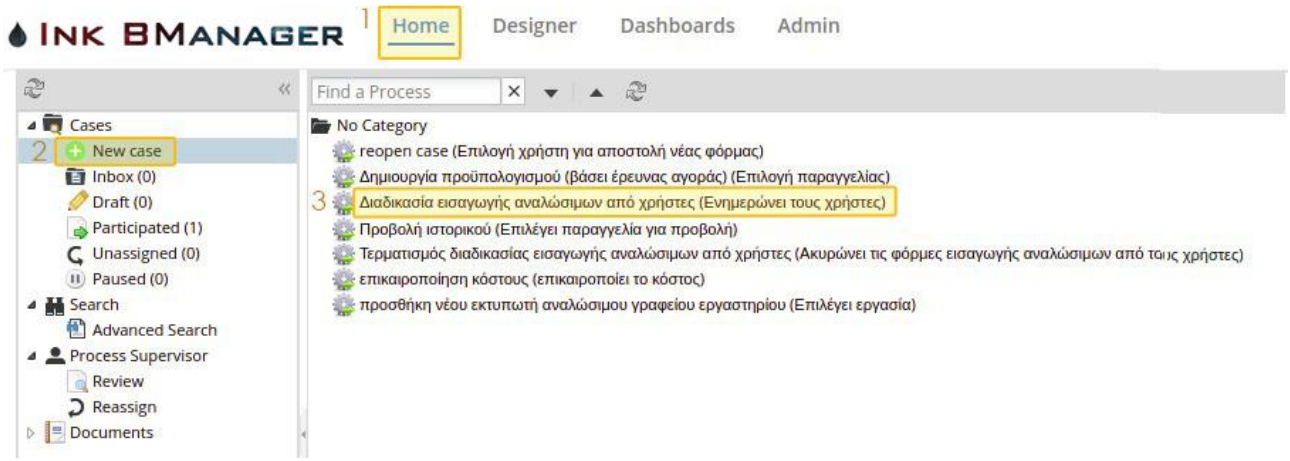
Αρχικά ο διαχειριστής διαλέγει τους χρήστες ο οποίοι θα συμπληρώσουν την παραγγελία τους. Αυτομάτως το σύστημα τους αποστέλλει ένα ενημερωτικό email. Στην συνέχεια οι χρήστες αποστέλλουν την παραγγελία τους. Σε αυτό το στάδιο, σε περίπτωση που κάποιος χρήστης χρειαστεί αλλαγή στην παραγγελία του, ο διαχειριστής θα πρέπει να ανοίξει την φόρμα του κάνοντας reopen. Εφόσον οι χρήστες τελειώσουν με την εισαγωγή των προϊόντων στον διαχειριστή θα εμφανιστεί μια λίστα με τις επιλογές των χρηστών τις οποίες είτε θα αποδεχτεί είτε θα απορρίψει. Εάν υπάρχουν προϊόντα που έχουν απορριφθεί, ο διαχειριστής θα μπορεί να καλέσει τους χρήστες να διορθώσουν την παραγγελία τους. Όταν ολοκληρωθεί η διαδικασία παραγγελίας από όλους τους χρήστες το σύστημα αυτόματα συγκεντρώνει όλα τα προϊόντα χωρίζοντας τα σε συμβατά και γνήσια.

10.1.6.1.1 Εκτέλεση της διαδικασίας

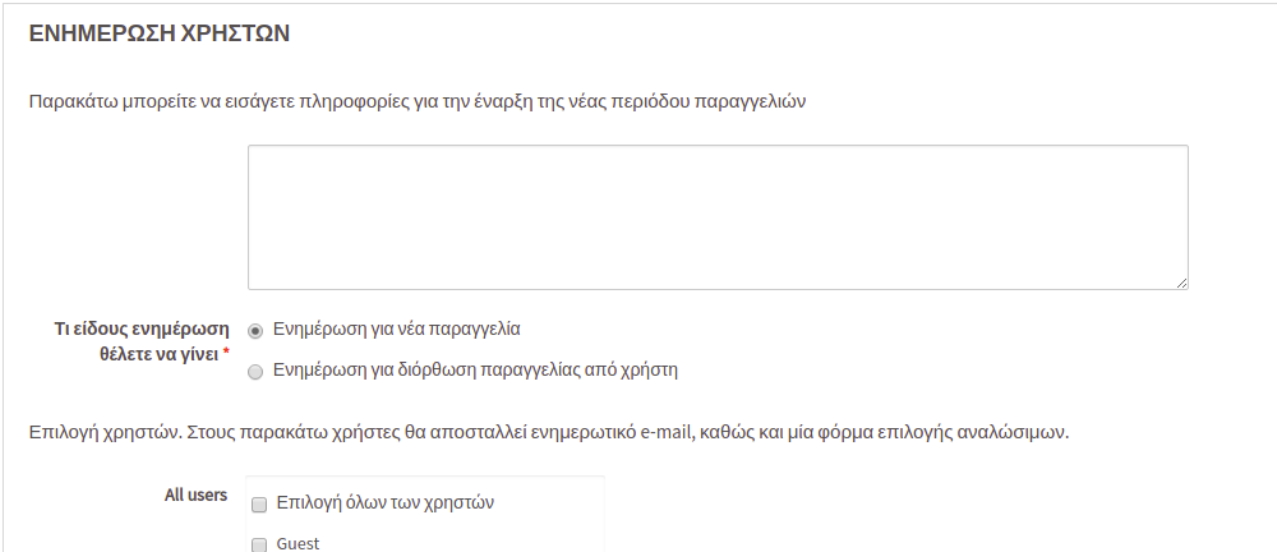
10.1.6.1.1.1 Task: Ενημερώνει τους χρήστες

1. Στην καρτέλα home, στα αριστερά πατάμε new case

2. Στην συνέχεια από τις διαθέσιμες διαδικασίες κάνουμε διπλό κλικ "Διαδικασία εισαγωγής αναλώσιμων από χρήστες"



3. Αυτομάτως ανοίγει μία φόρμα για συμπλήρωση.



Στο πρώτο πεδίο ο διαχειριστής μπορεί να εισάγει ένα μήνυμα το οποίο θα εμφανιστεί στις φόρμες όλων των χρηστών.

4.Στην συνέχεια επιλέγει τους χρήστες που θα πραγματοποιήσουν τις παραγγελίες τους.

5. Περιμένει τους χρήστες να συμπληρώσουν τις φόρμες τους.

<u>Σημαντικά σημεία για το στάδιο που περιμένουμε τους χρήστες:</u>

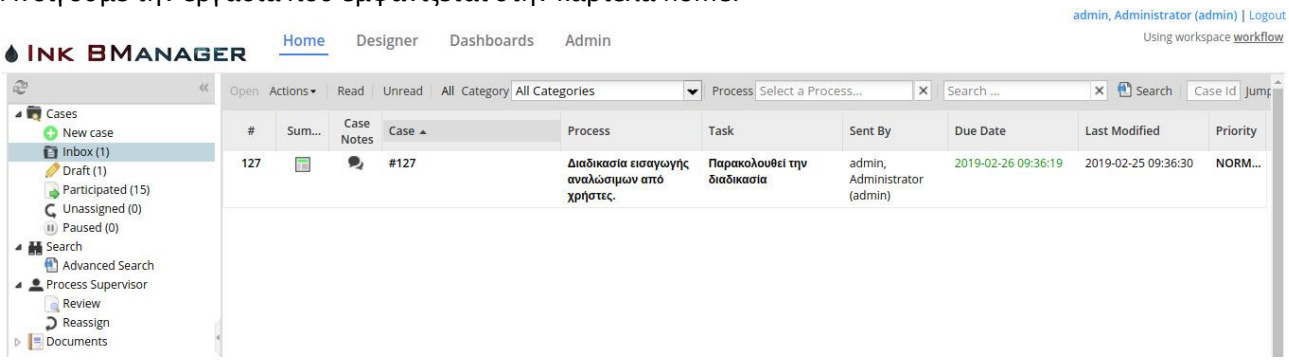
• Σε περίπτωση που κάποιος χρήστης χρειαστεί αλλαγή, από το new case ξεκινάμε το reopen.

• Για να προχωρήσει η διαδικασία και να δει ο διαχειριστής τις παραγγελίες των χρηστών, θα πρέπει όλοι χρήστες να έχουν συμπληρώσει την φόρμα τους. Υπάρχει περίπτωση κάποιος να μην συμπληρώσει την δική του και εν τέλει να μην μπορέσει να προχωρήσει η διαδικασία. Στο σημείο αυτό θα πρέπει από το new case να ξεκινήσει η διαδικασία **"**Τερματισμός διαδικασίας εισαγωγής αναλώσιμων από χρήστες**"**.

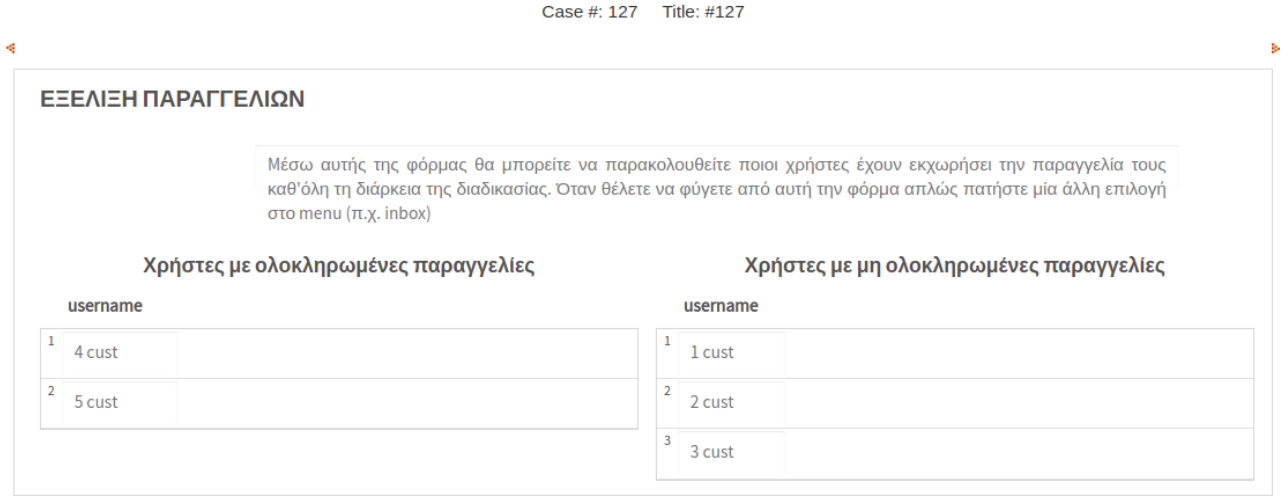
10.1.6.1.1.2 Παρακολουθεί την διαδικασία

Μετά την επιλογή χρηστών και καθώς ο διαχειριστής περιμένει τους χρήστες να συμπληρώσουν την φόρμα τους, μπορεί να παρακολουθεί ποιοι χρήστες έχουν συμπληρώσει ή όχι την φόρμα παραγγελίας. Για τον σκοπό αυτό στο inbox θα υπάρχει μία η εργασία "παρακολουθεί την διαδικασία".

1. Ανοίγουμε την εργασία που εμφανίζεται στην καρτέλα home.



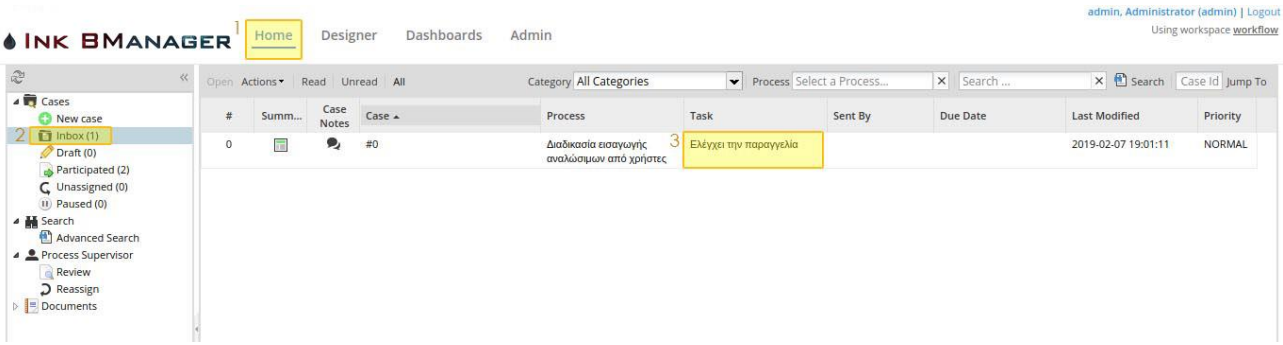
2. Μέσω τις φόρμας παρακολουθούμε την εξέλιξη της διαδικασίας. Για να φύγουμε από την φόρμα, απλώς πατάμε κάποια άλλη επιλογή στο menu (π.χ. inbox, home κτλ)



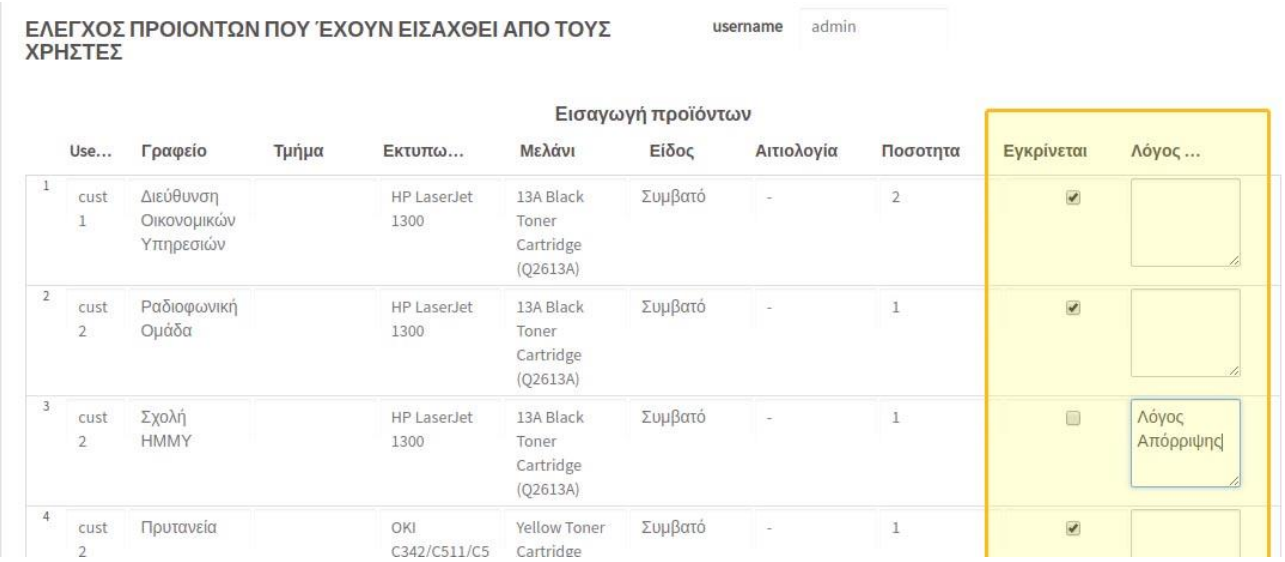
3. Όταν η διαδικασία των παραγγελιών ολοκληρωθεί και ο διαχειριστής θα πρέπει να ελέγξει την παραγγελία, στο inbox θα εμφανιστούν οι δύο εργασίες –ελέγχει την παραγγελία και παρακολουθεί την διαδικασία-. Εμείς πατάμε "ελέγχει την παραγγελία".

