Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**ΣΥΣΤΗΜΑ ERP ΕΦΟΔΙΑΣΤΙΚΗΣ ΑΛΥΣΙΔΑΣ ΣΕ ANDROID** **ENTERPRISE ANDROID ERP APPLICATION IN LOGISTICS**
Ονοματεπώνυμο Φοιτητή	**ΚΑΒΟΥΚΛΗΣ ΜΙΧΑΗΛ**
Πατρώνυμο	**ΣΠΥΡΙΔΩΝ**
Αριθμός Μητρώου	**ΜΠΣΠ/ 14029**
Επιβλέπων	**ΔΟΥΛΗΓΕΡΗΣ ΧΡΗΣΤΟΣ,ΚΑΘΗΓΗΤΗΣ**

Ημερομηνία Παράδοσης **10/2018**

Τριμελής Εξεταστική Επιτροπή

ΔΟΥΛΗΓΕΡΗΣ ΧΡΗΣΤΟΣ	ΚΟΤΖΑΝΙΚΟΛΑΟΥ ΠΑΝΑΓΙΩΤΗΣ	ΚΟΠΑΝΑΚΗ ΕΥΑΓΓΕΛΙΑ
ΚΑΘΗΓΗΤΗΣ	ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ	ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΡΙΑ

ΠΕΡΙΛΗΨΗ

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος ERP (Enterprise Resource Planning) για κινητές συσκευές Android με στόχο την κάλυψη των βασικών αναγκών μικρών ή μικρομεσαίων επιχειρήσεων του κλάδου της εφοδιαστικής αλυσίδας. Οι κατηγορίες αναγκών των χρηστών που καλύπτονται από το σύστημα είναι η διαχείριση αποθεμάτων, η διαχείριση αποθήκης, η διαχείριση παραγγελιών και η επικοινωνία με άλλους χρήστες του συστήματος. Μέσω του συστήματος, που αναπτύχθηκε, ο χρήστης μπορεί (ανάλογα με τον ρόλο χρήστη που έχει και τα αντίστοιχα δικαιώματα) να καταχωρήσει στο σύστημα αποθήκης, κατηγορίες προϊόντων, προϊόντα και στην συνέχεια να προχωρήσει σε παραγγελίες καταχωρημένων προϊόντων και να τις διαχειριστεί πλήρως (ακύρωση, έλεγχος κατάστασης και ιστορικού). Επιπλέον, μπορεί να ελέγξει το υπάρχον αποθεματικό διαθέσιμο του συνόλου των προϊόντων ή συγκεκριμένου προϊόντος, να ελέγξει τα προϊόντα τα οποία βρίσκονται κάτω από το όριο ασφαλείας αποθεματικού με παράλληλη δυνατότητα παραγγελίας της διαφοράς έως το όριο ασφαλείας. Τέλος, υπάρχει η δυνατότητα επικοινωνίας με τους άλλους χρήστες του συστήματος μέσω φόρμας ανταλλαγής μηνυμάτων.

Η παρούσα εργασία αποτελεί μια ενναλακτική πρόταση στα πληροφοριακά συστήματα ERP που χρησιμοποιεί ο κλάδος της εφοδιαστικής αλυσίδας και έχει ως στόχο να καλύψει τις βασικές ανάγκες μιας μικρής επιχείρησης. Επίσης, αποτελεί μία βάση για την περαιτέρω ανάπτυξη του συστήματος, ώστε να καλύπτει πολύ μεγαλύτερο εύρος αναγκών του χώρου.

ABSTRACT

The scope of the present thesis is the development of an ERP (Enterprise Resource Planning) information system for Android mobile devices designed to satisfy the basic needs of small and medium enterprises of the logistics sector. The categories of the user's needs that are covered by the system are Inventory Management, Warehouse Management, Orders Management and communication with the other users of the system. Via the system developed, the user can (depending on his/her user role and its corresponding privileges) register in the system warehouses, item categories, specific products and then proceed with ordering registered products. Following the order, the user can fully manage the placed orders (e.g. cancel an order, view the order history, view the order status etc.). Furthermore, the user can access the present stock of all registered items, can search for the availability of a specific item and can check for the products that are below the safety stock limit, with the option to automatically order the necessary quantity to meet the limit. Finally, there is a messaging system among the users of the system via a message form, available to each user.

The present thesis constitutes an alternative option in ERP information systems of the logistics sector and its purpose is to satisfy the basic needs of the specific sector. Moreover, it will form the basis for further development that will cover a much wider range of needs that the users have.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ

1.1 Αντικείμενο Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία έχει ως στόχο την ανάπτυξη ενός mobile ERP συστήματος για την εξυπηρέτηση των αναγκών μικρών ή μικρομεσαίων επιχειρήσεων που δραστηριοποιούνται στον χώρο των logistics.Αφορά τις επιχειρήσεις εκείνες, των οποίων το επίπεδο μηχανογράφησης δεν είναι ιδιαίτερα προηγμένο και οι εργαζόμενοι τους, σε πολλές περιπτώσεις, φέρουν πολλαπλούς ρόλους και συμμετέχουν σε πολλές διαδικασίες της εταιρείας. Σκοπός της εργασίας είναι να καλύψει τις βασικές ανάγκες των εργαζόμενων του χώρου (κατηγορίες Warehouse Management, Inventory Management,Orders & Deliveries), καθώς επίσης και να ενσωματώσει επιπλέον στοιχεία που θα διευκολύνουν την εργασία τους (επικοινωνία χρηστών σε πραγματικό χρόνο με μηνύματα και ανάθεση εργασιών).Τα παραπάνω θα ενσωματωθούν στην όχι ευρέως υιοθετημένη τεχνολογία των mobile ERP συστημάτων.

1.2 Logistics και ERP

Ο όρος Logistics συνοπτικά περιγράφει την διαδικασία διαχείρισης της εφοδιαστικής αλυσίδας μίας εταιρείας ή ενός οργανισμού και ενσωματώνει τις λειτουργίες του σχεδιασμού και της υλοποίησης της αποδοτικότερης αποθήκευσης των προϊόντων και της μεταφοράς τους, καθώς επίσης και την διαχείριση της πληροφορίας που σχετίζεται με τις συγκεκριμένες διαδικασίες. Οι ανάγκες που προκύπτουν από αυτές τις επιχειρησιακές διαδικασίες και που μπορούν να καλυφθούν από ένα πληροφοριακό σύστημα είναι μεταξύ άλλων η ταχύτητα της πρόσβασης στην πληροφορία (π.χ. έλεγχος αποθέματος προϊόντος) και η ακρίβεια της πληροφορίας αυτής, η διαθεσιμότητα της πληροφορίας σε πραγματικό χρόνο, η υποστήριξη λήψης αποφάσεων που αφορούν την βέλτιστη επιλογή αποθήκευσης ενός προϊόντος (π.χ. πρόταση του πληροφοριακού συστήματος για το που πρέπει να αποθηκευτεί ένα νέο προϊόν που εισάγεται σε μία αποθήκη) και η διευκόλυνση της διαδικασίας παραγγελίας νέων προϊόντων και ενημέρωση του αποθεματικού μετά από παραλαβή.

Τις ανάγκες αυτές θα προσπαθήσουμε να καλύψουμε μέσα από το σύστημα που θα αναπτυχθεί ως μέρος αυτής της διπλωματικής εργασίας.

1.3 Οργάνωση κειμένου

Η παρούσα εργασία έχει εκπονηθεί με την ακόλουθη δομή κειμένου :

Στο κεφάλαιο 2 θα γίνει μία βιβλιογραφική επισκόπηση του πεδίου των ERP συστημάτων από

την σκοπιά της παρακολούθησης της εξέλιξης τους, των επιπτώσεων που έφερε η χρήση τους στους κλάδους όπου υιοθετήθηκαν, των δυνατοτήτων που προσφέρουν και της αναμενόμενης μελλοντικής τους εξέλιξης. Επιπλέον, θα δούμε τα ιδιαίτερα χαρακτηριστικά της υποκατηγορίας με την οποία καταπιάνεται η εργασία (mobile ERP συστήματα).

Στο κεφάλαιο 3 θα γίνει η παρουσίαση της ανάλυσης και του σχεδιασμού του συστήματος. Ζητήματα, όπως, η ανάλυση των απαιτήσεων της εφαρμογής και των αναγκών που πρέπει να καλύπτει ο σχεδιασμός της υλοποίησης, καθώς και η επιλογή των τεχνολογιών υλοποίησης θα καλυφθούν στο κεφάλαιο αυτό.

Στο κεφάλαιο 4 θα γίνει μία αναλυτική περιγραφή του συστήματος. Θα παρουσιαστούν όλες οι δυνατότητες που προσφέρει, θα γίνει ο έλεγχός του, με δεδομένα σενάρια ελέγχου και γενικότερα θα καταγραφούν τα απαιτούμενα ενός εγχειριδίου χρήσης του.

Στο κεφάλαιο 5 θα γίνει αναφορά σε διάφορα τεχνικά ζητήματα που αφορούν την υλοποίηση του συστήματος. Θα δούμε τα βασικότερα τμήματα του κώδικα που δημιούργησε την εφαρμογή και θα εξετάσουμε ζητήματα που αφορούν την ασφάλεια της εφαρμογής και την απόδοσή της.

Στο κεφάλαιο 6 θα συνοψίσουμε την εμπειρία της εργασίας και θα προχωρήσουμε στα συμπεράσματα στα οποία καταλήξαμε με την ολοκλήρωση της εστιάζοντας στην συνεισφορά της εργασίας στον τομέα των ERP συστημάτων για τον κλάδο των logistics και στις προτάσεις μας για μελλοντική έρευνα στο αντικείμενο.

Τέλος, θα ακολουθήσει μία καταγραφή της βιβλιογραφίας που χρησιμοποιήθηκε.

ΚΕΦΑΛΑΙΟ 2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

2.1 Ορισμός ERP συστημάτων

Τα ERP (Enterprise Resource Planning) συστήματα αποτελούν εφαρμογές λογισμικού χρησιμοποιούμενες από εταιρείες ή οργανισμούς για να ενσωματώσουν και να οργανώσουν πληροφορίες από κάθε επιχειρησιακή διεργασία *(Ellen F. Monk & Bret J. Wagner,2013)* ενώ παράλληλα βοηθούν τον οργανισμό/εταιρεία να διαχειριστεί τις διεργασίες αυτές χρησιμοποιώντας μια κοινή βάση δεδομένων και κοινόχρηστα εργαλεία ελέγχου. Ο σκοπός τους είναι να διευκολύνουν τη ροή των πληροφοριών μεταξύ όλων των επιχειρησιακών λειτουργιών μέσα στα όρια της οργάνωσης και να καταφέρουν τις συνδέσεις προς τα έξω με τα ενδιαφερόμενα μέρη. Η λέξη κλειδί πίσω από την έννοια ERP είναι η ενοποίηση *(Nishad Nawaz,2013).*

2.2 Μοντέλα λειτουργίας ERP συστημάτων

Εξετάζοντας κανείς τους τρόπους διάθεσης ενός ERP συστήματος (erp delivery models) εν έτη 2018 θα καταλήξει σε 5 διαφορετικές υλοποιήσεις οι οποίες διαφοροποιούνται ως προς το περιβάλλον εγκατάστασης του λογισμικού, τον τρόπο πρόσβασης σε αυτό αλλά και τον διαχωρισμό των ευθυνών συντήρησης του συστήματος μεταξύ του παρόχου/κατασκευαστή του συστήματος και του πελάτη.