10.1.6.1.1.3 Task: Ελέγχει την παραγγελία

1. Στον διαχειριστή στην καρτέλα home εμφανίζεται η εργασία "ελέγχει την παραγγελία". Εδώ κάνουμε διπλό κλικ για να εκτελέσουμε την εργασία.



2. Σε μία φόρμα εμφανίζονται οι παραγγελίες των χρηστών και καλείτε να δεχτεί ή να απορρίψει τα προϊόντα, τσεκάρωντας ή αγνοώντας το πεδίο "Εγκρίνεται". Το πεδίο "λόγος απόρριψης" πρέπει να συμπληρωθεί για την ενημέρωση του χρήστη σε περίπτωση απόρριψης. Τέλος ο διαχειριστής θα πρέπει να επιλέξει πως θα συνεχίσει (διόρθωση παραγγελίας από τους χρήστες ή τελική εκχώρηση των επιλεγμένων προϊόντων).

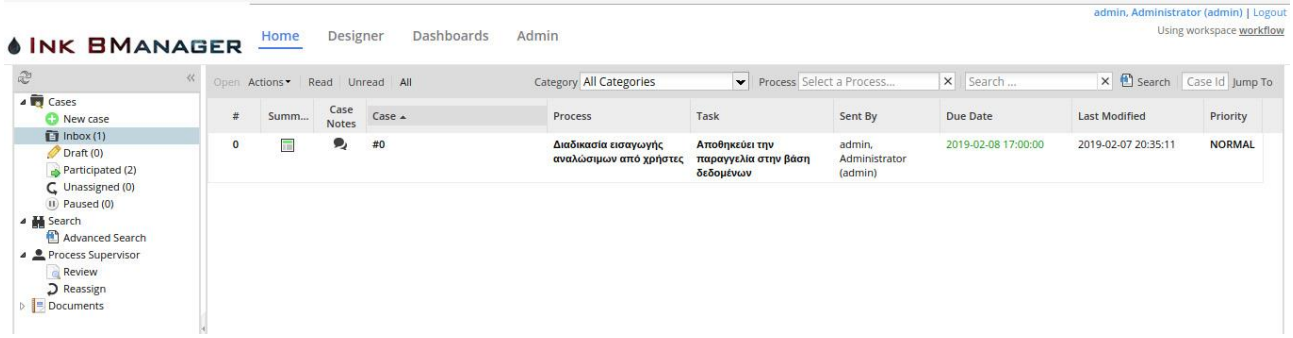


10.1.6.1.1.3.1 Περίπτωση που ο χρήστης έχει ζητήσει μελάνι που δεν υπάρχει στην βάση δεδομένων.

Όπως αναφέρεται και στις οδηγίες στην φόρμα, σε περίπτωση που υπάρχει κάποιο προϊόν που έχει ζητηθεί να προστεθεί από τον χρήστη διότι δεν υπήρχε στην βάση δεδομένων, θα πρέπει ο διαχειριστής μετά την υποβολή της φόρμας να μεταβεί στην εργασία "προσθήκη νέου εκτυπωτή αναλώσιμου γραφείου εργαστηρίου". Εκεί θα πρέπει να εκχωρήσει το όνομα του προϊόντος και του εκτυπωτή και στην συνέχεια αυτά θα σωθούν στην βάση δεδομένων. Έχοντας σώσει τα εν λόγω προϊόντα με το επίσημο όνομα θα μπορέσει να συνεχίσει.

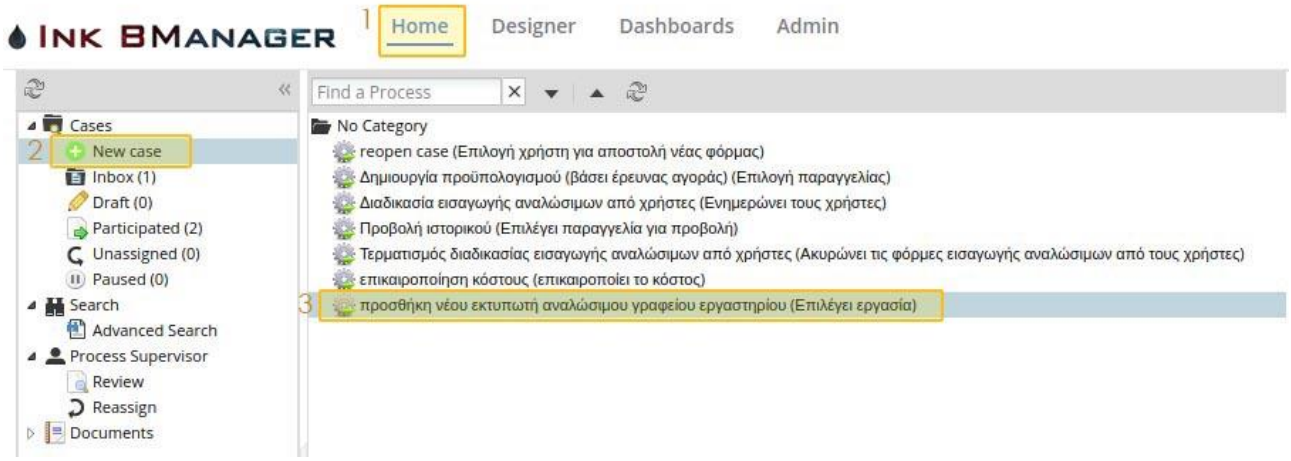
Βήμα 1:

Αγνοούμε την εργασία που μας εμφανίζει το σύστημα.



Βήμα 2:

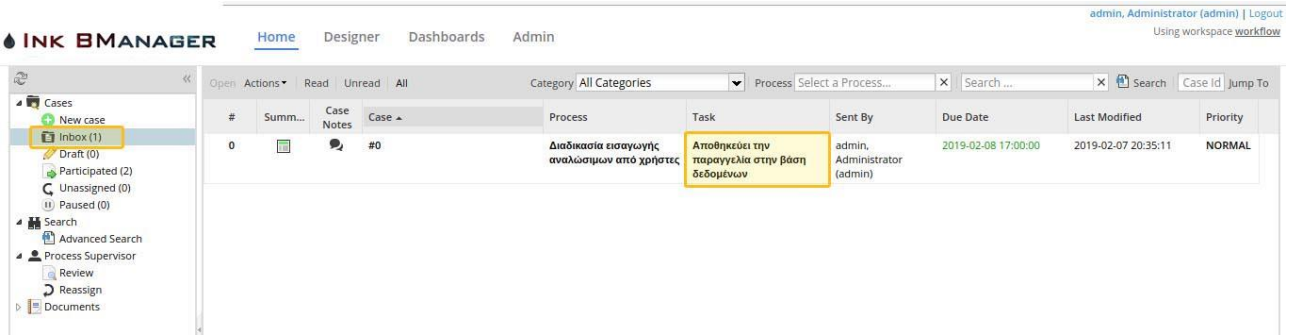
Ξεκινάμε νέα διαδικασία προσθήκης αναλώσιμου.



Βήμα 3:

Αφού τελειώσουμε την διαδικασία προσθήκης, εκτελούμε την εργασία που αγνοήσαμε στο Βήμα 1.

10.1.6.1.1.4 Task: Αποθηκεύει την παραγγελία στην βάση δεδομένων



1. Με διπλό κλικ εκτελούμε την εργασία.

2. Βλέπουμε την παραγγελία. Το σύστημα έχει συγκεντρώσει όλα τα προϊόντα που έχουν δηλώσει οι χρήστες και τα έχει χωρίσει σε συμβατά και γνήσια.
Επίσης στο πεδίο "επιλογές χρηστών για τις οποίες χρειάστηκε νέα προσθήκη στο σύστημα" φαίνονται τα νέα μελάνια που προστέθηκαν με το όνομα που τους έχει δώσει ο admin.

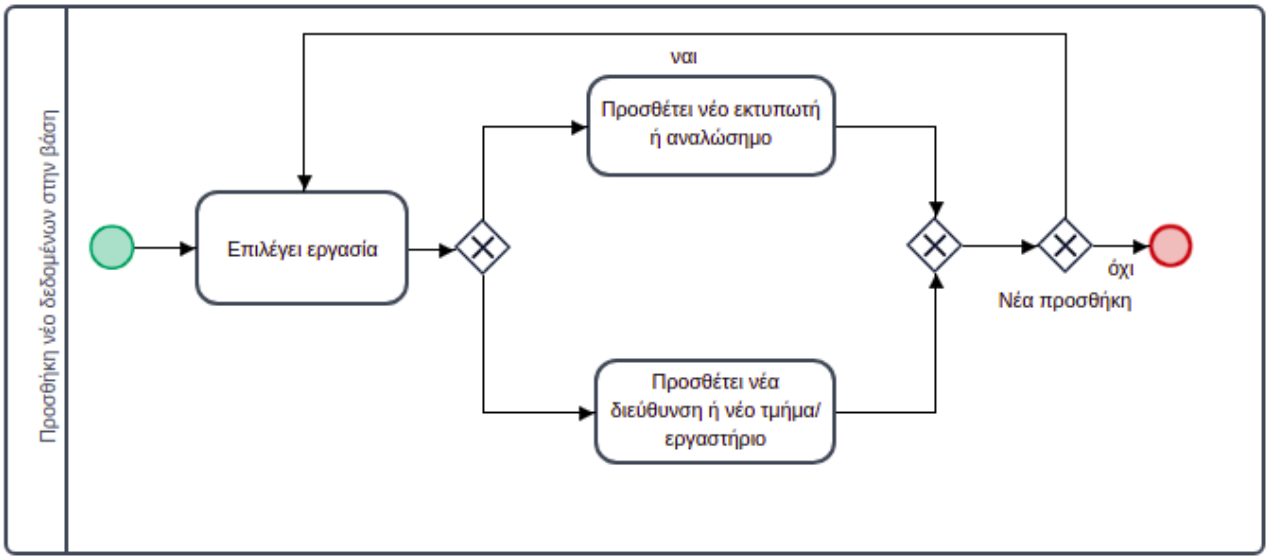
Επιλογές Χρηστών

	Μελάνι	Σύνολο Συμβ...	Σύνολο Γνήσ...
1	13A Black Toner Cartridge (Q2613A)	4	0
2	Black Toner Cartridge (45807102)	2	0
3	MX-235 Black Toner Cartridge	1	0
4	Yellow Toner Cartridge (44469704)	1	0

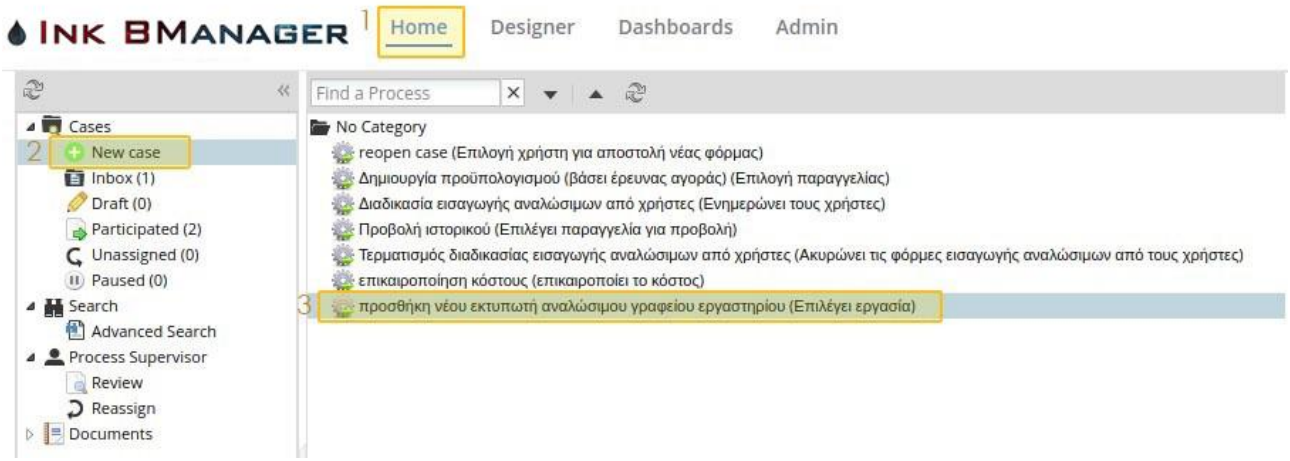
Επιλογές Χρηστών για τις οποίες χρειάστηκε νέα προσθήκη στο σύστημα

	Μελάνι	Σύνολο Συμβ...	Σύνολο Γνήσ...
1	HP toner cartridge Black (CF283A, 83A)	1	0
2	HP toner cartridge Cyan	1	0

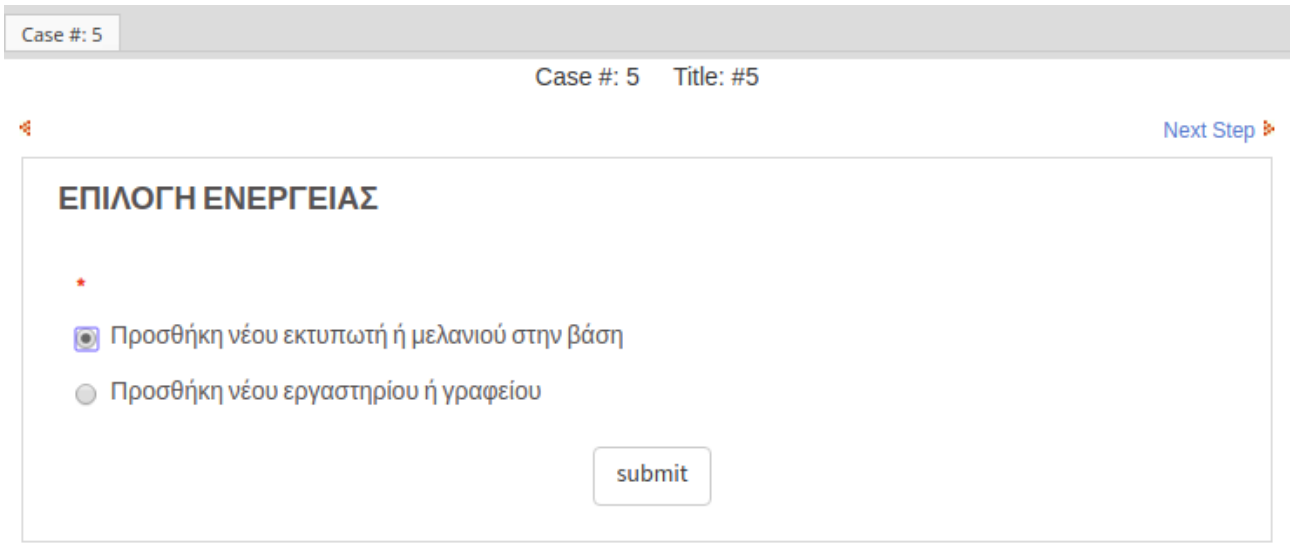
10.1.6.2 Διαδικασία "προσθήκη νέου εκτυπωτή αναλώσιμου γραφείου εργαστηρίου"



1. Στην καρτέλα home στα αριστερά επιλέγουμε new case. Από τις διαθέσιμες επιλογές κάνουμε διπλό κλικ στο "προσθήκη νέου εκτυπωτή αναλώσιμου γραφείου εργαστηρίου".



2. Μας ανοίγει μία φόρμα και επιλέγουμε αν θέλουμε να προσθέσουμε νέο εκτυπωτή/αναλώσιμο ή νέα διεύθυνση/τμήμα.

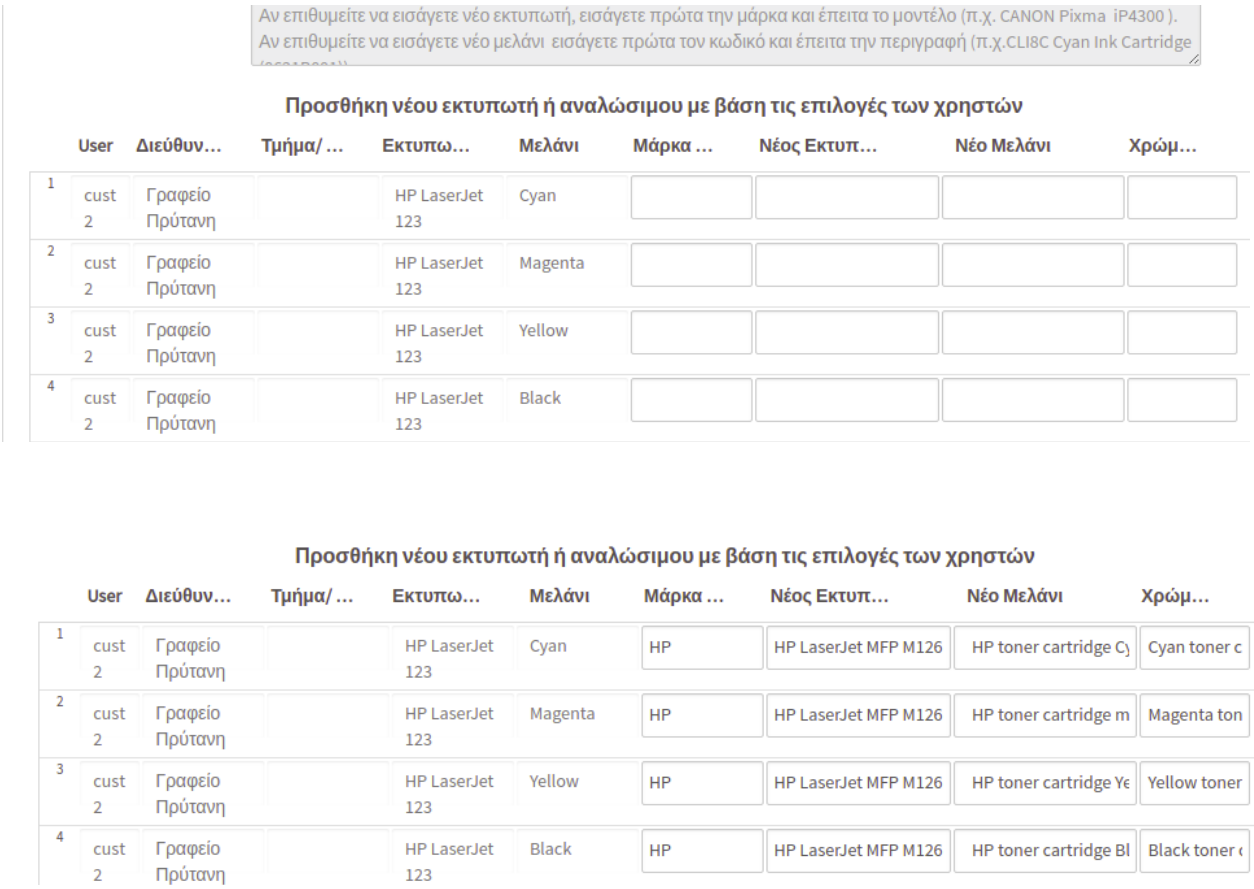


10.1.6.2.1 Προσθήκη Νέου εκτυπωτή/ αναλώσιμου

Εδώ έχουμε δύο επιλογές. Είτε να προσθέσουμε κάποιον εκτυπωτή/ αναλώσιμο βάσει της παραγγελίας ενός χρήστη, είτε να κάνουμε μία ανεξάρτητη προσθήκη (π.χ. σκεφτήκαμε ότι θα έπρεπε να προστεθεί ο εκτυπωτής HP LaserJet Pro M402dn).

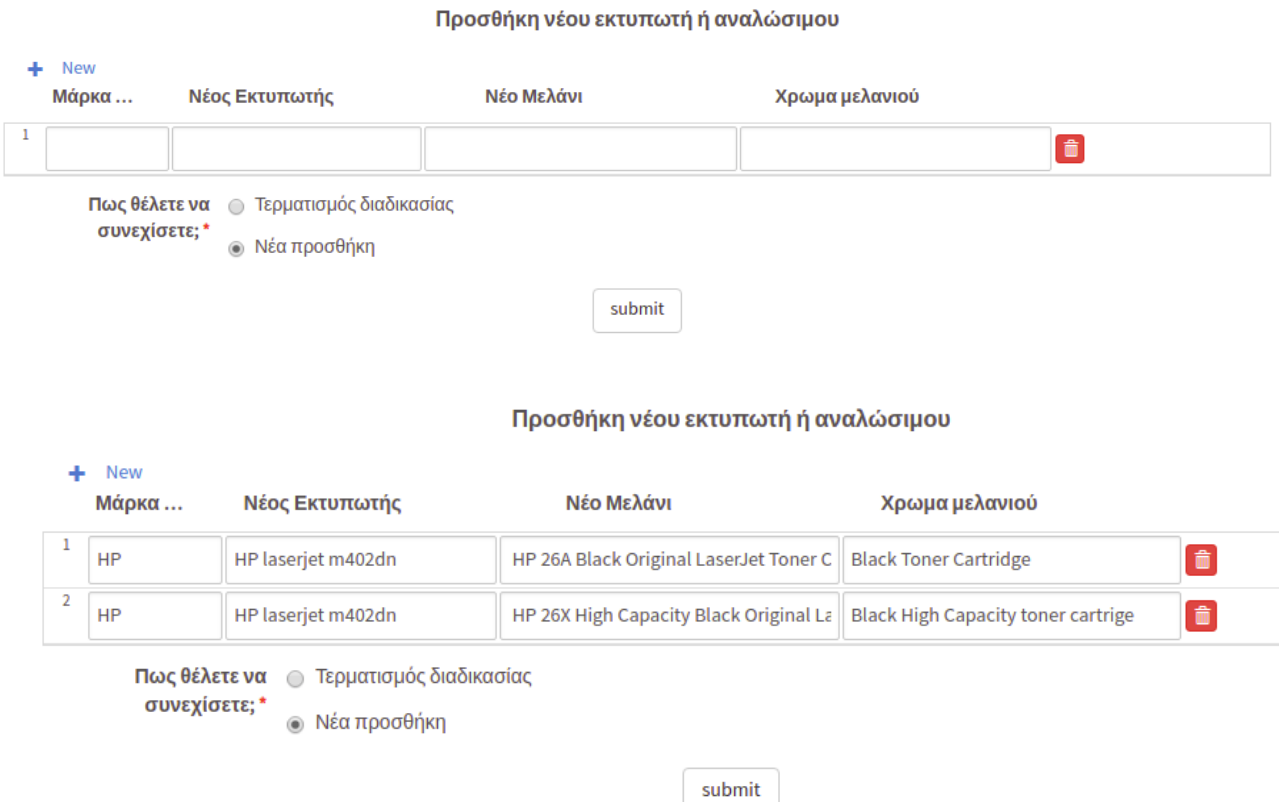
10.1.6.2.1.1 Προσθήκη με βάση την παραγγελία ενός χρήστη.

Εδώ έχει συμπληρωθεί ένας πίνακας που μας παρουσιάζει τις επιλογές των χρηστών. Βάσει αυτών των επιλογών εμείς εκχωρούμε τον εκτυπωτή και τα αναλώσιμα με τα ονόματα που θέλουμε να υπάρχουν στην βάση. Δεν ξεχνάμε να συμπληρώσουμε το πεδίο χρώμα, αφού αυτό είναι το πεδίο που θα βλέπει ο χρήστης όταν επιλέγει ένα αναλώσιμο.



10.1.6.2.1.2 Ανεξάρτητη προσθήκη

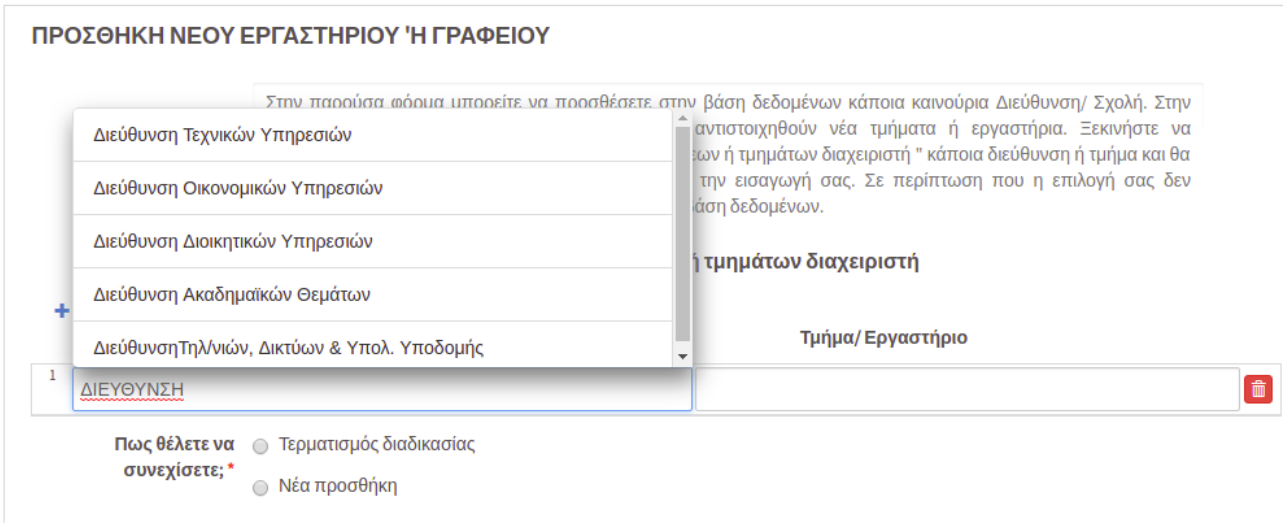
Εδώ συμπληρώνουμε το πεδίο "προσθήκη νέου εκτυπωτή ή αναλώσιμου".



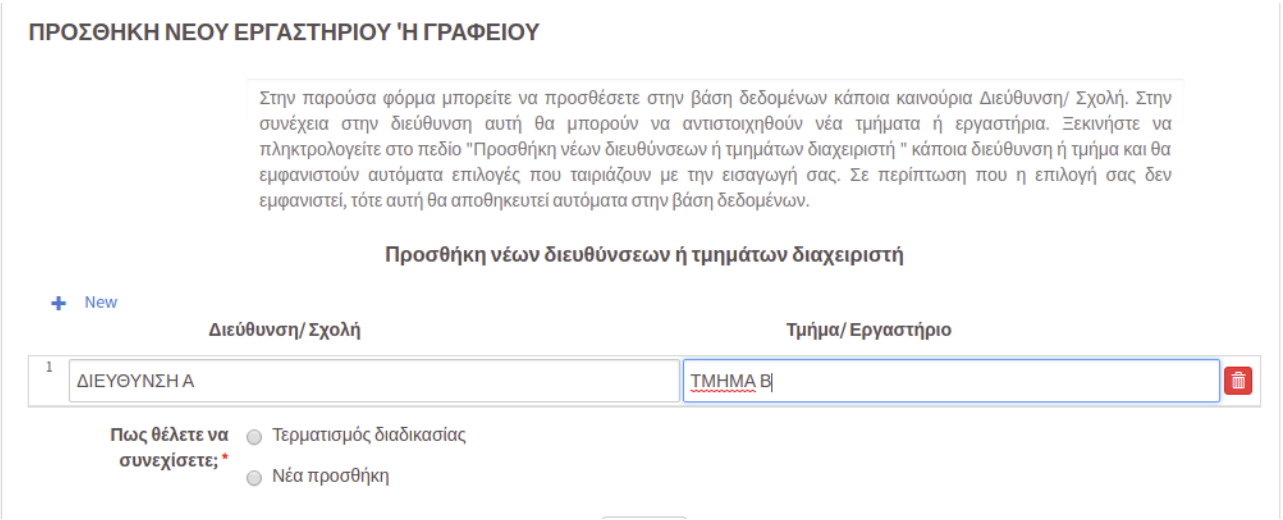
10.1.6.2.2 Προσθήκη νέας Διεύθυνσης ή Τμήματος εργαστηρίου

Συμπληρώνουμε την φόρμα με τις διευθύνσεις και τα τμήματα που θέλουμε να προσθέσουμε. Εδώ πρέπει να σημειωθεί ότι στην βάση υπάρχουν οι διευθύνσεις και τα τμήματα όπως παρουσιάζονται στο οργανόγραμμα του Πολυτεχνείου Κρήτης. Επίσης όταν κάποιος χρήστης που συμπληρώνει μια φόρμα παραγγελίας και εκχωρήσει ένα νέο τμήμα ή μία νέα διεύθυνση (που δεν υπάρχει στα dropdown menus), τότε αυτομάτως αυτή εκχωρείται στην βάση δεδομένων.

1. Ελέγχουμε αν η διεύθυνση ή το τμήμα που πάμε να εκχωρήσουμε υπάρχει ήδη στην βάση. Ενώ πληκτρολογούμε το σύστημα μας προτείνει επιλογές που ταιριάζουν με την εισαγωγή μας.

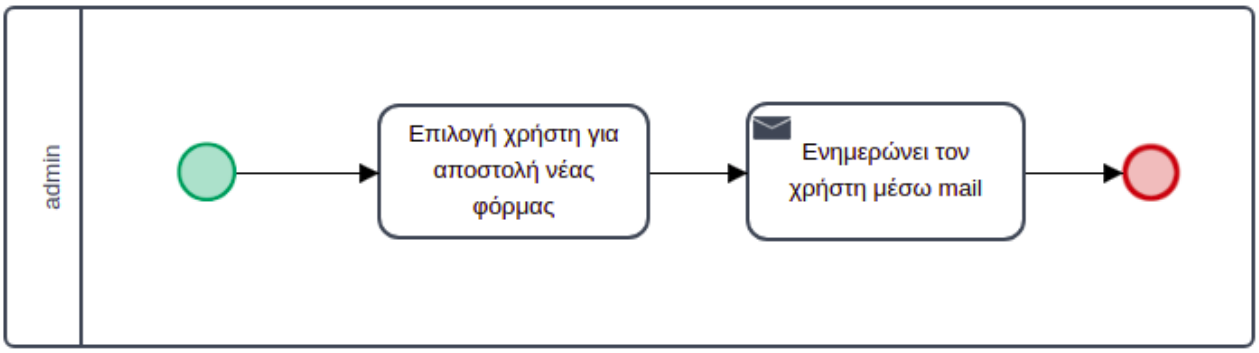


2. Αν όντως δεν υπάρχει η εκχώρηση μας, πατώντας "Υποβολή" αποθηκεύουμε τις διευθύνσεις και τα τμήματα στην βάση



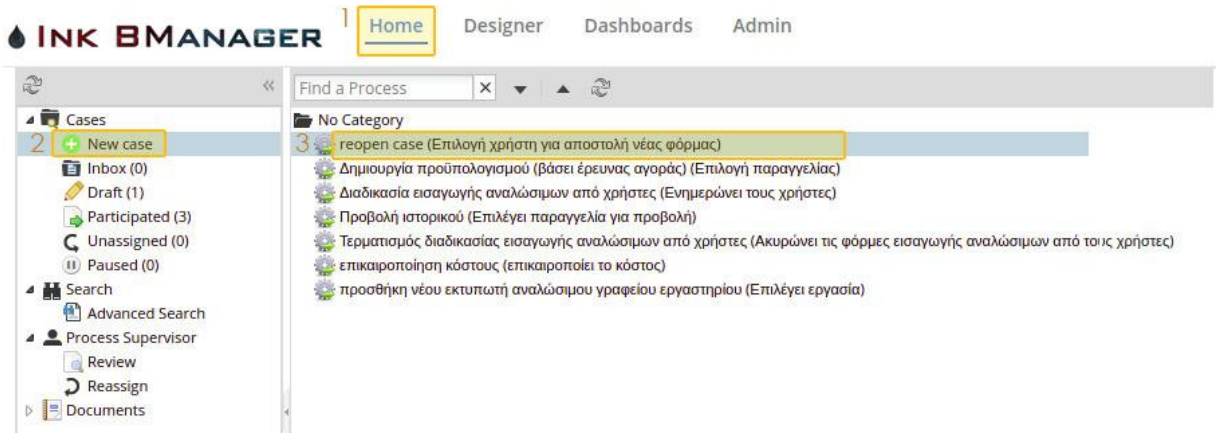
3. Επιλέγουμε πως θέλουμε να συνεχίσουμε.