Η πρώτη κατηγορία διάθεσης και η πιο παραδοσιακή είναι η εγκατάσταση *on-premises*.Σε αυτήν την μορφή διάθεσης ο πελάτης που αγοράζει το λογισμικό το έχει εγκατεστημένο σε δικούς του servers οι οποίοι βρίσκονται στο χώρο του. Συνοπτικά, ο πελάτης αγοράζει το λογισμικό από τον κατασκευαστή του και έναν αριθμό αδειών και στην συνέχεια το προσωπικό του τμήματος IT του πελάτη θα εγκαταστήσει το λογισμικό σε υπάρχοντες servers ή σε νέους που θα αγοραστούν με υποδείξεις προδιαγραφών από τον πάροχο του λογισμικού και θεωρητικά εφόσον ο πελάτης δεν παραβιάσει του όρους άδειας χρήσης μπορεί να συνεχίσει να χρησιμοποιεί το λογισμικό για πάντα *(Jonathan Gross,2012)*.Στην συγκεκριμένη κατηγορία διάθεσης ο πελάτης επωμίζεται το βάρος της συντήρησης των servers που φιλοξενούν τα τμήματα της εφαρμογής (database server,application server) και παράλληλα φέρει την ευθύνη της διαθεσιμότητας του συστήματος.

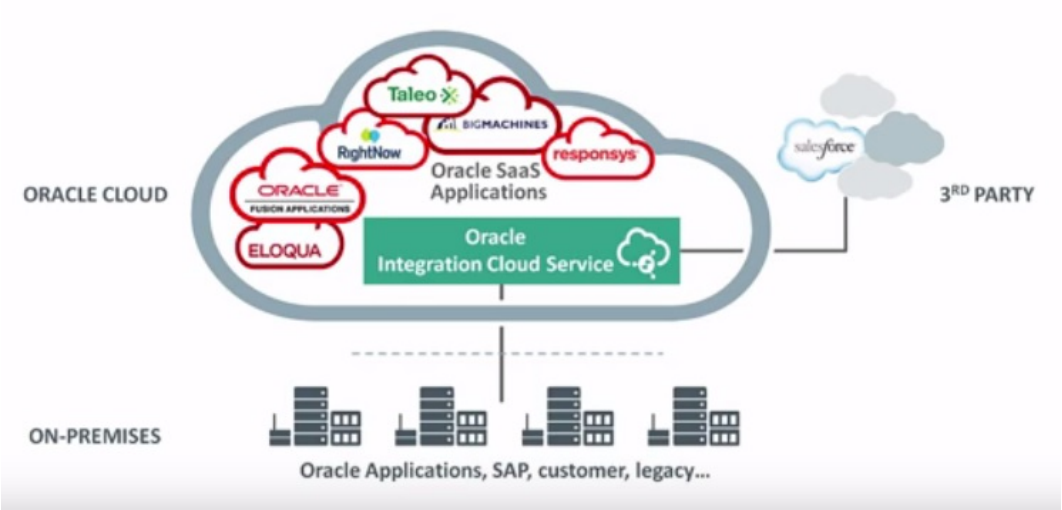
Η δεύτερη κατηγορία είναι *η φιλοξενούμενη (hosted)* εγκατάσταση. Εδώ ο πελάτης κατέχει (μετά από αγορά) το λογισμικό αλλά όχι τον server της εγκατάστασης τον οποίο και ενοικιάζει από μια εταιρεία πάροχο cloud υπηρεσιών τύπου Infrastucture as a Service (Iaas) *(Booz & Company, 2013)*.Σε αυτού του τύπου τις εγκαταστάσεις η πρόσβαση στο λογισμικό πραγματοποιείται μέσα από εικονικά ιδιωτικά δίκτυα (Virtual Private Network) αντί μέσω του

διαδικτύου για λόγους ασφάλειας επικοινωνίας μεταξύ client και server.

Η τρίτη κατηγορία είναι η εγκατάσταση του λογισμικού σε public cloud server του κατασκευαστή και η ενοικίαση από την πλευρά του πελάτη της άδειας χρήσης του. Η κατηγορία αυτή καλείται επίσης και Saas (Software as a Service).Το μοντέλο Saas ορίζεται ως "μία εφαρμογή ή μία υπηρεσία που έχει αναπτυχθεί σε ένα κεντρικό data center εντός ενός δικτύου και στην οποία παρέχεται πρόσβαση προς χρήση έναντι στη βάση μίας επαναλαμβανόμενης αμοιβής όπου οι χρήστες συνήθως ενοικιάζουν την εφαρμογή/υπηρεσία από έναν κεντρικό προμηθευτή" *(Hoch et al,2013).*

Η τέταρτη κατηγορία είναι η εγκατάσταση του λογισμικού σε private cloud server.Στην κατηγορία αυτή ο πάροχος λειτουργεί για κάθε πελάτη ένα αποκλειστικό ξεχωριστό cloud περιβάλλον. Ο κάθε πελάτης κατέχει το hardware και τις άδειες χρήσης και έχει το δικαίωμα να διατηρήσει τον απόλυτο έλεγχο της διαχείρισης της εφαρμογής και των servers ή να τον παραχωρήσει εξ ολοκλήρου στο πάροχο του private cloud *(Oracle Accelerate for midsize companies,2011).*

Στην πέμπτη κατηγορία βρίσκουμε τις υβριδικές εγκαταστάσεις (μίξη public και private cloud ή και on premise εγκατάστασης).Τέτοιες εγκαταστάσεις αφορούν κυρίως την διατήρηση στις εγκαταστάσεις του πελάτη legacy εφαρμογών που δεν υποστηρίζονται πλέον από τον κατασκευαστή και την παροχή μέσω cloud των σύγχρονων εφαρμογών του παρόχου *(Oracle Accelerate for midsize companies,2011).*



Εικόνα 2.1 Oracle Integration Cloud Service - Πηγή : https://blogs.oracle.com/

2.3 Πλεονεκτήματα χρήσης ERP συστημάτων

Σήμερα η αγορά ενός ERP συστήματος από μία εταιρεία δεν αποτελεί πολυτέλεια αλλά αναγκαιότητα. Το να έχει μία εταιρεία ή ένας οργανισμός ένα ERP σύστημα σωστά εγκατεστημένο και προσωπικό πλήρως εκπαιδευμένο να το χρησιμοποιεί με τον καλύτερο δυνατό τρόπο είναι απαραίτητο για την επιβίωση της εταιρείας σε έναν σκληρό και ανταγωνιστικό κόσμο (*Alexis Leon,2008*).

Εξετάζοντας την βιβλιογραφία μπορούμε να κατανοήσουμε τους λόγους της τεράστιας επιτυχίας των συγκεκριμένων πληροφοριακών συστημάτων καταγράφοντας κάποια από τα οφέλη που έχουν οι εταιρείες που υιοθετούν ένα τέτοιο σύστημα :

- *Τα ERP ενσωματώνουν σε ένα σύστημα όλες τις διεργασίες μιας εταιρείας.*

  Οι διεργασίες διαφορετικών τμημάτων μιας εταιρείας συχνά διασταυρώνονται και διαφορετικά δεδομένα τα οποία υπήρχαν διασκορπισμένα σε ετερογενή συστήματα με την χρήση των Enterprise Resource Planning συστημάτων ενοποιούνται σε μία πλατφόρμα (*Daniel O'Leary,2000*).

- *Τα ERP ενσωματώνουν "best practices" διαδικασίες*

  Τα ERP συστήματα έχουν ενσωματωμένες χιλιάδες best practice επιχειρησιακές διεργασίες οι οποίες μπορούν να χρησιμοποιηθούν για να βελτιώσουν τον τρόπο με τον οποίο οι εταιρείες κάνουν την δουλειά τους. Η επιλογή και η ένταξη ενός ERP συστήματος στην καθημερινότητα της εταιρείας προϋποθέτει την υιοθέτηση τέτοιου τύπου διαδικασιών (*Daniel O'Leary,2000*).

- *Μεγάλη βελτίωση στην ακρίβεια των αποθηκευμένων πληροφοριών σχετικά με το ιστορικό εταιρικών διαδικασιών (Alexis Leon,2008*).

- *Ταχύτερη επικοινωνία και ροή πληροφορίας στα πλαίσια της επιχείρησης*

Τα ERP συστήματα διευκολύνουν την επικοινωνία μεταξύ των χρηστών και την μεταβίβαση κρίσιμων πληροφοριών που αφορούν το σύνολο της επιχείρησης. Οι εργαζόμενοι έχουν πρόσβαση σε δεδομένα πραγματικού χρόνου, έγγραφα και αναφορές που αφορούν τα καθήκοντά τους στην εταιρεία. Η πρόσβαση στην ροή της πληροφορίας είναι πλήρως προσαρμόσιμη ανάλογα με τη θέση του κάθε εργαζόμενου στην εταιρεία (*Andrejs Tambovcevs & Tatjana Tambovceva ,2013*).

- *Υποστήριξη αποφάσεων βασισμένων σε έγκυρες λογιστικές αναφορές*

Σε ταχέως αναπτυσσόμενες και ανεπτυγμένες αγορές, οι εταιρείες έρχονται αντιμέτωπες με έναν αυξανόμενο αριθμό λογιστικών διαδικασιών και έχουν εισέλθει σε μία ψηφιοποιημένη λογιστική εποχή. Οι χαμηλής ποιότητας λογιστικές/οικονομικές αναφορές μπορούν να οδηγήσουν σε λανθασμένες αποφάσεις που μπορεί να έχουν αρνητική επίπτωση στην εταιρεία (MIT Open Access Articles,*2009*)

- *Εξάλειψη ασσυμετριών της πληροφορίας στα πλαίσια του οργανισμού*

Τα ERP συστήματα αποθηκεύουν όλη την πληροφορία στην ίδια βάση δεδομένων, εξαλείφοντας

πιθανές ασσυμετρίες της που μπορεί να προκύψουν από την καταγραφή σε διαφορετικά συστήματα ίδιων συναλλαγών από διαφορετικά τμήματα (*Daniel O'Leary,2000*).

2.4 Συγκριτική μελέτη εγκατάστασης on-premise και cloud

Σε προηγούμενη παράγραφο (2.2) είδαμε τις διαφορετικές εκδοχές εγκατάστασης με τις οποίες μπορεί να προσφέρεται ένα ERP σύστημα. Αν αναλύσουμε περισσότερο τις πέντε κατηγορίες (on-premise,hosted,public cloud,private cloud,hybrid) θα δούμε ότι οι βασικές επιλογές που προσφέρονται στην εταιρεία πελάτη είναι η εγκατάσταση στο δικό της εταιρικό περιβάλλον (on-premises) ή σε ένα cloud περιβάλλον κάποιας μορφής (hosted ή SaaS).Θα επιχειρήσουμε να εντοπίσουμε τις βασικές διαφορές και τα βασικά πλεονεκτήματα και μειονεκτήματα των μοντέλων αυτών.