10.1.6.3 Διαδικασία "Reopen Case"

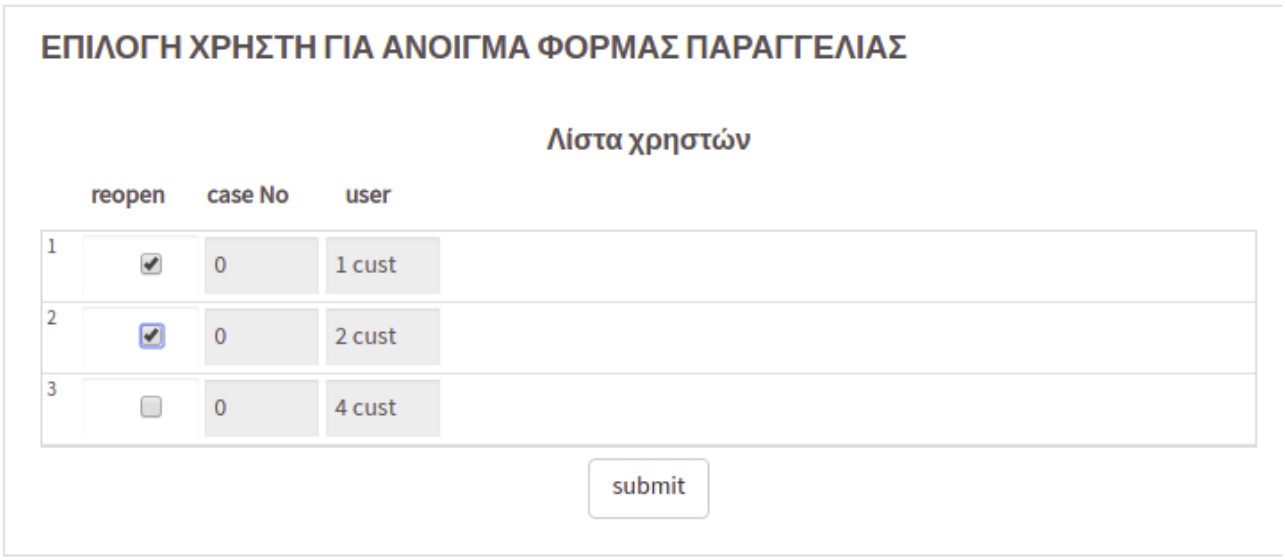


Η διαδικασία αυτή χρησιμοποιείται κατά την διάρκεια της "διαδικασίας εισαγωγής αναλώσιμων από τους χρήστες". Έχει σχεδιαστεί για να ξαναδίνει πρόσβαση στους χρήστες στην παραγγελία τους, ώστε να μπορούν να προσθέσουν νέα προϊόντα ή να αλλάξουν κάτι (π.χ. μία ποσότητα).

1. Από την καρτέλα home στα αριστερά πατάμε new case και με διπλό κλικ επιλέγουμε από τις διαθέσιμες διαδικασίες το reopen case.

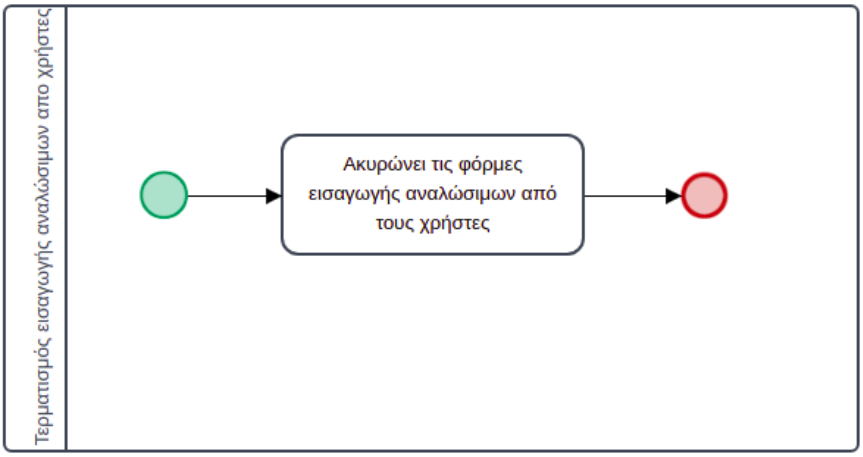


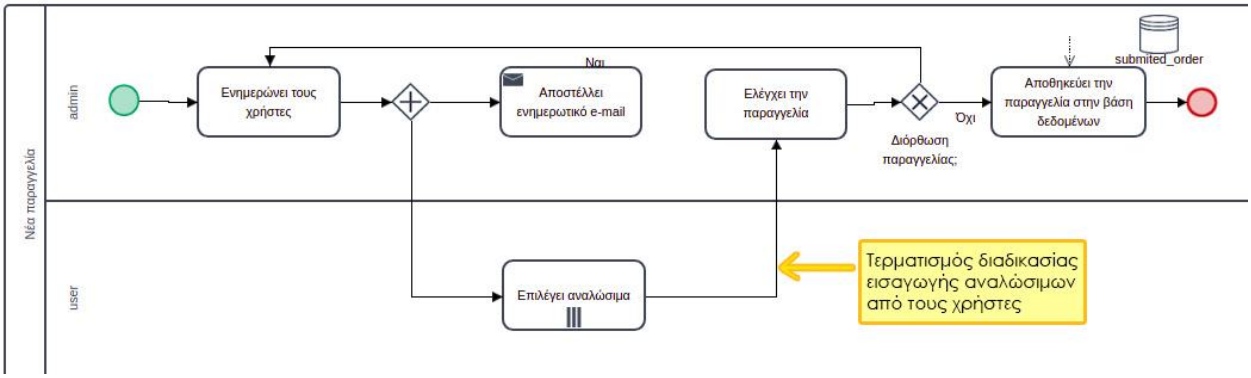
2. Μας εμφανίζεται μία φόρμα. Σε αυτή την φόρμα διαλέγουμε τους χρήστες των οποίων θα ανοίξουμε την φόρμα παραγγελίας.



3. Αυτομάτως αποστέλλεται σε αυτούς ένα ενημερωτικό email πως μπορούν να αλλάξουν την παραγγελία τους.

10.1.6.4 Διαδικασία "Τερματισμός διαδικασίας εισαγωγής αναλώσιμων από τους χρήστες"

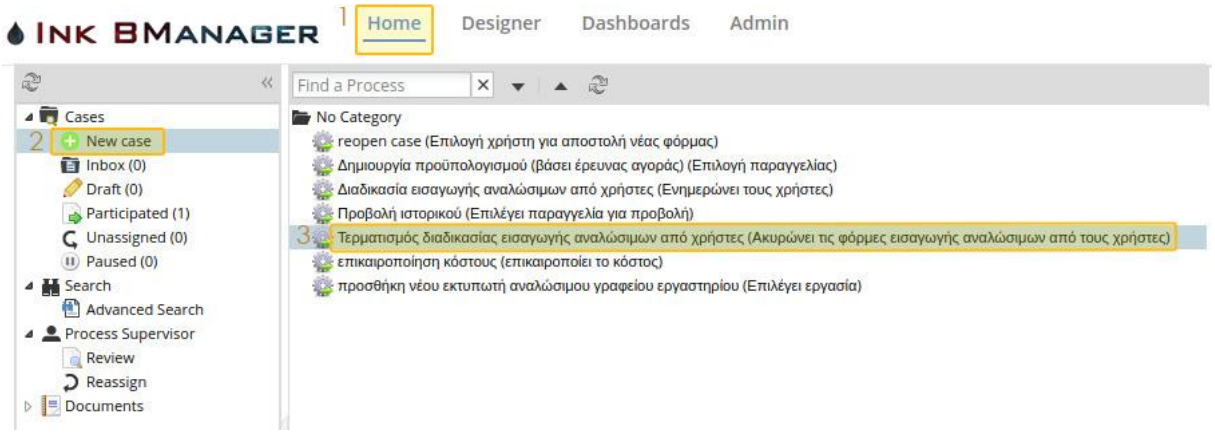




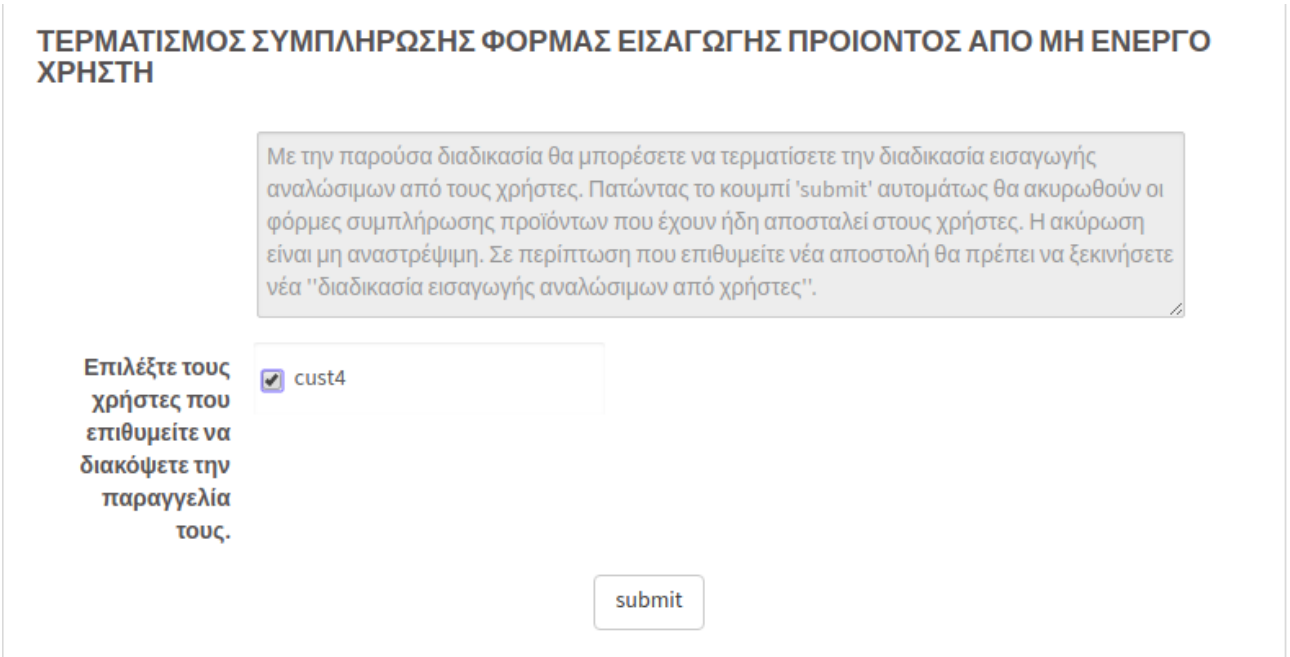
Όπως αναφέρθηκε παραπάνω για να μεταβεί ο διαχειριστής στην εργασία "ελέγχει την παραγγελία" της διαδικασίας εισαγωγής αναλώσιμων από χρήστες, θα πρέπει όλοι οι χρήστες να έχουν συμπληρώσει την

παραγγελία τους. Υπάρχει περίπτωση κάποιος χρήστης να μην συμπληρώσει την δική του και έτσι ο διαχειριστής να μην μπορέσει να ελέγξει της παραγγελίες όλων των υπόλοιπων χρηστών. Για την αποφυγή μιας τέτοια κατάστασης έχει δημιουργηθεί αυτή η διαδικασία. Μπορούμε να την εκτελέσουμε όποτε θέλουμε να σταματήσει η διαδικασία υποβολή παραγγελιών από τους χρήστες.

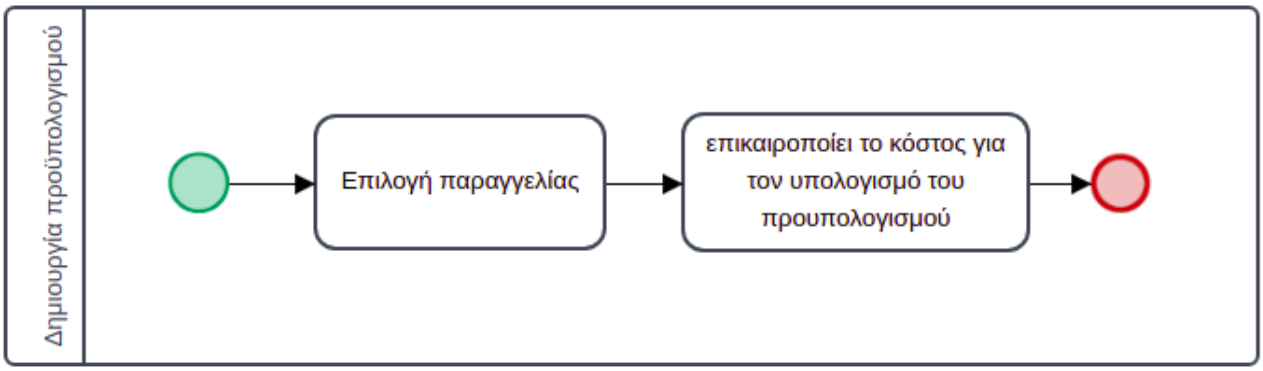
1. Από την καρτέλα home πατάμε new case και από τις διαθέσιμες διαδικασίες κάνουμε διπλό κλικ στην διαδικασία "τερματισμός διαδικασίας εισαγωγής αναλώσιμων από χρήστες"



2. Μας εμφανίζεται μία φόρμα. Στην φόρμα αυτή υπάρχουν οι users οι οποίοι έχουν ανοιχτές παραγγελίες. Μπορούμε να κλείσουμε την φόρμα παραγγελίας σε ένα μόνο χρήστη ή σε όλους. Σε περίπτωση που κλείσουμε τις φόρμες όλων των χρηστών, θα προχωρήσει η βασική διαδικασία.



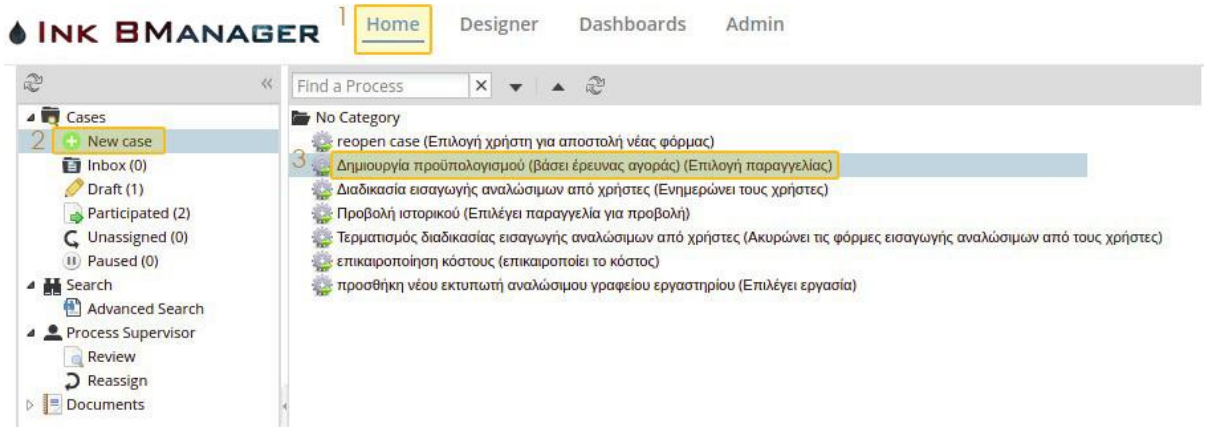
10.1.6.5 Διαδικασία "Δημιουργία προϋπολογισμού βάσει έρευνας αγοράς"



Στην διαδικασία αυτή ο διαχειριστής κοστολογεί τα προϊόντα βάσει έρευνα αγοράς.

Να σημειωθεί ότι μπορούμε να δημιουργήσουμε ένα προϋπολογισμό για μία νέα παραγγελία, ή να δούμε έναν παλαιότερο προϋπολογισμό. Επίσης για μία τρέχουσα παραγγελία, εφόσον έχουν εκχωρήσει τα κόστη των προϊόντων μπορούμε να τροποποιήσουμε τις ήδη υπάρχουσες τιμές.

1. Στην καρτέλα home πατάμε new case και κάνουμε διπλό κλικ στην δημιουργία προϋπολογισμού βάσει έρευνα αγοράς.

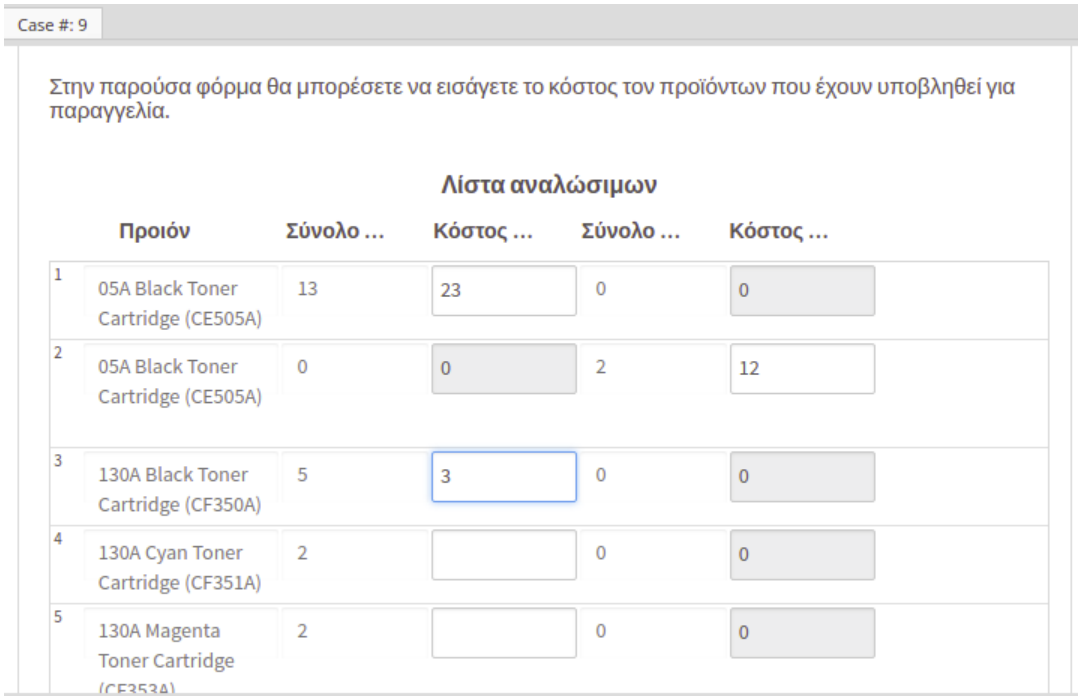


2. Επιλέγουμε ποια παραγγελία θα δούμε/ επεξεργαστούμε. Αν η παραγγελία είναι καινούρια τότε πατάμε κατάρτιση νέου προϋπολογισμού. Εάν θέλουμε να αλλάξουμε την τιμή κάποιου προϊόντος σε μία ήδη υπάρχουσα παραγγελία επιλέγουμε την συγκριμένη παραγγελία.

Πατάμε continue μέχρι να εμφανιστεί η εργασία "επικαιροποιεί το κόστος για τον υπολογισμό του προϋπολογισμού" στην αρχική μας οθόνη. Επιλέγουμε με διπλό κλικ την εργασία και στην φόρμα που εμφανίζεται καταχωρούμε τις τιμές.

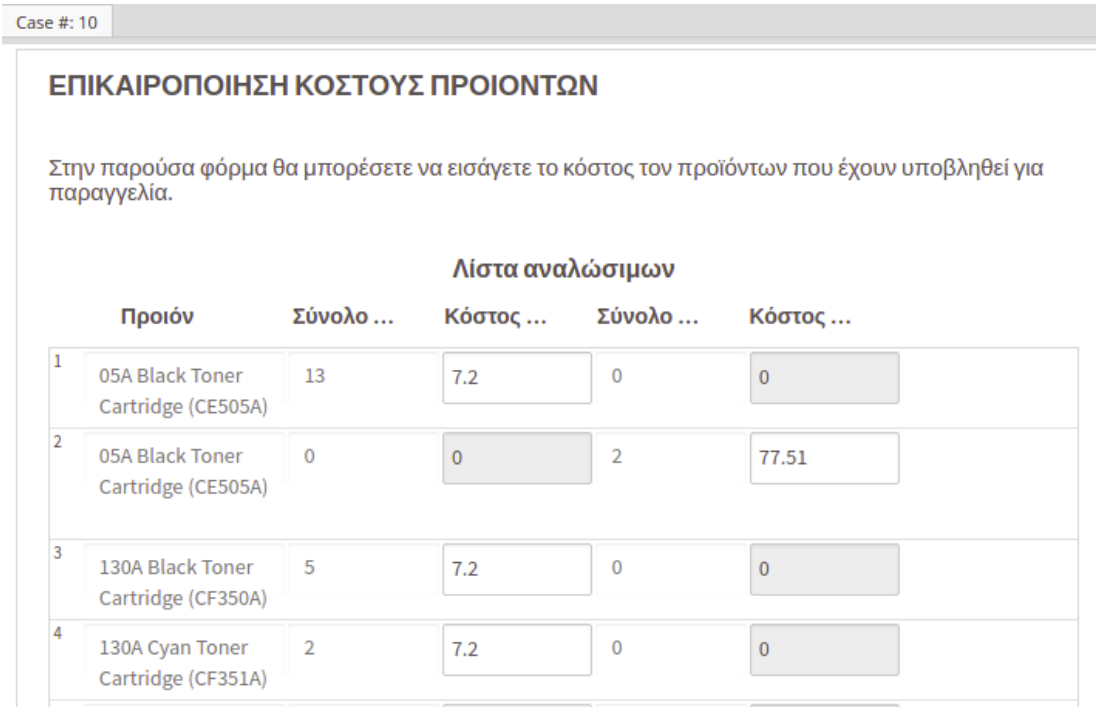
10.1.6.5.1 Νέα Παραγγελία (Κατάρτιση νέου προϋπολογισμού):

Το σύστημα μπορεί να παρουσιάζει αυτόματα τις αντίστοιχες τιμές των μελανοδοχείων, όπως έχουν καταχωρηθεί σε προηγούμενες παραγγελίες. Οι εμφανιζόμενες τιμές είναι οι τελευταίες εκχωρήσεις που υπάρχουν στην βάση.



Ήδη υπάρχουσα παραγγελία:

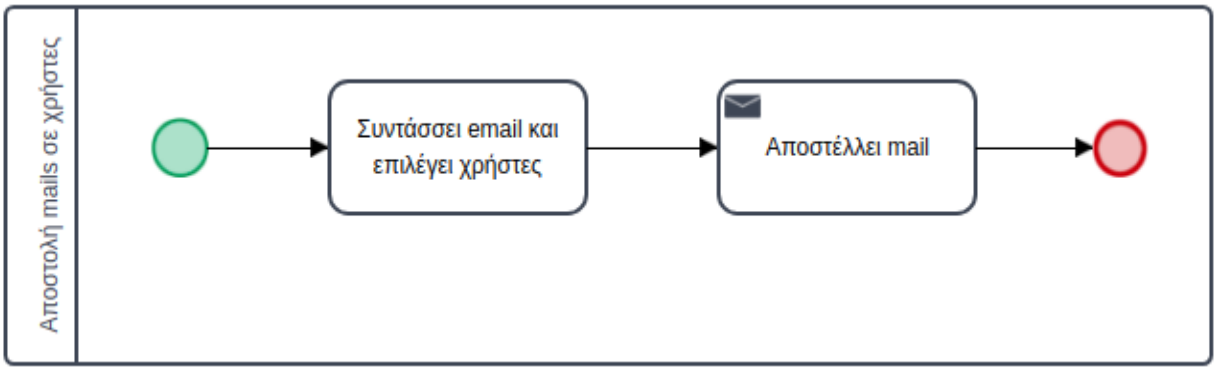
Εδώ μας παρουσιάζεται μία ήδη συμπληρωμένη φόρμα και αλλάζουμε όποια τιμή χρειάζεται αλλαγή.



3. Τσεκάρουμε το πεδίο "Θέλετε να σώσετε αυτό τον προϋπολογισμό στο σύστημα;" ώστε να αποθηκεύσουμε τα δεδομένα της φόρμας στην βάση.

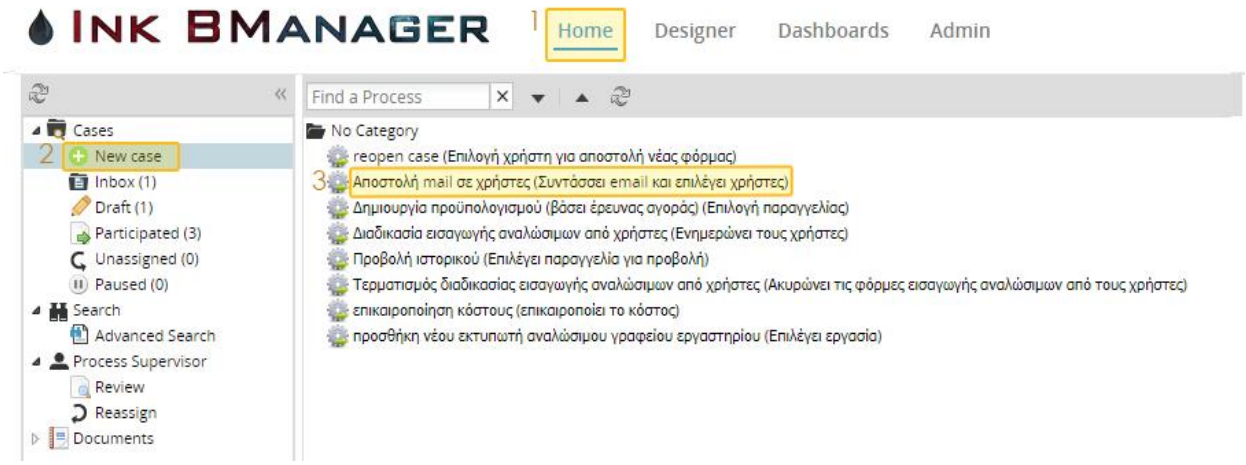
Εάν θέλουμε απλά να δούμε την παραγγελία δεν το τσεκάρουμε.

10.1.6.6 Αποστολή email σε χρήστες

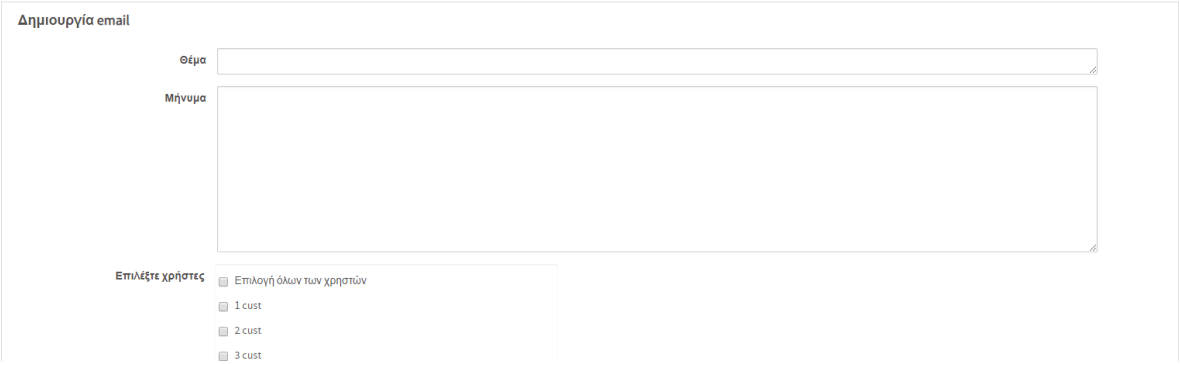


Με την παρούσα διαδικασία ο διαχειριστής μπορεί να αποστέλλει ομαδικά ή μεμονωμένα email στους χρήστες μέσω της πλατφόρμας.

1. Στην καρτέλα home, επιλέγουμε new case και στην εμφανιζόμενη λίστα διαλέγουμε με διπλό κλικ την διαδικασία "αποστολή email σε χρήστες"



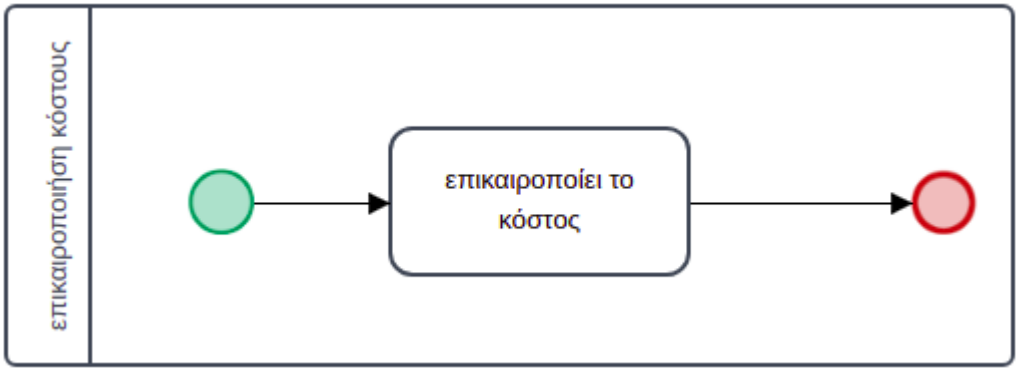
2. Αυτομάτως μας ανοίγει η παρακάτω φόρμα.



3. Αρχικά εισάγουμε το θέμα του email και έπειτα το μήνυμα μας.