Οι τεχνολογίες cloud είναι ένα προχωρημένο μοντέλο εκχώρησης αρμοδιοτήτων και υπηρεσιών IT σε τρίτους το οποίο επιτρέπει σε έναν οργανισμό να χρησιμοποιήσει πόρους τρίτων σαν υπηρεσίες μέσω του διαδικτύου χωρίς να χρειάζεται να φιλοξενεί του πόρους αυτούς σε φυσική μορφή εσωτερικά του οργανισμού (*University of Sheffield,2013*).Ακριβώς λόγω αυτής της εκχώρησης των αρμοδιοτήτων και της παροχής κάποιων υπηρεσιών εκτός εταιρείας το μεγάλο πλεονέκτημα των cloud υπηρεσιών είναι ότι μία σειρά από κοστοβόρα σε χρήματα, ανθρωποώρες και ενέργεια ζητήματα όπως η συντήρηση των φυσικών μηχανών, η εξασφάλιση διαθεσιμότητας των προγραμμάτων, ενημερώσεις λειτουργικών συστημάτων, η διασφάλιση αντιγράφων ασφαλείας,cyber security ζητήματα κ.α. περνούν στην ευθύνη τρίτων οργανισμών ελευθερώνοντας εταιρικούς πόρους που μπορούν να χρησιμοποιηθούν σε άλλες εταιρικές διαδικασίες.

Ένα επιπλέον σημείο ελέγχου των πλεονεκτημάτων και μειονεκτημάτων κάθε μοντέλου είναι σίγουρα το οικονομικό σκέλος του κάθε σεναρίου υλοποίησης. Στο σενάριο της on-premise εγκατάστασης η εταιρεία πρέπει να αγοράσει τις άδειες χρήσης του software καθώς και τους servers που θα το φιλοξενήσουν όπως επίσης και το λογισμικό που θα χρησιμοποιούν οι servers. Σε σχέση με μία cloud υλοποίηση το κόστος στην on-premise εγκατάσταση είναι πολύ μεγαλύτερο στην αρχική επένδυση λόγω της ανάγκης για αγορά των φυσικών μηχανημάτων όπως επίσης και του παρελκόμενου απαραίτητου εξοπλισμού (μονάδες UPS,συστήματα ψύξης κλπ.). Στην cloud εγκατάσταση ο πελάτης ουσιαστικά ενοικιάζει τμήματα φυσικών μηχανών που φιλοξενούνται σε άλλα datacenters μηδενίζοντας την ανάγκη για αρχική επένδυση σε hardware (*R. Meganathan , P. Jeyanthi ,2016*).

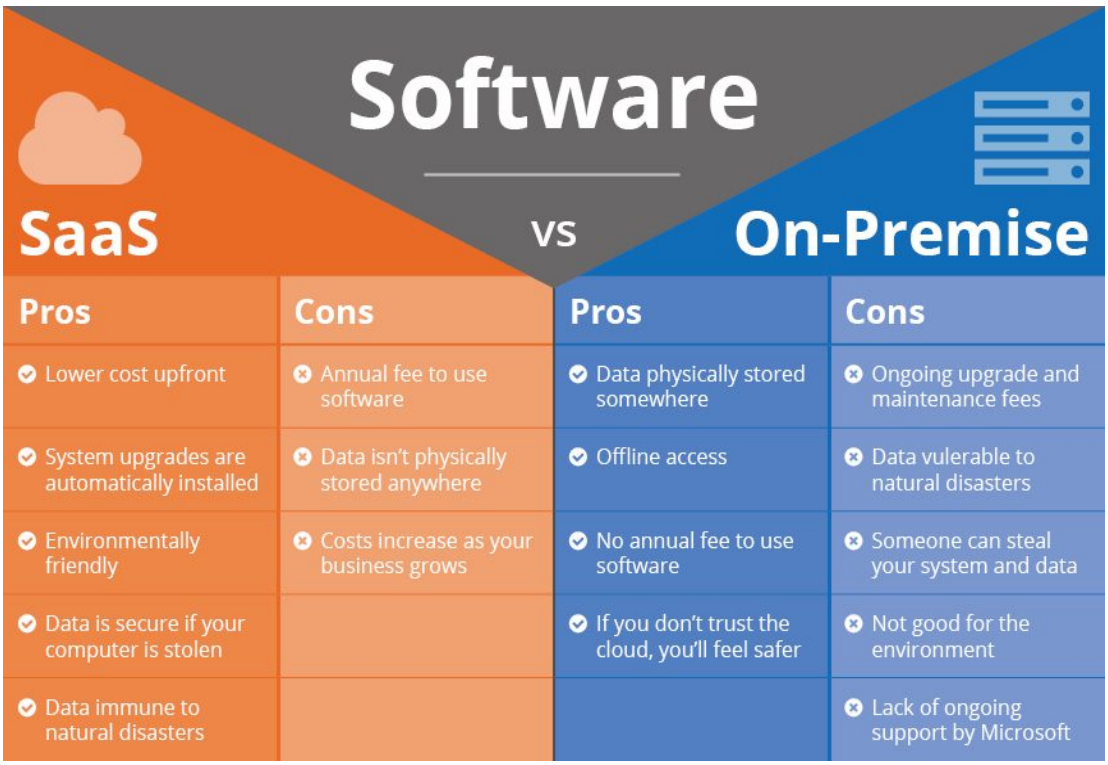
Εκτός της αρχικής επένδυσης υπάρχει και ένα μηνιαίο κόστος για την χρήση του ERP το οποίο αφορά την άδεια χρήσης του (το κόστος αυτό είναι ίδιο και στα δύο σενάρια υλοποίησης) και το μηνιαίο κόστος των μηχανών που το φιλοξενούν. Στην on-premise εγκατάσταση το κόστος αυτό συνίσταται στα έξοδα συντήρησης του εξοπλισμού και στο κόστος της ενέργειας για τη λειτουργία του ενώ στην cloud εγκατάσταση το κόστος αυτό συνοψίζεται στο αντίτιμο που η εταιρεία πληρώνει στον πάροχο των cloud υπηρεσιών. Στην κατηγορία της cloud εγκατάστασης εκτός της επιλογής μίας hosted λύσης σε αρκετές περιπτώσεις (εξαρτάται από τον κατασκευαστή του ERP) υπάρχει η δυνατότητα χρήσης του ERP σαν Software as a Service (SaaS).Στο σενάριο αυτό ο πελάτης απαλλάσσεται και από το κόστος της ενοικίασης των μηχανών που θα φιλοξενήσουν το ERP σύστημα και πληρώνει στον πάροχο (που είναι ο ίδιος ο

κατασκευαστής του ERP) ένα αντίτιμο για την χρήση του προσφερόμενου μέσω διαδικτύου συστήματος και το οποίο είναι ανάλογο με τη χρήση που κάνει (A.Dubey,2007).

Περαιτέρω, ένα πολύ σημαντικό κεφάλαιο που απασχολεί τους IT experts των εταιρειών σε σχέση με την μετάβαση από μία παραδοσιακή on-premise εγκατάσταση σε μία cloud είναι η ασφάλεια των δεδομένων της εταιρείας. Οι ενστάσεις των υπεύθυνων στον χώρο του IT στελεχών έχει να κάνει με την δυσκολία τους να εμπιστευθούν τρίτες εταιρείες με τα δεδομένα τους γενικά και κυρίως με τα ευαίσθητα εταιρικά δεδομένα (*Berkeley University,2009*) .

Η μη φυσική πρόσβαση στα δεδομένα και η εκτός του δικού τους ελέγχου διασφάλιση της διαβαθμισμένης πρόσβασης στα δεδομένα τους είναι οι βασικοί λόγοι για αυτούς τους ενδοιασμούς οι οποίοι ωστόσο συνεχώς και κάμπτονται και όλο και περισσότερες εταιρείες μεταφέρουν τις υποδομές τους σε cloud περιβάλλοντα.

Στην παρακάτω εικόνα μπορούμε να δούμε συνοπτικά τα πλεονεκτήματα και αντίστοιχα τα μειονεκτήματα που προκύπτουν μεταξύ της υλοποίησης on-premise και της Software as a Service λύσης.



Εικόνα 2.2 **Πηγή : www.servicecore.com**

2.5 Συνοπτική ιστορική εξέλιξη των ERP συστημάτων

Τα ERP συστήματα εμφανίζονται στις αρχές της δεκαετίας του 90 σαν διάδοχοι των πρόδρομών τους MRP συστημάτων (Material Requirements Planning) τα οποία χρησιμοποιούνταν στην κατασκευαστική βιομηχανία και σχεδόν αποκλειστικός στόχος της χρήσης τους ήταν η βέλτιστη διαχείριση των αποθεματικών των πρώτων υλών και των μετέπειτα MRP II συστημάτων που είχαν ενσωματώσει κάποιες παραπάνω επιχειρησιακές λειτουργίες στο λογισμικό τους (Teodor Beleţ , Anca Alexandra Purcărea,2017) .Τα πρώτα αυτά συστήματα (MRP) εμφανίζονται τη δεκαετία του 60 και τα παρείχε η εταιρεία IBM και επρόκειτο για μεγάλα, ακριβά συστήματα που απαιτούσαν αρκετό προσωπικό για να συντηρεί τα mainframe computers.Τη δεκαετία του 70 εμφανίζονται μία σειρά από νέα συστήματα της IBM,όπως το COPICS (Communications Oriented Production and Control System) ,το MMAS (Manufacturing Managementand Account System) και το MAPICS (Manufacturing,Accounting and Production Information and Control System) το οποίο θεωρήθηκε ένας πραγματικός πρόδρομος των ERP. Στο τέλος της δεκαετίας εμφανίζεται στην αγορά και το R/2 σύστημα της SAP (σημερινής παγκόσμιας πρωταθλήτριας εταιρείας στον χώρο των ERP συστημάτων).Την δεκαετία του 80 ο όρος MRP II επινοείται για να χαρακτηρίσει τα νεότερα συστήματα με τις επαυξημένες δυνατότητες (International Business Machines Corp (1972)).

Στις αρχές της δεκαετίας του 90 (1992) έχουμε την εμφάνιση στην αγορά του SAP R/3 το οποίο ξεχώρισε από οποιοδήποτε άλλο σύστημα υπήρχε στην αγορά λόγω της υιοθέτησης της αρχιτεκτονικής πελάτη-εξυπηρετητή (client-server) στην hardware υποδομή του.(Teodor Beleţ , Anca Alexandra Purcărea ,2017).Η επανάσταση αυτή υιοθετήθηκε γρήγορα τόσο από μεγάλους οργανισμούς όσο και από μικρές και μεσαίες επιχειρήσεις και έδειξε τον δρόμο για τον σχεδιασμό των ERP συστημάτων που το ακολούθησαν. Θα μπορούσαμε να πούμε ότι το προϊόν αυτό της SAP όρισε την αρχή της εποχής των ERP συστημάτων. Οι σημαντικές εταιρείες που παρέχουν ERP συστήματα την δεκαετία του 90 είναι η SAP,η ORACLE και η JD Edwards and Baan και οι οποίες καλύπτουν με τα συστήματά τους όλες τις βασικές επιχειρησιακές λειτουργίες. Αξίζει να σημειωθεί ότι όρος ERP επινοήθηκε από την GartnerGroup.

Στο τέλος της δεκαετίας του 90 (1998) εμφανίζεται η εταιρεία NetSuite η οποία έχει ως σκοπό την δημιουργία ενός ERP συστήματος το οποίο και θα παραδίδεται μέσω internet στον πελάτη (*www.balloonone.com - history of erp*).

Η δεκαετία του 2000 χαρακτηρίζεται από την είσοδο της τεχνολογίας web στον χώρο του ERP και την ραγδαία άνοδο του internet.Με τις τεχνολογίες διαδικτύου να εξελίσσονται διαρκώς οι χρήστες μπορούν να συνδεθούν απομακρυσμένα σε οποιοδήποτε server στον κόσμο και να αποκτήσουν πρόσβαση σε συστήματα ERP είτε με τεχνολογίες remote application όπως της Citrix είτε με πρόσβαση μέσω web browser (Paul Lindopp,*2015*).

Τέλος, η δεκαετία του 2010 είναι χωρίς αμφιβολία η δεκαετία που χαρακτηρίζεται από την επέκταση των ERP συστημάτων σε cloud υποδομές. Οι συνθήκες είναι πλέον ώριμες λόγω της εξέλιξης της τεχνολογίας του διαδικτύου και των γλωσσών προγραμματισμού. Οι Software as a Service λύσεις είναι πλέον mainstream κομμάτι της αγοράς και πολλές από τις ERP

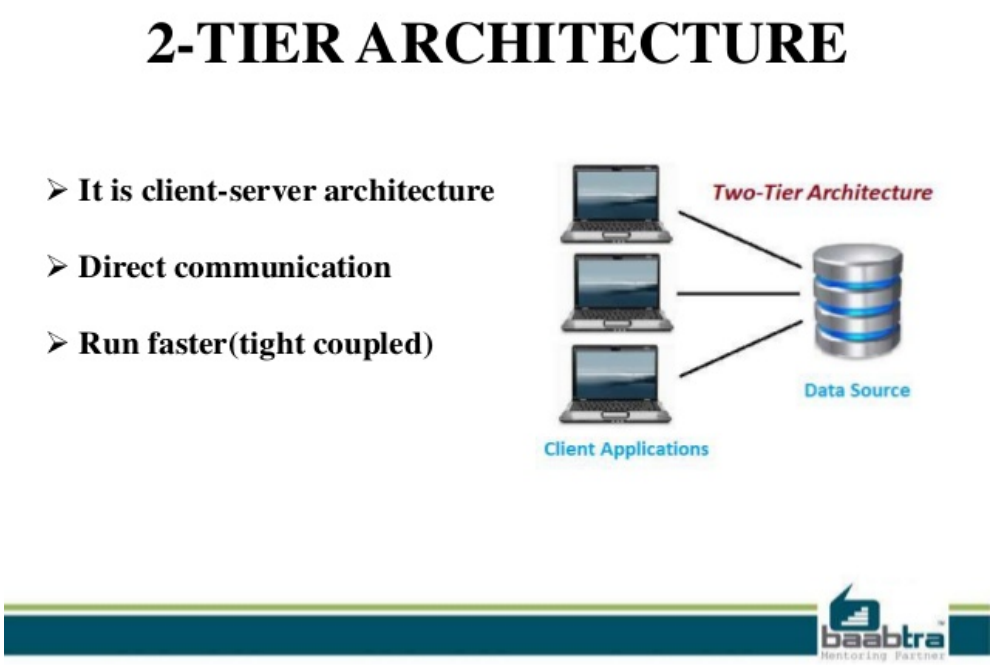
σουίτες γράφονται σε ισχυρές γλώσσες διαδικτύου και συνοδεύονται από εφαρμογές που τρέχουν σε κινητές συσκευές.

2.6 Αρχιτεκτονικές σχεδίασης ERP συστημάτων

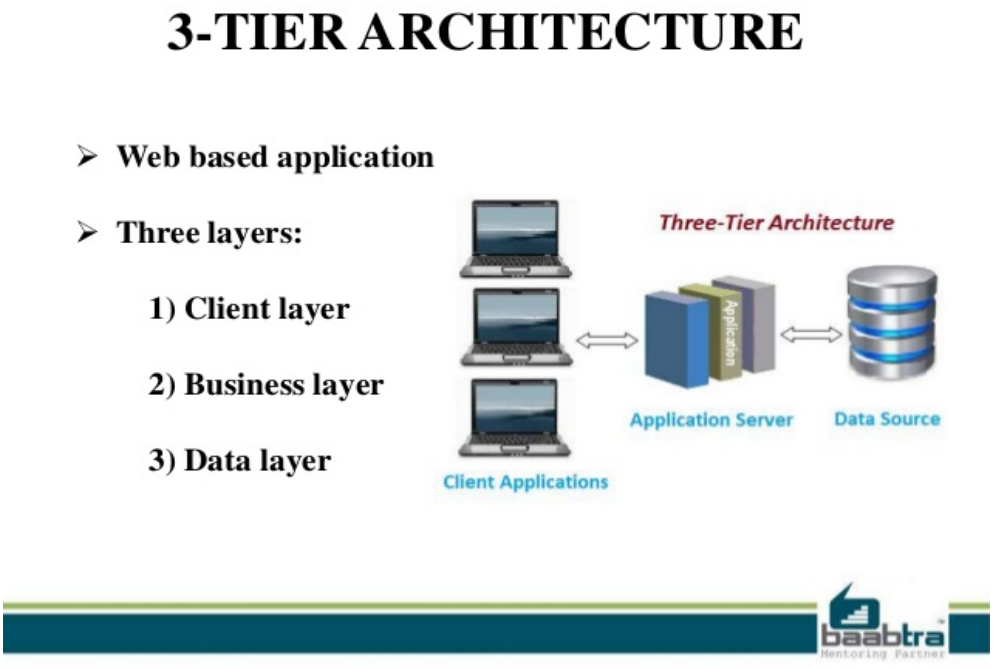
Τα ERP συστήματα είτε στην μορφή των web-based συστημάτων, είτε στην μορφή των local applications,έχουν κοινές αρχιτεκτονικές σχεδίασης που τα χαρακτηρίζει. Οι δύο βασικές αρχιτεκτονικές διαχωρίζονται με βάση τα επίπεδα επικοινωνίας των τμημάτων της εφαρμογής μέχρι τον τελικό χρήστη και είναι η αρχιτεκτονική δύο επιπέδων (2-tier architexture) και η αρχιτεκτονική τριών επιπέδων (3-tier architexture).

Τα πρώτα ERP συστήματα που έφεραν την επανάσταση της χρήσης του μοντέλου client-server ακολουθούσαν το 2-tier μοντέλο σχεδίασης όπου τα δύο επίπεδα ήταν το server-tier και το client-tier.Στο επίπεδο του server-tier υπάρχει η βάση δεδομένων του συστήματος (database server) ενώ στο client-tier είναι εγκατεστημένο το user-interface κομμάτι της εφαρμογής το οποίο επικοινωνεί με την database.

Τα 3-tier συστήματα αποτελούνται από 3 επίπεδα (Presentation,Application και Database) και η βασική διαφοροποίηση με τα 2-tier συστήματα είναι ότι ο χρήστης μέσω του presentation layer δεν έχει απευθείας επικοινωνία με την βάση δεδομένων. Το μεσαίο επίπεδο που μεσολαβεί (middle-tier) είναι υπεύθυνο για την εκτέλεση του business-logic κομματιού και είναι το σημείο όπου τα δεδομένα συλλέγονται από την βάση δεδομένων και μεταφέρονται σε δεύτερο χρόνο στο presentation layer.



Εικόνα 2.3 Απεικόνιση 2 - tier αρχιτεκρονικής



Εικόνα 2.4 Απεικόνιση 3 - tier αρχιτεκρονικής

2.7 Σύνοψη Κεφαλαίου

Συνοψίζοντας το κεφάλαιο της βιβλιογραφικής επισκόπησης είδαμε αρχικά το τι είναι ένα ERP σύστημα μέσα από τον ορισμό του και τα μοντέλα εγκατάστασης με τα οποία διατίθεται. Τα Enterprise Resource Planning συστήματα είναι πληροφοριακά συστήματα που ενοποιούν σε μία εφαρμογή και μία βάση δεδομένων όλες τις επιχειρησιακές διεργασίες ενός οργανισμού ή μιας εταιρείας. Οι βασικές κατηγορίες διάθεσης ενός ERP συστήματος είναι η εγκατάσταση στις υποδομές του πελάτη (on-premise) και η εγκατάσταση σε cloud περιβάλλον (ολόκληρης του συστήματος ή μερικώς).

Στην συνέχεια είδαμε τα πλεονεκτήματα που προκύπτουν από την χρήση των ERP συστημάτων για έναν οργανισμό με βασικά την ενοποίηση όλων των εταιρικών δεδομένων σε ένα σύστημα, την υιοθέτηση μέσω αυτού best practise διαδικασιών, την εξάλειψη των διπλών καταχωρήσεων ίδιων διεργασιών, την πρόσβαση στην εταιρική πληροφορία σε πραγματικό χρόνο και τις ακριβείς λογιστικές αναφορές.

Έπειτα, εξετάσαμε τις βασικές διαφορές και τα αντίστοιχα πλεονεκτήματα και μειονεκτήματα μιας on-premise και μιας cloud εγκατάστασης με βασικές διαφορές το μειωμένο αρχικό κόστος επένδυσης και τον χρόνο imlementation της Software as a Service λύσης.

Στην συνέχεια κάναμε μία συνοπτική αναφορά στην ιστορική εξέλιξη των ERP συστημάτων με κομβικό σημείο την εμφάνιση του SAP R\3 συστήματος το 1992 που εισήγαγε την αρχιτεκτονική client-server στον χώρο και τέλος εξετάσαμε τις διαφορές μεταξύ των βασικών αρχιτεκτονικών σχεδιασμού ενός ERP συστήματος που είναι η διαμεσολάβηση ενός application layer μεταξύ του user interface του συστήματος και της βάσης δεδομένων που υπάρχει στα 3-tier συστήματα σε αντίθεση με τα 2-tier όπου η πρόσβαση στην βάση δεδομένων γίνεται μέσω του user interface της εφαρμογής.