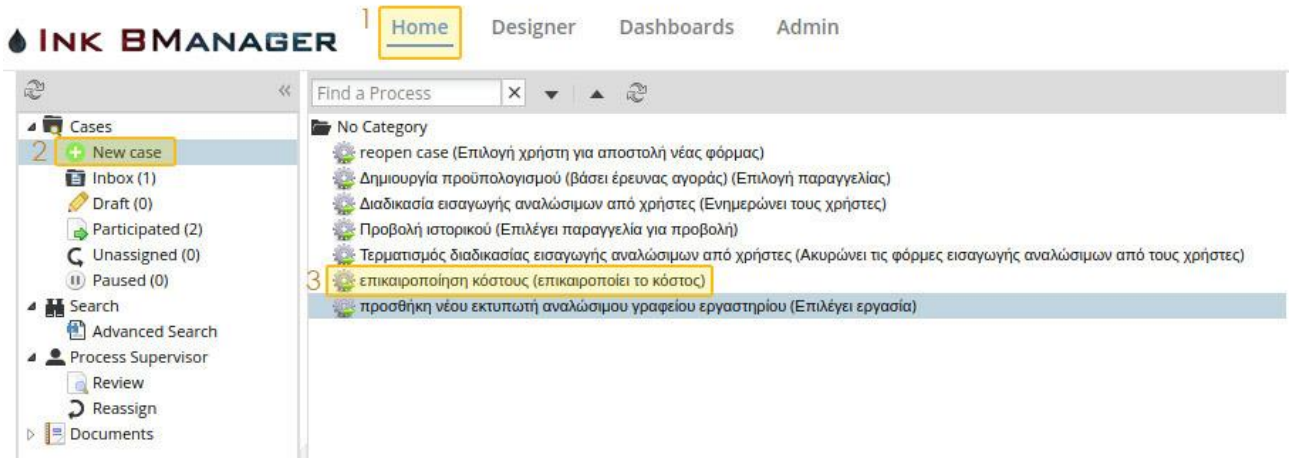
4. Τέλος επιλέγουμε έναν ή περισσότερους χρήστες και πατάμε υποβολή.

10.1.6.7 6.7 Διαδικασία "επικαιροποίηση κόστους"

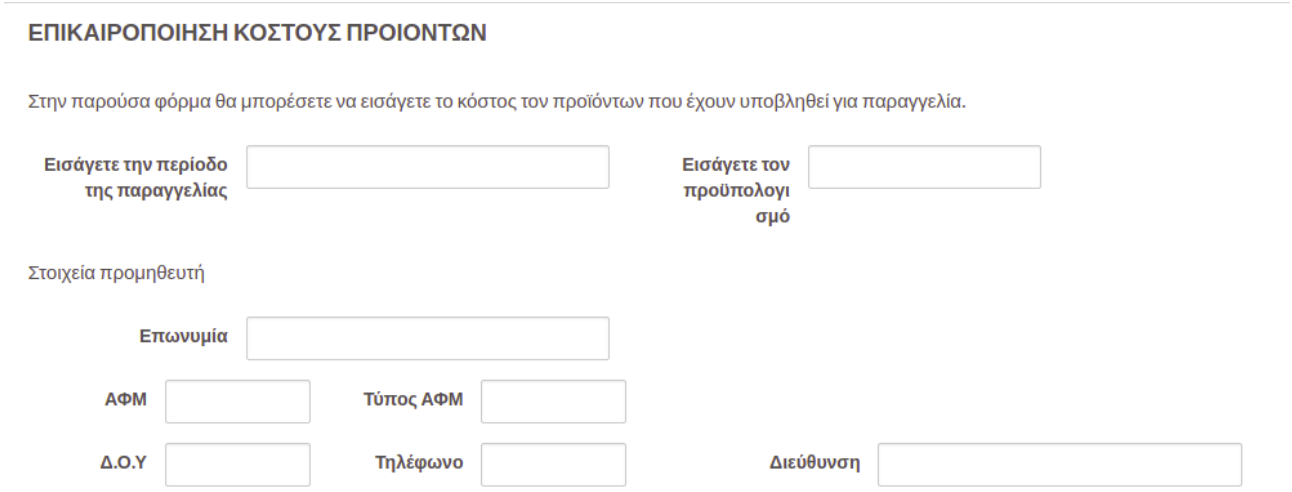


Στην διαδικασία αυτή εκχωρούνται όλα τα στοιχεία της παραγγελίας (στοιχεία προμηθευτή και τιμές προϊόντων) μετά το πέρας του διαγωνισμού. Είναι η τελική εκχώρηση των στοιχείων της παραγγελίας και πρέπει να γίνει με προσοχή.

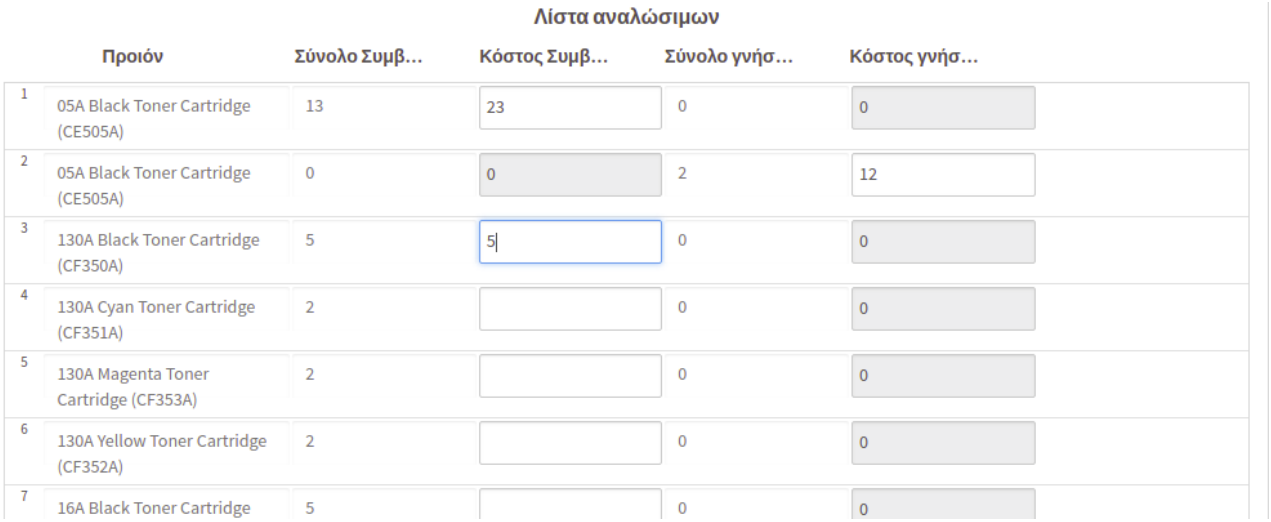
1. Από την καρτέλα home επιλέγουμε new case και στην συνέχεια με διπλό κλικ στο "επικαιροποίηση κόστους" ξεκινάμε την διαδικασία.



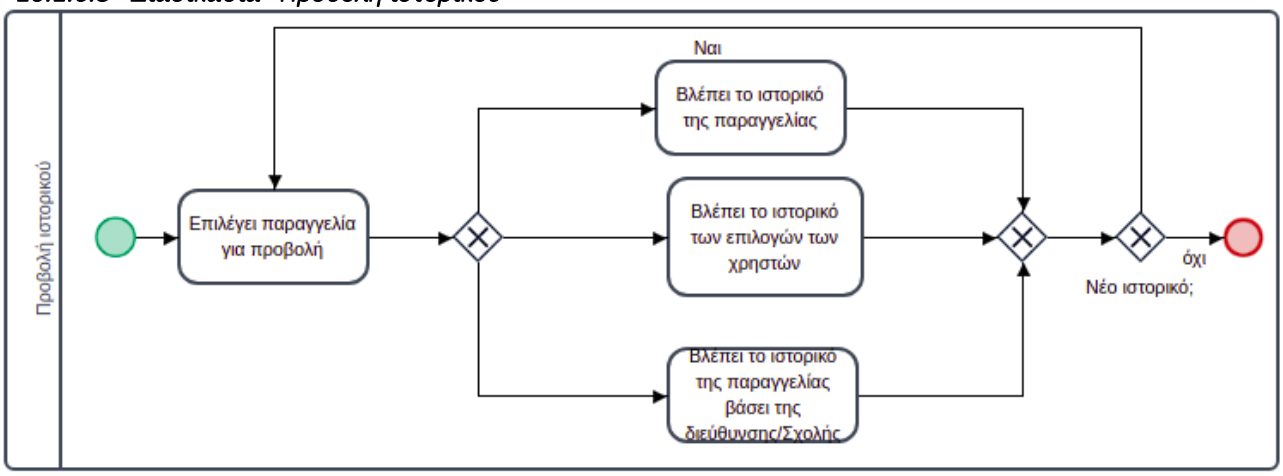
2. Μας εμφανίζεται μία φόρμα. Αρχικά συμπληρώνουμε τα στοιχεία της παραγγελίας και τα στοιχεία του προμηθευτή.



3. Στο πεδίο "Λίστα αναλώσιμων" εκχωρούμε τις τιμές των προϊόντων όπως αυτές διαμορφώθηκαν στον διαγωνισμό.

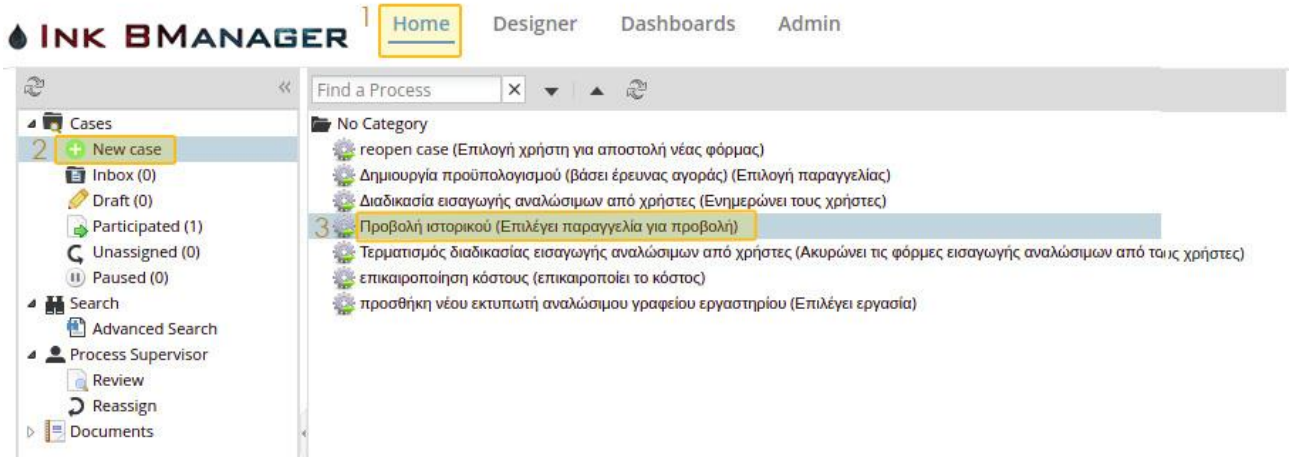


10.1.6.8 Διαδικασία "Προβολή ιστορικού"

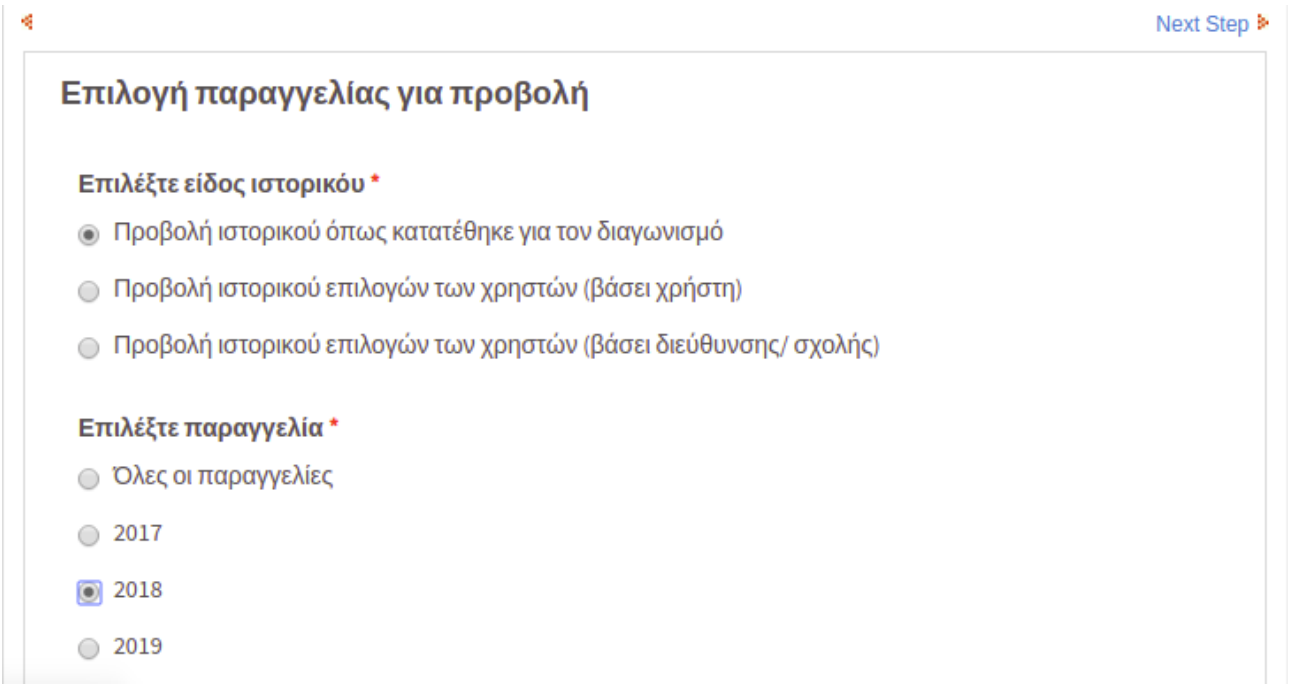


Στην διαδικασία αυτή μπορούμε να δούμε τις παλαιότερες παραγγελίες. Μπορούμε να τις δούμε όπως διαμορφωθήκαν στον διαγωνισμό (προβολή στοιχείων προμηθευτή και τελικές τιμές προϊόντων) ή να τις δούμε έχοντας επιλέξει κάποιον συγκεκριμένο χρήστη ή τμήμα, ώστε να βρούμε την κατανάλωση.

1. Στην καρτέλα home πατάμε new case και με διπλό κλικ επιλέγουμε την διαδικασία "προβολή ιστορικού".



2. Στην φόρμα που μας εμφανίζεται επιλέγουμε ποια παραγγελία επιθυμούμε να δούμε.



3. Πατάμε "Υποβολή" και continue μέχρι να εμφανιστεί στην αρχική μας οθόνη η εργασία "βλέπει το ιστορικό της παραγγελίας", την οποία και εκτελούμε με διπλό κλικ.

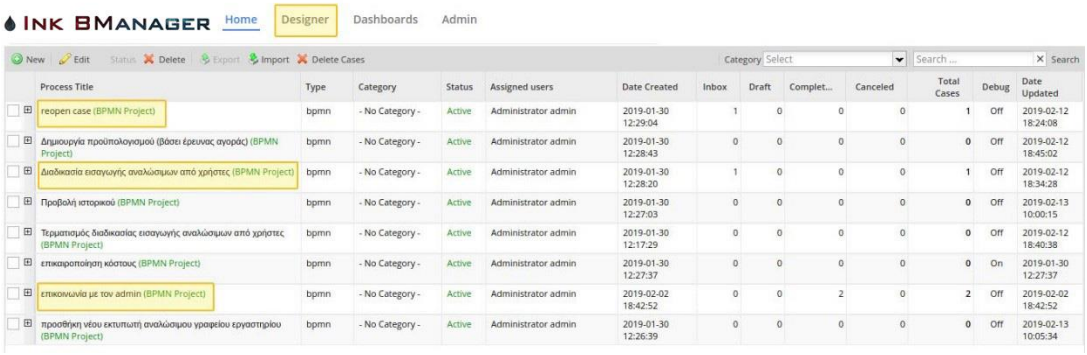
4. Βλέπουμε την παραγγελία που επιλέξαμε.