ΚΕΦΑΛΑΙΟ 3 ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ ΣΥΣΤΗΜΑΤΟΣ

3.1 Ανάλυση απαιτήσεων συστήματος

Πριν την έναρξη της διαδικασίας δημιουργίας του ERP συστήματος που αποτελεί το θέμα της παρούσας διπλωματικής εργασίας ζητούμενο ήταν να γίνει μία ανάλυση των απαιτήσεων που υπήρχαν από το συγκεκριμένο σύστημα σε τρία βασικά επίπεδα. Πρώτον, στο επίπεδο των αναγκών των χρηστών του συστήματος που θα καλύπτονταν μέσα από τη χρήση του. Δεύτερον, στις τεχνικές προδιαγραφές που θα έπρεπε να έχει το σύστημα (αρχιτεκτονική επικοινωνίας,delivery model,τεχνολογίες υλοποίησης) και τρίτον στη δομή του user interface τμήματος της εφαρμογής (τον τρόπο με τον οποίο η εφαρμογή θα οδηγούσε τον χρήστη να ολοκληρώσει μία συγκεκριμένη εργασία).

ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ - ΚΑΛΥΨΗ ΑΝΑΓΚΩΝ ΧΡΗΣΤΩΝ

Η ανάλυση των αναγκών των χρηστών τις οποίες καλείται το προϊόν της παρούσας διπλωματικής εργασίας να καλύψει προέκυψαν από τη μελέτη βιβλιογραφίας του χώρου, τη συσχέτιση των αναγκών αυτών με τις παραδοσιακές λειτουργίες που προσφέρουν τα ERP συστήματα και τη μελέτη άλλων συστημάτων που απευθύνονται στον χώρο.

Οι βασικές κατηγορίες, που κρίθηκε ότι πρέπει να καλύπτει (εξ ολοκλήρου ή τμηματικά) σε επίπεδο δυνατοτήτων η εφαρμογή είναι :

I) Η διαχείριση αποθήκης ή Warehouse Management.

II) Η διαχείριση αποθεμάτων ή Inventory Management.

III) Η διαχείριση παραγγελιών και παραλαβών ή Orders and Delivery Management.

Επιπλέον των παραπάνω κατηγοριών κρίθηκε σκόπιμο να προστεθεί ένα επιπλέον στοιχείο που είναι η δυνατότητα επικοινωνίας και ανάθεσης εργασιών σε πραγματικό χρόνο μεταξύ των χρηστών. Το στοιχείο αυτό συνάδει με την έννοια της αδιάληπτης διαθεσιμότητας της επικοινωνίας χρήστη συστήματος και η οποία προκύπτει από την φύση του συστήματος προς υλοποίηση, δηλαδή ενός mobile ERP ,το οποίο και αίρει τους περιορισμούς που προκύπτουν από τη χρήση ενός σταθερού σταθμού εργασίας (διαθεσιμότητα επικοινωνίας χρήστη-συστήματος μόνο εφόσον ο χρήστης βρίσκεται στο σταθμό εργασίας του).

Στις βασικές κατηγορίες του συστήματος υπάρχουν υποκατηγορίες διαδικασιών που θα έπρεπε να εξυπηρετούνται με τη χρήση της εφαρμογής και οι οποίες θα συνέθεταν το σύνολο των δυνατοτήτων που θα παρείχε το σύστημα στους χρήστες ανά κατηγορία. Οι διαδικασίες αυτές κρίθηκε σκόπιμο να κατηγοριοποιηθούν περαιτέρω και σε επίπεδο χρηστών, δηλαδή διαφορετικές διαδικασίες που θα εξυπηρετούνταν για διαφορετικούς τύπους χρηστών. Οι τύποι

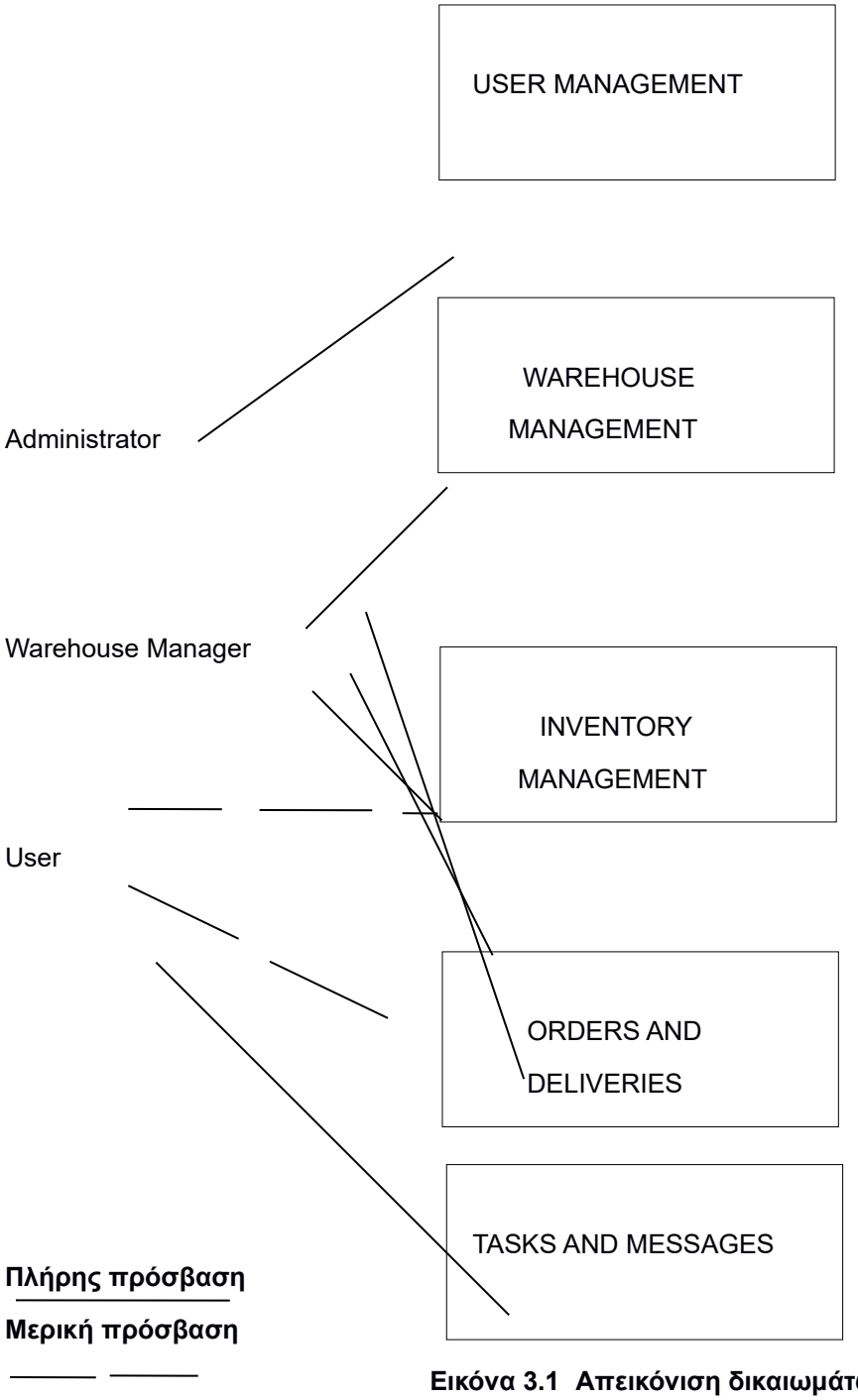
χρηστών του συστήματος κρίθηκε ότι πρέπει να είναι τρεις (administrator,warehouse manager και user).

Ο πρώτος τύπος χρήστη είναι ο administrator και ο οποίος δεν έχει δυνατότητα πρόσβασης και παραμετροποίησης του συστήματος σε επίπεδο Warehouse,Inventory ή Orders and Deliveries Management και του οποίου ο αποκλειστικός ρόλος είναι η διαχείριση των χρηστών του συστήματος (User Management).Σε ένα χρήστη administrator δίνεται η δυνατότητα να δημιουργήσει νέο χρήστη στο σύστημα με οποιονδήποτε ρόλο, να απενεργοποιήσει έναν υπάρχον ή να επεξεργαστεί το προφίλ ήδη υπάρχοντος χρήστη.

Ο δεύτερος τύπος χρήστη είναι ο warehouse manager ο οποίος είναι ο χρήστης που μπορεί να χρησιμοποιήσει το σύστημα στο πλήρες εύρος των δυνατοτήτων του. Συγκεκριμένα του δίνεται η δυνατότητα πρόσβασης χωρίς κανένα περιορισμό σε όλες τις κατηγορίες του κεντρικού μενού του συστήματος (Warehouse Management,Inventory Management,Orders and Deliveries και My Taks and Messages). Στην κατηγορία διαχείρισης αποθήκης μπορεί να εισάγει και να παραμετροποιήσει σε δεύτερο χρόνο τις αποθήκες που χρησιμοποιεί η εταιρεία . Ο ορισμός των παραμέτρων της κάθε αποθήκης καλύπτει όλα τα βασικά χαρακτηριστικά της (το όνομά της, τον αριθμό των διαδρόμων, τον αριθμό των θέσεων αποθήκευσης ανά διάδρομο, το μέγιστο βάρος αποθήκευσης ανά θέση και τις διαστάσεις της θέσης) και η εισαγωγή της στο σύστημα μπορεί να γίνει μόνο από χρήστη με ρόλο Warehouse Manager. Αντίστοιχα, μπορεί να προσθέσει στο σύστημα ένα νέο προϊόν και μία νέα κατηγορία προϊόντος κάνοντας έτσι το προϊόν διαθέσιμο για μελλοντικές παραγγελίες, έλεγχο αποθεματικών και άλλες επιλογές. Παράλληλα έχει πλήρη πρόσβαση στις άλλες κατηγορίες του κεντρικού μενού όπως η διαχείριση των αποθεματικών (Inventory Management) όπου μπορεί να ελέγξει το αποθεματικό ενός συγκεκριμένου προϊόντος ή να ενημερωθεί για τα προϊόντα που βρίσκονται κάτω από το όριο ασφαλείας αποθεματικό κ.α. Στην κατηγορία της διαχείρισης παραγγελιών και παραλαβών μπορεί να καταχωρήσει νέα παραγγελία και να ακυρώσει μία ήδη καταχωρημένη, μπορεί να ελέγξει ποιες παραγγελίες δεν έχουν παραληφθεί ακόμα (pending status) καθώς και να προχωρήσει στην καταχώρηση μιας παραλαβής μιας παραγγελίας που εκκρεμεί. Τέλος, έχει πρόσβαση στην ανταλλαγή μηνυμάτων με στόχο την επικοινωνία ή την ανάθεση κάποιας εργασίας σε έναν άλλο χρήστη του συστήματος.

Ο τρίτος τύπος χρήστη είναι ο user και ο οποίος έχει λιγότερες αρμοδιότητες από έναν warehouse manager το οποίο και αντανακλάται και σε επίπεδο πρόσβασης στις δυνατότητες του συστήματος. Ο τύπος χρήστη user δεν μπορεί να καταχωρήσει ή να παραμετροποιήσει τα χαρακτηριστικά μιας αποθήκης ,δεν μπορεί να εισάγει νέα προϊόντα στο σύστημα ή κατηγορίες προϊόντων. Στην κατηγορία διαχείρισης παραγγελιών του δίνεται η δυνατότητα καταχώρησης νέας παραγγελίας αλλά όχι ακύρωσης ήδη υπάρχουσας. Η αναζήτηση αποθεματικών των προϊόντων και η πρόσβαση στην επικοινωνία με άλλους χρήστες του επιτρέπεται χωρίς κάποιον περιορισμό.

Πρόσβαση στις δυνατότητες του συστήματος ανά κατηγορία χρήστη-συστήματος



Εικόνα 3.1 Απεικόνιση δικαιωμάτων χρηστών

3.2 Τεχνικές προδιαγραφές

Στο σημείο αυτό και αφού έχουμε αναφερθεί στις ανάγκες των χρηστών που καλύπτονται από το σύστημα θα αναλύσουμε τις τεχνικές προδιαγραφές του. Στις τεχνικές προδιαγραφές κατατάσσονται τόσο η αρχιτεκτονική σχεδίασης όσο και οι τεχνολογίες υλοποίησης της.

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΧΕΔΙΑΣΗΣ

Αρχικά πρέπει να οριστεί η κατηγορία του. Το σύστημα ανήκει στην ευρύτερη κατηγορία των πληροφοριακών συστημάτων, στην συνέχεια στην υποκατηγορία αυτών που ονομάζονται Enterprise Resource Planning και ακόμα πιο συγκεκριμένα στην υποκατηγορία των mobile ERP συστημάτων. Η αρχιτεκτονική των εν λόγω συστημάτων ακολουθεί το μοντέλο client-server με διαφοροποίηση στην υλοποίηση (2-tier μοντέλο,3-tier μοντέλο). Το σύστημα που αναπτύχθηκε στα πλαίσια της παρούσας εργασίας εγκαθιστά μία εφαρμογή (application) τοπικά σε μία κινητή συσκευή με λειτουργικό σύστημα Android OS η οποία λειτουργεί σαν client συσκευή σε ένα client-server μοντέλο. Η συσκευή επικοινωνεί με έναν web server που φιλοξενεί τα αρχεία κώδικα μέσω των οποίων γίνεται η επικοινωνία με τον database server και την βάση δεδομένων της εφαρμογής. Το αρχιτεκτονικό μοντέλο που χρησιμοποιήθηκε είναι το 3-tier μοντέλο με βάση το οποίο ο client δεν έχει απ' ευθείας πρόσβαση στην βάση δεδομένων αλλά μέσω ενός ενδιάμεσου (web στην συγκεκριμένη περίπτωση) server.

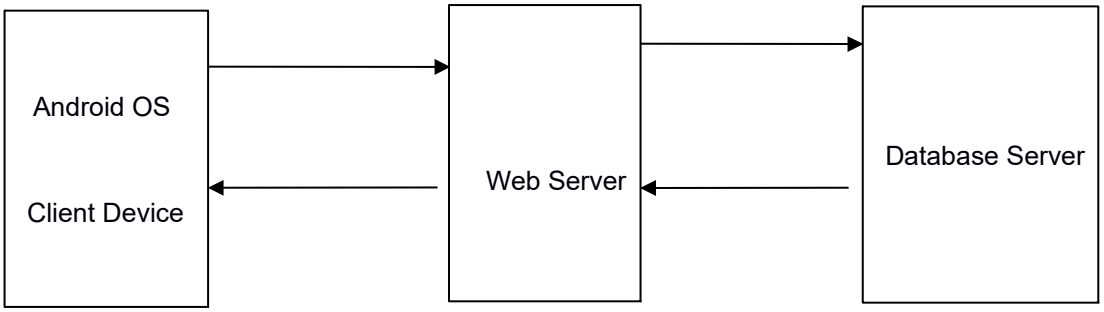
Από την παραπάνω περιγραφή μπορούμε να συλλέξουμε τα βασικότερα τεχνικά προαπαιτούμενα για να έχουμε πλήρη λειτουργία του συστήματος:

I) Συσκευή με λειτουργικό σύστημα Android OS (εγκατάσταση client εφαρμογής)

II) Web Server που θα φιλοξενεί τα αρχεία επικοινωνίας με την βάση δεδομένων

III) Σύστημα διαχείρισης βάσεων δεδομένων που θα φιλοξενεί την βάση της εφαρμογής

Σχηματική απεικόνιση ροής δεδομένων :



Εικόνα 3.2 Ροή δεδομένων μεταξύ application layers

ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ

Κάθε στάδιο της υλοποίησης της εφαρμογής χαρακτηρίζεται από το αντίστοιχο layer του 3-tier μοντέλου σχεδίασης. Ξεκινώντας από το πιο απομακρυσμένο επίπεδο από τον χρήστη (database layer) τα τρία στάδια αυτά είναι η σχεδίαση της βάσης δεδομένων και των σχέσεων των πινάκων που την απαρτίζουν, η δημιουργία των αρχείων του κώδικα που θα φιλοξενούνται σε έναν web server και τα οποία θα εκτελούν τις CRUD (Create,Read,Update,Delete) διαδικασίες στην επικοινωνία με τη βάση δεδομένων καθώς και το τελικό επίπεδο σχεδίασης (application layer) που αφορά τον κώδικα της android εφαρμογής που θα εγκατασταθεί στην συσκευή του τελικού χρήστη.

Κάθε ένα από αυτά τα στάδια προσφέρει μία σειρά επιλογών από διαφορετικές τεχνολογίες που μπορούν να υλοποιήσουν το τελικό ζητούμενο. Οι τεχνολογίες που επιλέχθηκαν για την ανάπτυξη του συστήματος ανά κατηγορία είναι η MySQL σαν Relational Database Management System και συγκεκριμένα η 10.1.32-MariaDB έκδοση ,ο Appache Web Server για να φιλοξενήσει τα αρχεία κώδικα που θα καλούνται από την εφαρμογή για να κάνουν την επικοινωνία με την βάση δεδομένων και σαν γλώσσα προγραμματισμού για τα αρχεία αυτά επιλέχθηκε η PHP (7.2.5 version).Για το επίπεδο της εφαρμογής (client layer) επιλέχθηκε το Android Studio IDE της Google και οι γλώσσες προγραμματισμού για την ανάπτυξη είναι η Java για το back end κομμάτι της εφαρμογής και η xml για την σχεδίαση του user interface κομματιού.

MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) ανοιχτού κωδικού το οποίο υποστηρίζεται από την εταιρεία Oracle και είναι βασισμένο στην SQL (Structured Query Language).Η MySQL είναι συμβατή με λειτουργικά συστήματα Windows,Linux και Unix.Παρόλο που η MySQL μπορεί να χρησιμοποιηθεί σε ένα μεγάλο εύρος εφαρμογών, συχνότερα χρησιμοποιείται σε εφαρμογές διαδικτύου και σε ηλεκτρονικές δημοσιεύσεις.
(Ορισμός searchoracle.techtarget.com)
Στην ανάπτυξη της εφαρμογής η MySQL χρησιμοποιήθηκε σαν το σύστημα διαχείρισης της βάσης δεδομένων (DBMS) και ο έλεγχος της έγινε με το εργαλείο ελέγχου PHP MyAdmin .

PHP

Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργασθεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script.

(Ορισμός από Wikipedia)

Στην ανάπτυξη της εφαρμογής η PHP χρησιμοποιήθηκε σαν server side script γλώσσα στην επικοινωνία του Apache web server με την βάση δεδομένων MySQL.

SQL

Η SQL (αγγλ αρκτ. από το Structured Query Language) είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. Η SQL ήταν μία από τις πρώτες γλώσσες για το σχεσιακό μοντέλο του Edgar F. Codd, στο σημαντικό άρθρο του το 1970, και έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές βάσεις δεδομένων.

(Ορισμός από Wikipedia)

Στην ανάπτυξη της εφαρμογής η SQL χρησιμοποιήθηκε σαν τμήμα του PHP κώδικα για τον σχηματισμό SQL queries που εκτελούν τις CRUD (CREATE,READ,UPDATE,DELETE) διαδικασίες με τις οποίες γίνεται η επικοινωνία με την βάση δεδομένων.

APACHE WEB SERVER

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Συντηρείται από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

(Ορισμός από Wikipedia)

Στην ανάπτυξη της εφαρμογής ο Apache χρησιμοποιήθηκε για να φιλοξενήσει τα αρχεία με τον PHP κώδικα που πραγματοποιούν την επικοινωνία με την βάση δεδομένων.

JAVA

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Πατέρας της θεωρείται ο James Gosling.

Σαν γλώσσα επίσημα εμφανίστηκε το 1995 (σαν μετονομασία της OAK). Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. (Ορισμός από computerhope.com)

Στην ανάπτυξη της εφαρμογής η Java χρησιμοποιήθηκε για την ανάπτυξη του back-end μέρους της εφαρμογής στο περιβάλλον ανάπτυξης Android Studio.

XML

Η XML (Extensible Markup Language) είναι μία γλώσσα σήμανσης η οποία ορίζει ένα σύνολο κανόνων για την κωδικοποίηση εγγράφων με μία μορφή η οποία είναι κατανοητή το ίδιο από ανθρώπους και υπολογιστές. Οι εφαρμογές που χρησιμοποιούν την XML περιέχουν έναν αναλυτή (parser) ο οποίος διαβάζει τον xml κώδικα και αναγνωρίζει την λειτουργία του κάθε τμήματος του εγγράφου και στην συνέχεια αποθηκεύει αυτήν την πληροφορία στην μνήμη του υπολογιστή κάνοντας την διαθέσιμη στο υπόλοιπο πρόγραμμα.

(Ορισμός από quora.com)

Στην ανάπτυξη της εφαρμογής η XML χρησιμοποιήθηκε για τον σχεδιασμό του user interface τμήματος της εφαρμογής στο περιβάλλον ανάπτυξης Android Studio.

3.3 Η βάση δεδομένων

Η βάση δεδομένων που αποτελεί το πρώτο επίπεδο του συστήματος σχεδιάστηκε ώστε να περιγράφει τις βασικές οντότητες που συνθέτουν το σύστημα καθώς και τις μεταξύ τους συσχετίσεις. Αποτελείται από επτά πίνακες καθένας εκ των οποίων φέρει μια σειρά από γνωρίσματα/ιδιότητες που περιγράφουν την οντότητα.

Αρχικά υπάρχει ο πίνακας *users*.Ο πίνακας χρησιμοποιείται για να αποθηκεύει τους χρήστες του συστήματος και τις ιδιότητες που περιγράφουν έναν χρήστη. Τα πεδία που συνθέτουν τον πίνακα είναι τα User_id (που αποτελεί και το πρωτεύον κλειδί του πίνακα),Username και Password που αποθηκεύουν τα στοιχεία σύνδεσης κάθε χρήστη στο σύστημα, τα πεδία First_name και Last_name που αποθηκεύουν το όνομα και το επώνυμο του χρήστη, πεδίο User_role που περιγράφει τον ρόλο του χρήστη στο σύστημα (administrator,warehouse manager,user) και το πεδίο Flag_active που είναι τύπου boolean και δείχνει στο σύστημα αν ο χρήστης είναι ενεργός ή έχει απενεργοποιηθεί από κάποιον administrator.