10.1.6.9 Διαχείριση email

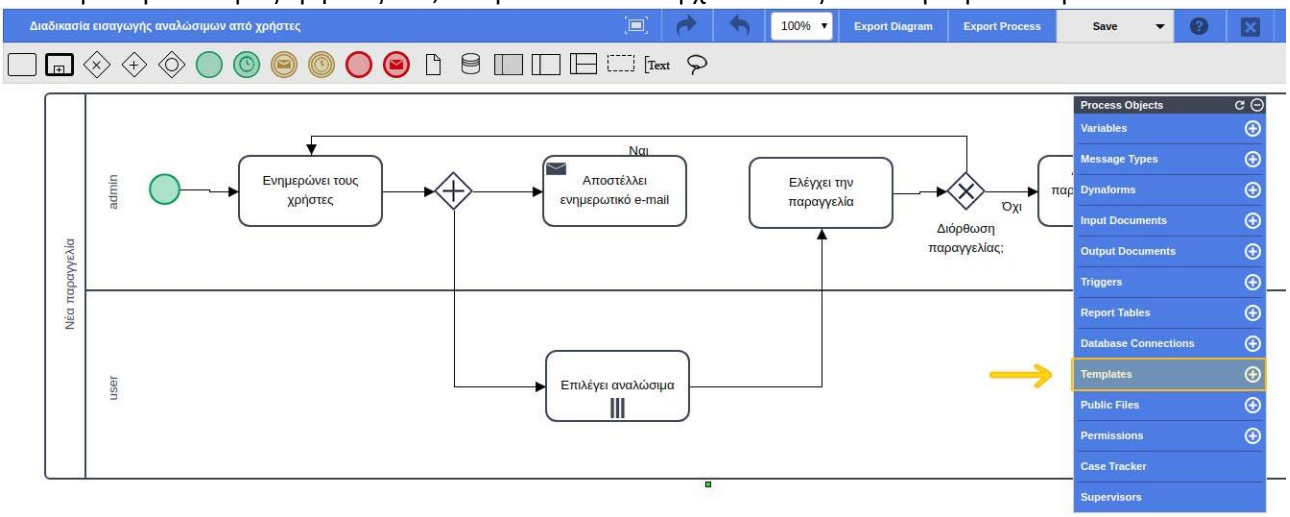
Κατά την διάρκεια της διαδικασίας παραγγελιών αποστέλλονται αυτόματα αρκετά mail. Τα mail αυτά είναι προ-προγραμματισμένα και δεν μπορούν να αλλάξουν εύκολα. Παρακάτω παρουσιάζονται τα βήματα για την αλλαγή τους.

10.1.6.9.1 Αλλαγή κειμένου email

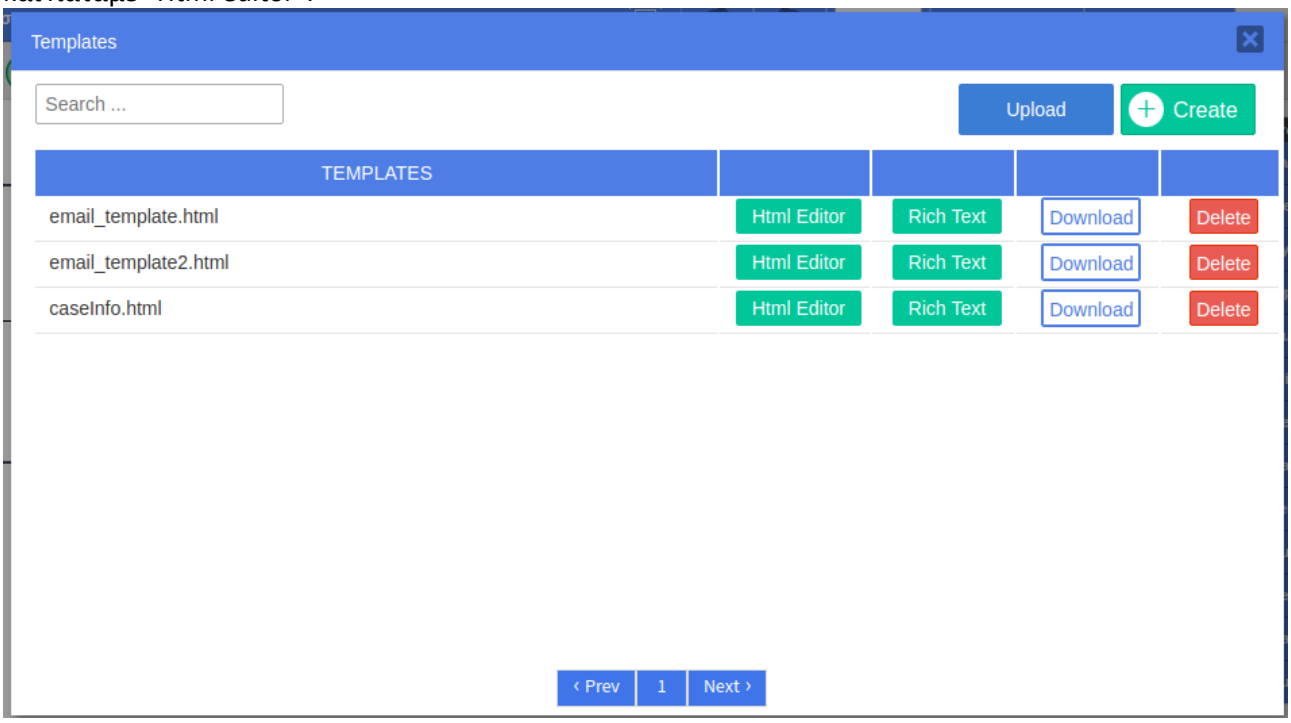
1. Από την καρτέλα designer επιλέγουμε την διαδικασία κατά την οποία αποστέλλεται το mail που επιθυμούμε να αλλάξουμε. Οι διαδικασίες κατά τις οποίες αποστέλλεται mail είναι οι "reopen case", "διαδικασία εισαγωγής αναλώσιμων από χρήστες" και "επικοινωνία με τον admin".



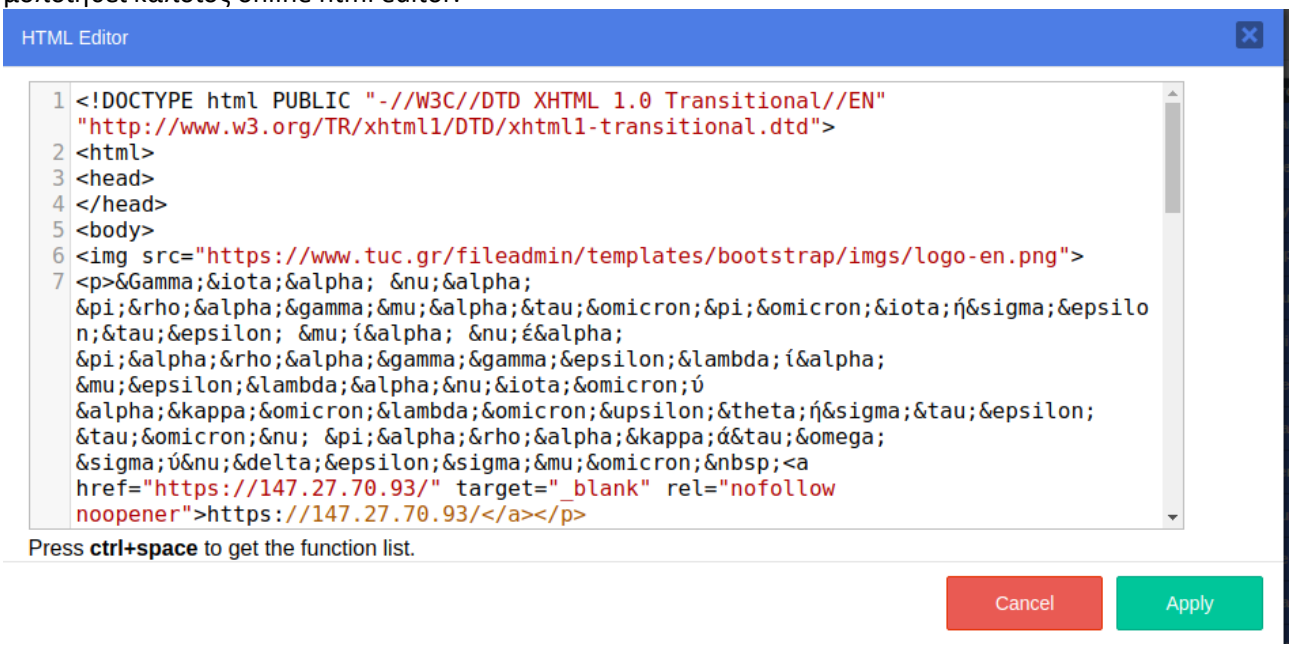
2. Στο παράθυρο που μας εμφανίζεται, στο μενού που υπάρχει στα δεξιά επιλέγουμε "templates"



3. Πατώντας templates εμφανίζονται όλα τα mail σε μορφή html. Επιλέγουμε αυτό που θέλουμε και πατάμε "Html editor".



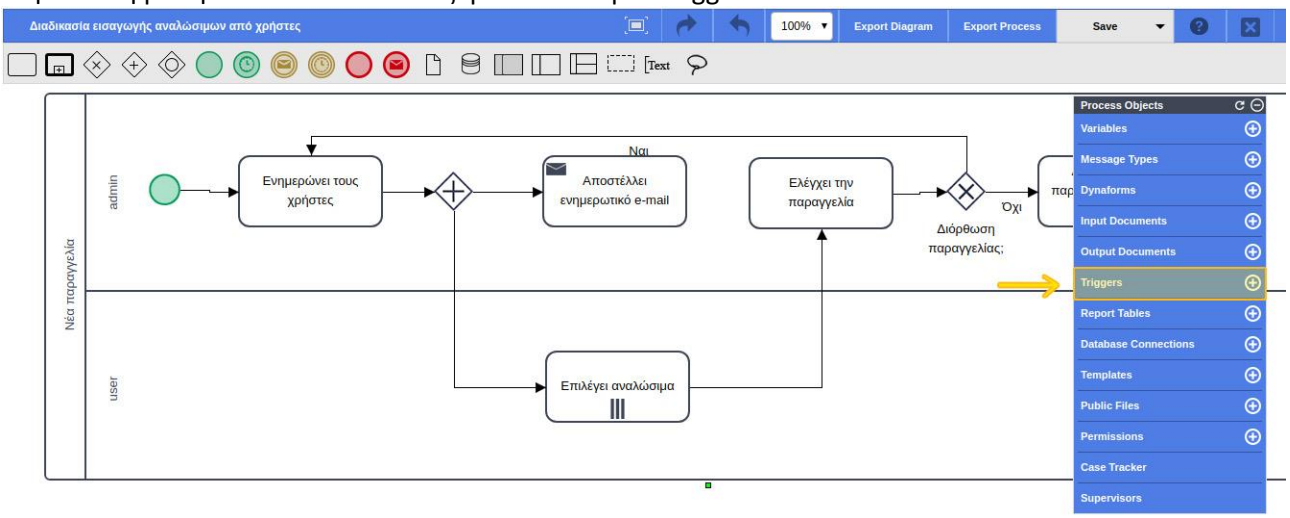
4. Εκεί υπάρχει το κείμενο του mail σε μορφή html. Για την αλλαγή του κειμένου μπορεί να χρησιμοποιηθεί κάποιος online html editor.



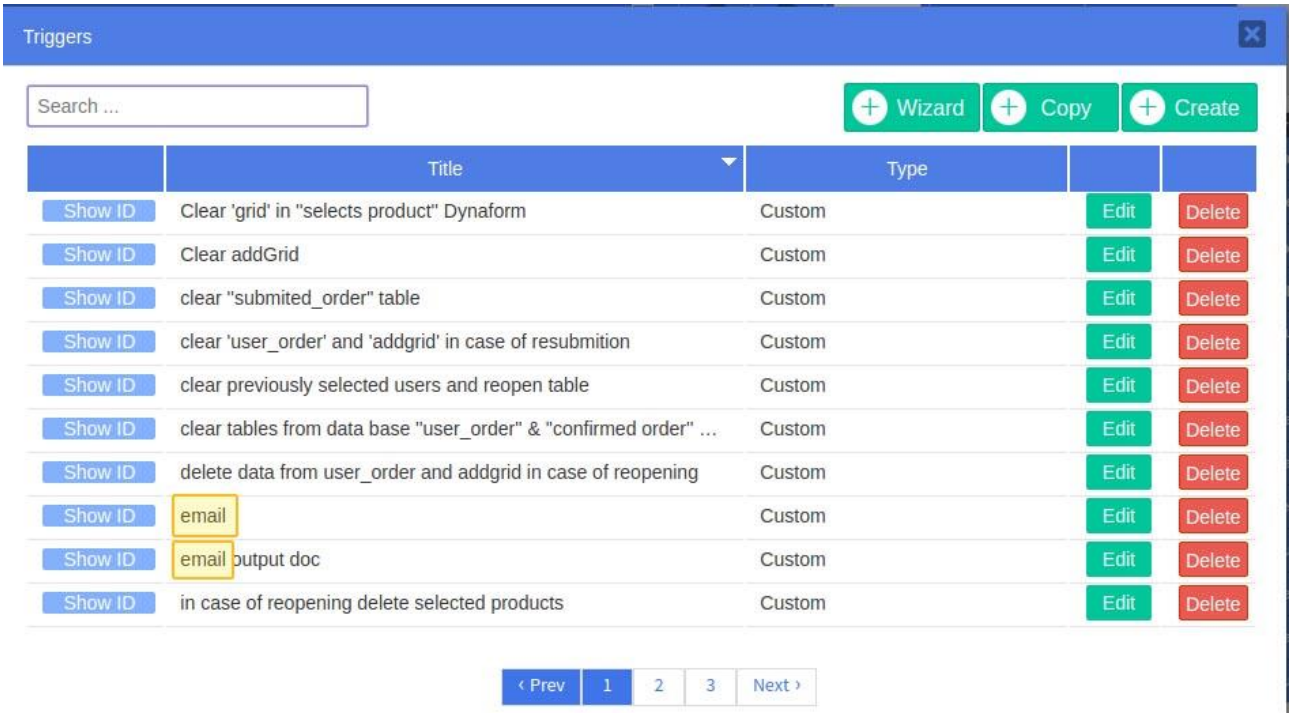
10.1.6.9.2 Αλλαγή αποστολέα/ θέματος

1. Ομοίως με το κεφ. 8.1. επιλέγουμε designer και την διαδικασία που μας ενδιαφέρει.

2. Στην οθόνη με την διαδικασία στο δεξί μενού πατάμε "Triggers".



3. Πατώντας triggers από τις διαθέσιμες επιλέγουμε κάποιο trigger που έχει στον τίτλο την λέξη email.