Στην συνέχεια υπάρχει ο πίνακας *warehouses*.Ο πίνακας χρησιμοποιείται για να περιγράψει την οντότητα αποθήκη, που στο σύστημα μας αποτελεί τον βασικό αποθηκευτικό

χώρο των προϊόντων των οποίων την αποθήκευση διαχειρίζεται η εταιρεία. Τα πεδία που συνθέτουν τον πίνακα είναι τα Warehouse_id (πρωτεύον κλειδί του πίνακα),Warehouse_name για το όνομα της αποθήκης,Num_of_paths και Spots_per_path που αντιπροσωπεύουν τον αριθμό των διαδρόμων μιας αποθήκης και τον αριθμό των θέσεων αποθήκευσης σε κάθε διάδρομο,Max_spot_weight που αντιπροσωπεύει το μέγιστο συνολικό βάρος σε κιλά των αποθηκευμένων προϊόντων σε μία θέση καθώς επίσης και τα Spot_length,Spot_width και Spot_height στα οποία αποθηκεύονται οι διαστάσεις σε μέτρα της κάθε θέσης αποθήκευσης.

Επόμενος πίνακας είναι ο *item_categories* που αποθηκεύει τις κατηγορίες των προϊόντων που εντάσσονται στο σύστημα από τον warehouse_manager.Τα πεδία που τον συνθέτουν είναι το Item_Category_id (πρωτεύον κλειδί) και το item_category_name όπου αποθηκεύεται το όνομα της κάθε κατηγορίας.

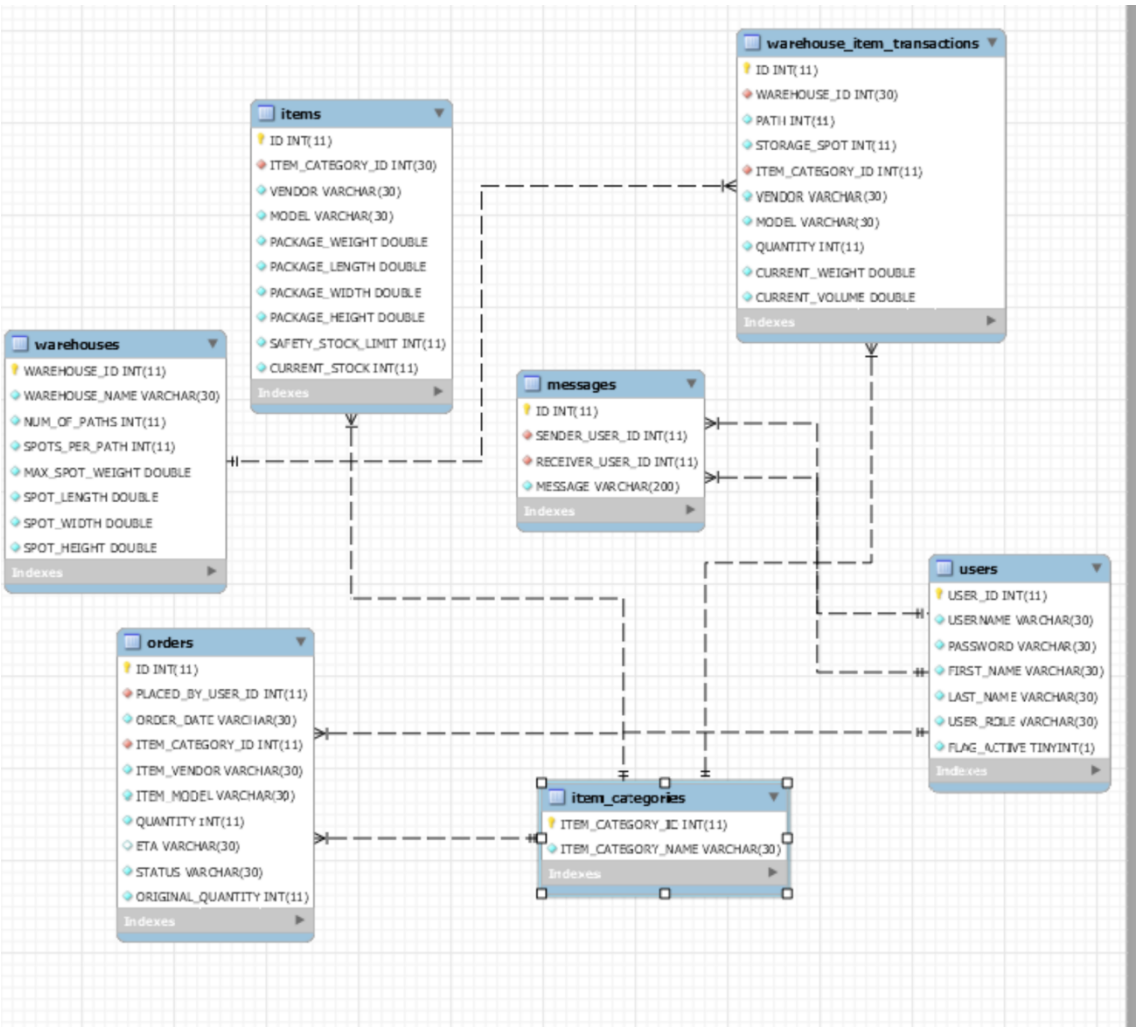
Στην συνέχεια υπάρχει ο πίνακας *items* που περιγράφει την οντότητα των προϊόντων. Οι ιδιότητες της οντότητας προϊόν που εκφράζονται μέσα από τα πεδία του πίνακα είναι οι ID (πρωτεύον κλειδί), Item_category_id που περιγράφει την κατηγορία προϊόντος (ξένο κλειδί στον πίνακα Item_Categories και την στήλη Item_Category_Id),Vendor όπου αποθηκεύεται ο κατασκευαστής του προϊόντος,Model όπου αποθηκεύεται το συγκεκριμένο μοντέλο που διαφοροποιεί προϊόντα ίδιας κατηγορίας (και ίδιου κατασκευαστή), τα Package_weight,Package_Length,Package_Width,Package_Height που περιγράφουν φυσικές μεταβλητές της συσκευασίας όπως το βάρος και οι διαστάσεις της και τέλος τα πεδία Safety_stock_limit και Current_stock μέσω των οποίων γίνεται έλεγχος διαθεσιμότητας του προϊόντος (Current_stock) και ασφάλειας αποθεματικού (διαφορά Current_Stock από Safety_stock_limit).

Έπειτα, ο πίνακας *messages* περιγράφει την οντότητα των μηνυμάτων που ανταλλάσσονται μεταξύ των χρηστών και τα πεδία που περιγράφουν τις ιδιότητες του είναι τα ID (πρωτεύον κλειδί),Sender_User_id και Receiver_User_id για να αποθηκεύσουν τον αποστολέα και τον παραλήπτη των μηνυμάτων (και οι δύο στήλες έχουν σχέση ξένου κλειδιού με την στήλη User_id του πίνακα Users) καθώς επίσης και το πεδίο message που αποθηκεύει το περιεχόμενο του μηνύματος που απεστάλη.

Στην συνέχεια, ο πίνακας *orders* περιγράφει την οντότητα της παραγγελίας που είναι η βασική οντότητα της κατηγορίας Orders and Deliveries.Τα πεδία που την συνθέτουν είναι τα ID (πρωτεύον κλειδί),Placed_by_User_id που καταχωρείται ο χρήστης που πραγματοποίησε την παραγγελία (ξένο κλειδί στον πίνακα Users και την στήλη User_id),Order_date για την αποθήκευση της ημερομηνίας παραγγελίας, τα πεδία Item_category,Item_vendor και Item_model τα οποία και αποθηκεύουν τις τιμές για την κατηγορία, τον κατασκευαστή και το μοντέλο του προϊόντος, το πεδίο Quantity όπου καταχωρείται ποσότητα του προϊόντος που δόθηκε ως παραγγελία καθώς και τα πεδία ETA,Status και Original_quantity όπου αντίστοιχα χρησιμοποιούνται για την αποθήκευση της εκτιμώμενης ημερομηνίας παράδοσης, της κατάστασης της παραγγελίας (Pending,Delivered,Canceled) καθώς και της αποθήκευσης εκ νέου της αρχικής ποσότητας της παραγγελίας για λόγους προγραμματιστικούς.

Τέλος, ο πίνακας *Warehouse_item_transactions* περιγράφει τις διαδικασίες προσθαφαιρέσεων προϊόντων από τις αποθήκες και τα δεδομένα της κάθε συναλλαγής. Τα πεδία που τον συνθέτουν είναι τα ID (πρωτεύον κλειδί), Warehouse_id για την καταγραφή της αποθήκης που γίνεται η συναλλαγή (ξένο κλειδί στον πίνακα Warehouses και την στήλη

Warehouse_id), Path και Storage_Spot για τον διάδρομο και την θέση αποθήκευσης,Item_category_id (ξένο κλειδί στον πίνακα Item_Categories),Vendor και Model για την αποθήκευση αντίστοιχα της κατηγορίας ,του κατασκευαστή και του μοντέλου του προϊόντος, Quantity για την ποσότητα της συναλλαγής,current_weight και current_volume για το τρέχον βάρος και όγκο που βρίσκεται στην συγκεκριμένη θέση.

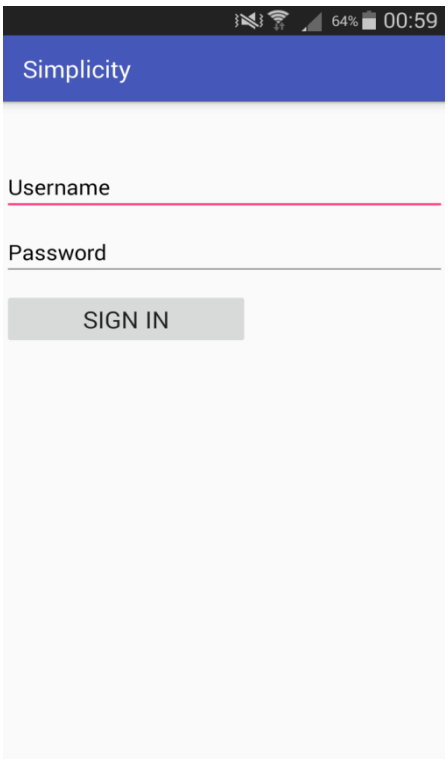


Εικονα 3.3 Διάγραμμα Συσχετίσεων Βάσης Δεδομένων

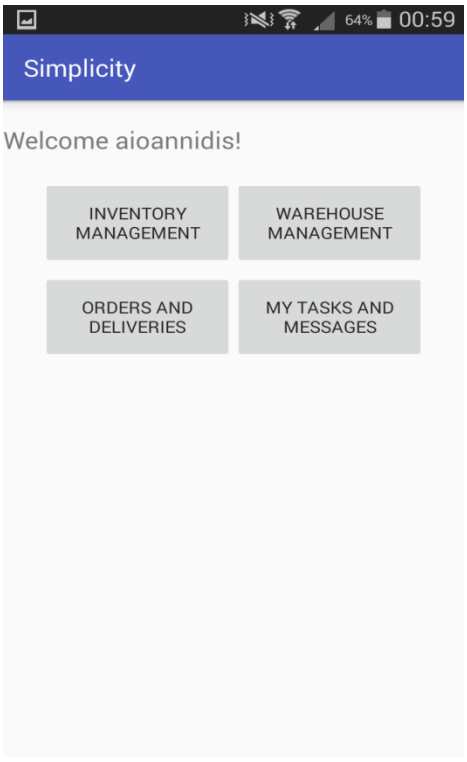
3.4 ΔΟΜΗ USER INTERFACE

Στο επίπεδο της εφαρμογής και με δεδομένο ότι ένας εκ των στόχων της ήταν η απλότητα στην χρήση και το φιλικό προς τον χρήστη περιβάλλον επιλέχτηκε η δομή του user interface να οδηγεί τον χρήστη στην ολοκλήρωση μιας ενέργειας μέσα από μία σειριακή επιλογή των υπερατηγοριών της διαδικασίας προς εκτέλεση. Κάθε οθόνη έχει μικρό αριθμό επιλογών και καθόλου κρυμμένα μενού με συνέπεια ο χρήστης να οδηγείται σε κάθε του επιλογή ένα βήμα πιο κοντά στην τελική ενέργεια.