4. Στον κώδικα που εμφανίζεται μας νοιάζει η συνάρτηση PMFSendMessage. Εκεί μπορούμε να αλλάξουμε το θέμα.

```
if(@@informUsersType == 0){
   PMFSendMessage(@@APPLICATION, $from, $to, '', '',
      "Έναρξη προμηθειών μελανοδοχείων", 'email_template.html');
}elseif(@@informUsersType == 1){
   PMFSendMessage(@@APPLICATION, $from, $to, '', '',
      "Αλλαγή παραγγελίας μελανιών", 'email_template2.html');
}
```

10.2 TABLES USED IN INK BMANAGER

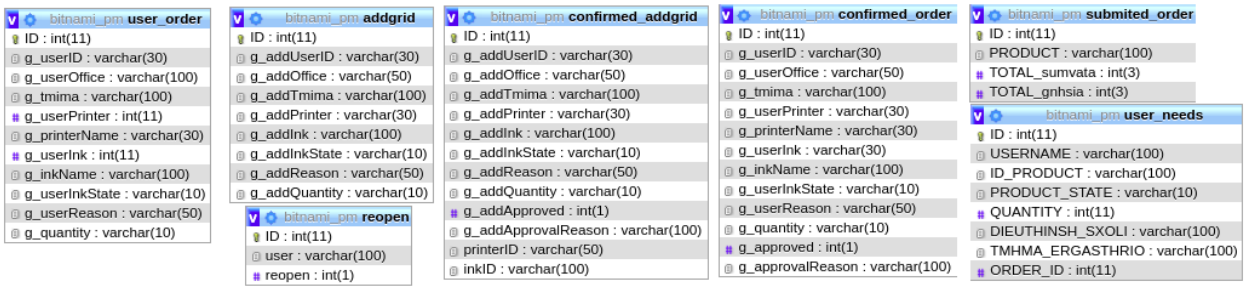


Figure 1 Tables used to save the products throughout the ordering phase

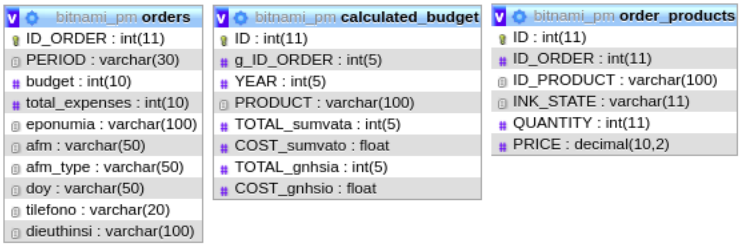


Figure 2 Tables used to permanently save the placed orders and their prices

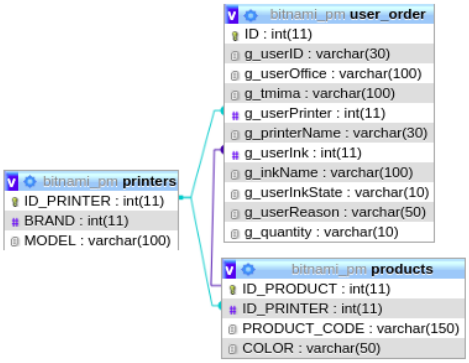


Figure 3 User_order table and its relations. This is the main table where the order products are saved.

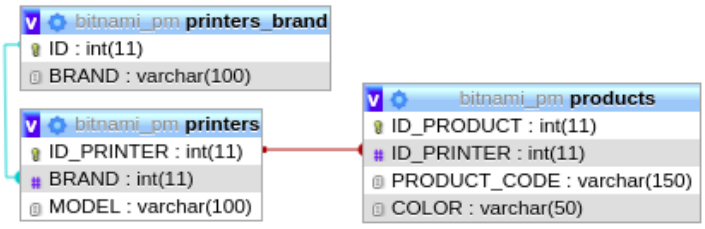


Figure 8 Figure 1 Relations between the tables that include the brands of the printers, the printers and their products

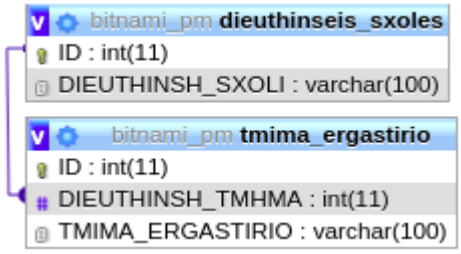


Figure 7Relations between the tables that include the departments of TUC and the corresponding labs or sections

10.3 FORM USED IN THE SATISFACTION SURVEY

Έρευνα ικανοποίησης από την χρήση του Ink BManager για παραγγελίες μελανοδοχείων

* Required

Δημογραφικά στοιχεία

1. **Φύλο** *
   Mark only one oval.
   - ◯ Άνδρας
   - ◯ Γυναίκα

2. **Ηλικία** *
   Mark only one oval.
   - ◯ 18-25
   - ◯ 26-35
   - ◯ 36-45
   - ◯ 45 και άνω

3. **Είστε** *
   Mark only one oval.
   - ◯ Διοικητικός Υπάλληλος
   - ◯ Μέλος ΕΔΙΠ
   - ◯ Καθηγήτρια/ Καθηγητής
   - ◯ Φοιτητής

4. **Πόση ώρα χρειαζόσασταν κατά μέσο όρο για να πραγματοποιήσετε μία παραγγελία χωρίς το Ink BManager.** *
   Mark only one oval.
   - ◯ έως 5 λεπτά
   - ◯ 6-10 λεπτά
   - ◯ 11-15 λεπτά
   - ◯ Παραπάνω από 15 λεπτά
   - ◯ Δεν έχω πραγματοποιήσει παραγγελία χωρίς το Ink BManager

5. **Πόση ώρα χρειαστήκατε για να πραγματοποιήσετε μία παραγγελία με το Ink BManager.** *
Mark only one oval.

- ◯ έως 5 λεπτά
- ◯ 6-10 λεπτά
- ◯ 11-15 λεπτά
- ◯ Παραπάνω από 15 λεπτά

6. **Θεωρείτε ότι το Ink BManager βελτίωσε την διαδικασία παραγγελίας μελανοδοχείων** *
Mark only one oval.

- ◯ Ναι
- ◯ ΄Οχι
- ◯ Ούτε βελτίωσε ούτε δυσκόλεψε την διαδικασία

7. **Ποιον/ Ποιους αφορούν τα μελάνια;**
Check all that apply.

- ☐ Προσωπικό Εκτυπωτή
- ☐ Ένα Εργαστήριο
- ☐ Ένα τομέα σχολής
- ☐ Μια Σχολή
- ☐ Μια Διεύθυνση

8. **Πόσα μελάνια παραγγείλατε;**
Mark only one oval.

- ◯ 1-3 τμχ
- ◯ 4-6 τμχ
- ◯ 7-9 τμχ
- ◯ ≥ 10 τμχ

Ερωτήσεις ικανοποίησης, πόσο ικανοποιημένος είστε σε σχέση με τα παρακάτω:

9. **Φιλικότητα προς τον χρήστη (menu, μηνύματα, επιλογές).** *
Mark only one oval.

	1	2	3	4	5	
Δυσαρεστημένος	◯	◯	◯	◯	◯	Απόλυτα ικανοποιημένος

10. **Ευκολία πλοήγησης μέσα στην πλατφόρμα.** *
Mark only one oval.

	1	2	3	4	5	
Δυσαρεστημένος	◯	◯	◯	◯	◯	Απόλυτα ικανοποιημένος

11. **Γραφικό περιβάλλον της πλατφόρμας.** *
Mark only one oval.

	1	2	3	4	5	
Δυσαρεστημένος	◯	◯	◯	◯	◯	Απόλυτα ικανοποιημένος

12. **Ποικιλία εκτυπωτών/ προϊόντων.** *
Mark only one oval.

	1	2	3	4	5	
Δυσαρεστημένος	◯	◯	◯	◯	◯	Απόλυτα ικανοποιημένος

13. **Συνολική ευκολία στην συμπλήρωση της φόρμας παραγγελίας.** *
Mark only one oval.

	1	2	3	4	5	
Δυσαρεστημένος	◯	◯	◯	◯	◯	Απόλυτα ικανοποιημένος

14. **Ευκολία στην επεξεργασία μίας παραγγελίας (πχ. αλλαγή ποσότητας, προσθήκη νέου προϊόντος). Συμπληρώστε μόνο εάν χρειάστηκε να επεξεργαστείτε μια παραγγελία.**
Mark only one oval.

	1	2	3	4	5	
Δυσαρεστημένος	◯	◯	◯	◯	◯	Απόλυτα ικανοποιημένος

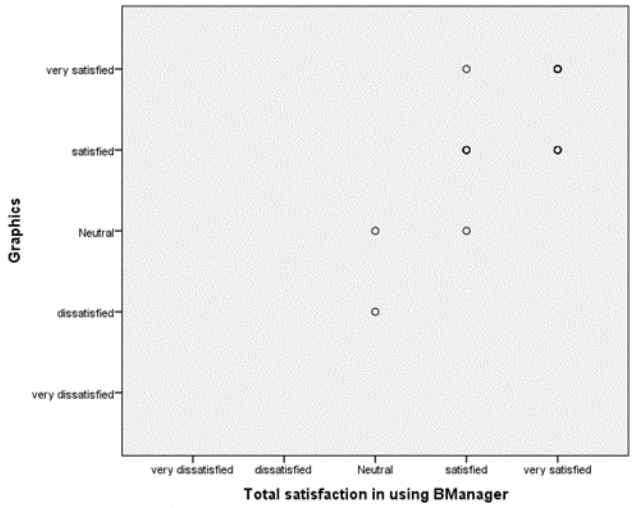
15. **Πόσο ικανοποιημένος είστε συνολικά από την χρήση του Ink BManager για την παραγγελία μελανοδοχείων.** *
Mark only one oval.

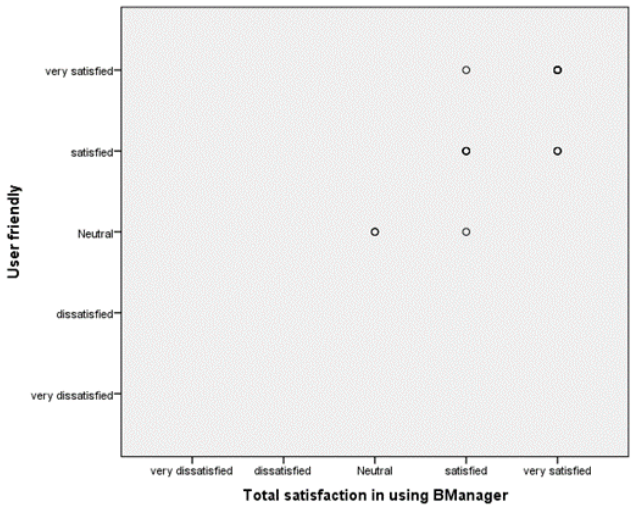
	1	2	3	4	5	
Δυσαρεστημένος	◯	◯	◯	◯	◯	Απόλυτα ικανοποιημένος

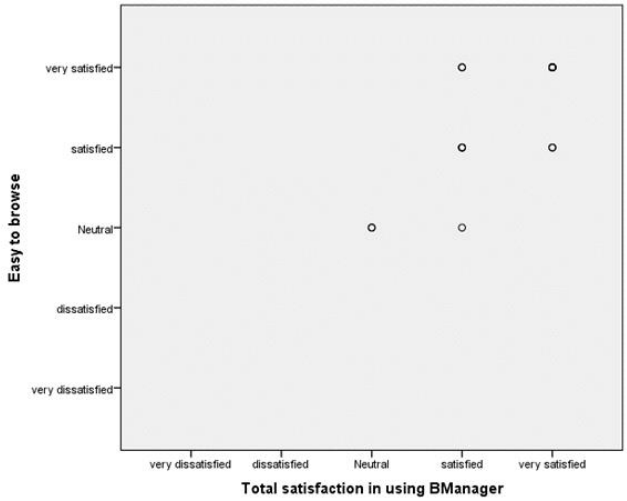
16. **Θα θέλατε να μας προτείνετε κάτι για την βελτίωση της διαδικασίας;** *

10.1 SPSS

Before conducting a Spearman correlation test, there should be an indication that the used datasets have a monotonic relation. This relation can be depicted in the following plots.





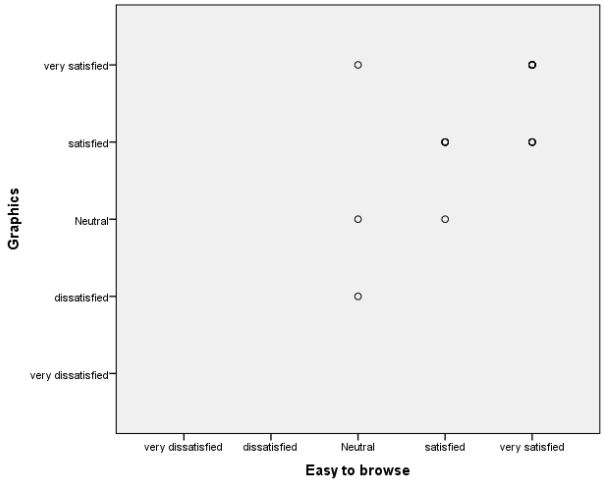


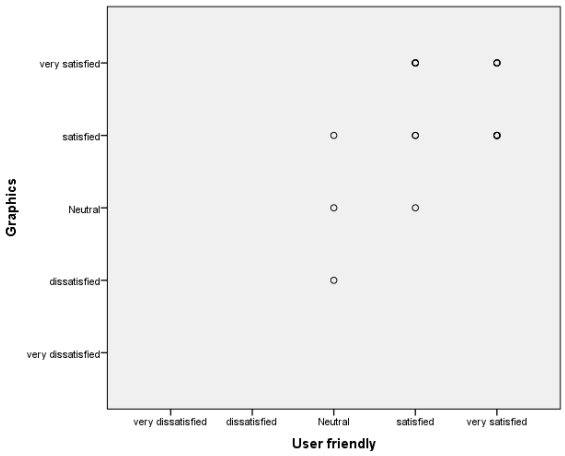
Correlations

			Total satisfaction in using BManager	User friendly	Easy to browse	Graphics
Spearman's rho	Total satisfaction in using BManager	Correlation Coefficient	1.000	.711**	.679**	.665**
		Sig. (2-tailed)	.	.001	.001	.002
		N	19	19	19	19
	User friendly	Correlation Coefficient	.711**	1.000	.561*	.354
		Sig. (2-tailed)	.001	.	.013	.137
		N	19	19	19	19
	Easy to browse	Correlation Coefficient	.679**	.561*	1.000	.583**
		Sig. (2-tailed)	.001	.013	.	.009
		N	19	19	19	19
	Graphics	Correlation Coefficient	.665**	.354	.583**	1.000
		Sig. (2-tailed)	.002	.137	.009	.
		N	19	19	19	19

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).





Correlations

			Graphics	Total satisfaction in using BManager	User friendly	Easy to browse
Spearman's rho	Graphics	Correlation Coefficient	1.000	.665**	.354	.583**
		Sig. (2-tailed)	.	.002	.137	.009
		N	19	19	19	19
	Total satisfaction in using BManager	Correlation Coefficient	.665**	1.000	.711**	.679**
		Sig. (2-tailed)	.002	.	.001	.001
		N	19	19	19	19
	User friendly	Correlation Coefficient	.354	.711**	1.000	.561*
		Sig. (2-tailed)	.137	.001	.	.013
		N	19	19	19	19
	Easy to browse	Correlation Coefficient	.583**	.679**	.561*	1.000
		Sig. (2-tailed)	.009	.001	.013	.
		N	19	19	19	19

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).