Παραδείγματα οθονών επιλογής

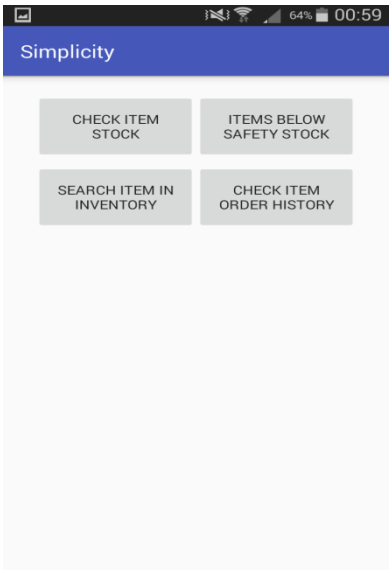


Εικόνα 3.4 Οθόνη Σύνδεσης



Εικόνα 3.5 Κεντρική οθόνη χρήστη

Τα χρώματα προτιμήθηκε να είναι ουδέτερα και όσο το δυνατόν λιγότερο κουραστικά και τα κουμπιά ή οι υποδοχείς κειμένου να έχουν όσο το δυνατόν πιο κατατοπιστικά κείμενα ώστε να μην προκαλείται καμία σύγχυση για το πως θα οδηγηθεί στην επιθυμητή ενέργεια.



Εικόνα 3.6 Μενού inventory management

3.5 ΣΥΝΟΨΗ ΚΕΦΑΛΑΙΟΥ

Στο κεφάλαιο αυτό προσπαθήσαμε να δούμε τη διαδικασία προετοιμασίας και σχεδιασμού που προηγείται της υλοποίησης της εφαρμογής και των επιλογών που έγιναν και αφορούν τις τεχνολογίες υλοποίησης. Αρχικά είδαμε το ποιες θα ήταν οι ανάγκες που το σύστημα θα κάλυπτε έτσι ώστε να υπάρχει ένα σύνολο ζητουμένων/στόχων από την εφαρμογή και στην συνέχεια είδαμε την αρχιτεκτονική της σχεδίασης και την ροή της πληροφορίας μεταξύ των 3 επιπέδων (client - web server - database server).Στην συνέχεια κάναμε μία αναφορά στις τεχνολογίες που επιλέχθηκε να υλοποιήσουν την εφαρμογή και είδαμε τα βασικά χαρακτηριστικά κάθε μίας. Τέλος είδαμε ποιες ήταν οι βασικές επιλογές που θα οδηγούσαν στην σχεδίαση του user interface (απλότητα στην χρήση, ουδέτερα χρώματα, όχι κρυμμένα μενού).

ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ

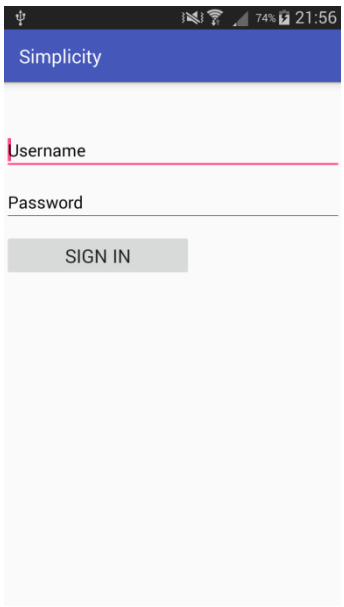
4.1 Σενάριο ελέγχου συστήματος

Στο κεφάλαιο αυτό θα περιγράψουμε τις δυνατότητες του συστήματος μέσα από ένα δεδομένο σενάριο ελέγχου. Στο σενάριο ελέγχου αυτό θα αντιμετωπίσουμε το σύστημα ως ένα σύστημα, που μόλις εγκαταστάθηκε, με μία άδεια βάση δεδομένων ακριβώς όπως θα συνέβαινε σε μία εταιρεία η οποία θα είχε μόλις προχωρήσει στην εγκατάσταση του.

Η μόνη εγγραφή που θα υπάρχει στην βάση δεδομένων θα είναι στον πίνακα users όπου και θα υπάρχει από εγκατάστασης ένας χρήστης με username administrator και με administrator ρόλο έτσι ώστε να μπορεί να δημιουργήσει τους χρήστες της εφαρμογής.

ΒΗΜΑ 1 - ΣΥΝΔΕΣΗ ΣΤΗΝ ΕΦΑΡΜΟΓΗ

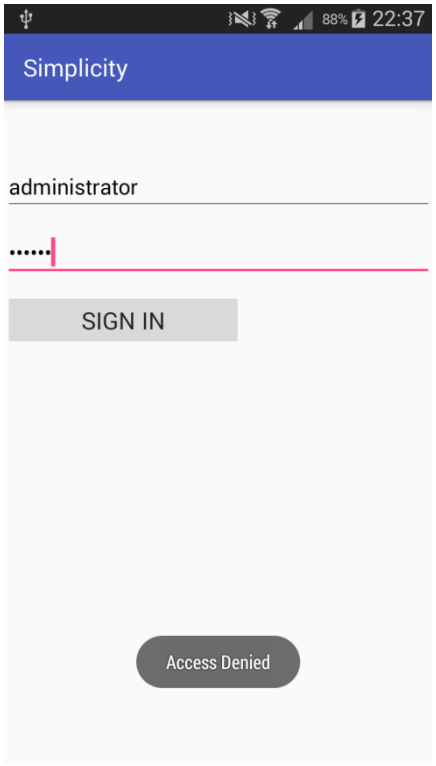
Αφού έχει ολοκληρωθεί η εγκατάσταση της εφαρμογής στην κινητή συσκευή με λειτουργικό σύστημα Android που θα την φιλοξενήσει,εκκινούμε την εφαρμογή και βρισκόμαστε στην πρώτη οθόνη της.



Εικόνα 4.1 Οθόνη Σύνδεσης

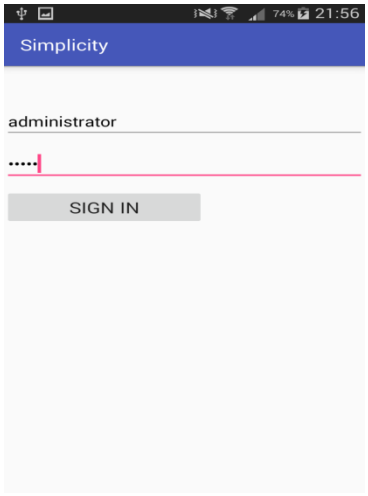
Στην πρώτη οθόνη μπορεί ο χρήστης να διακρίνει το όνομα της εφαρμογής (Simplicity) στο πάνω μέρος της και μέσα σε ένα πλαίσιο το οποίο στην ορολογία των android εφαρμογών ονομάζεται actionbar.Κάτω από το actionbar υπάρχουν δύο αντικείμενα τύπου edittext,δηλαδή δύο υποδοχείς κειμένου, τα οποία οδηγούν τον χρήστη στην συμπλήρωσή τους μέσω της

ιδιότητας android:hint η οποία στην πρώτη περίπτωση έχει τιμή Username και στην δεύτερη Password. Τέλος, υπάρχει ένα αντικείμενο τύπου button το οποίο έχει κείμενο SIGN IN για να οδηγήσει τον χρήστη να το πατήσει για να συνδεθεί στην εφαρμογή και του οποίου το πάτημα οδηγεί στην αποστολή των δύο αναγνωριστικών (username και password) που πληκτρολόγησε ο χρήστης στα αντίστοιχα πεδία από τον client της εφαρμογής (smartphone) στον application server και σε ένα συγκεκριμένο αρχείο του. Το αρχείο αυτό λαμβάνει τα δύο αναγνωριστικά και στην συνέχεια επικοινωνεί με την βάση δεδομένων ρωτώντας αν υπάρχει ο συγκεκριμένος χρήστης στο σύστημα, αν ο κωδικός ασφαλείας είναι αυτός που δόθηκε και αν ο χρήστης είναι στην κατηγορία των ενεργών χρηστών στην περίπτωση που υπάρχει καταχωρημένος στο σύστημα καθώς και το ποιος είναι ο ρόλος του χρήστη στο σύστημα. Αν οι τρείς πρώτες συνθήκες ικανοποιούνται (υπάρχει ο χρήστης, έχει δοθεί σωστός κωδικός και ο χρήστης είναι ενεργός) τότε ανάλογα με τον ρόλο του χρήστη η εφαρμογή τον οδηγεί στην αντίστοιχη δεύτερη οθόνη. Σε αντίθετη περίπτωση τον ενημερώνει ότι δεν του επιτρέπεται η πρόσβαση.

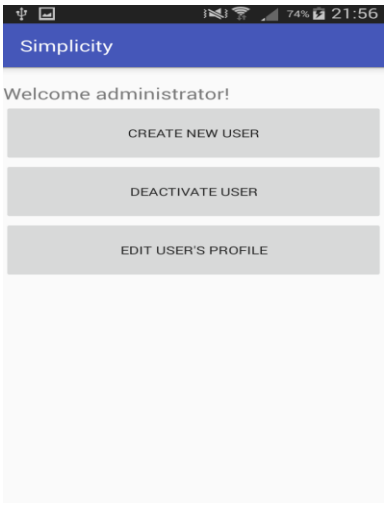


Εικόνα 4.2 Οθόνη σύνδεσης 2

Στην εικόνα 4.2 φαίνεται το αποτέλεσμα του συστήματος όταν δοθούν λανθασμένα αναγνωριστικά σύνδεσης.



Εικόνα 4.3 Οθόνη σύνδεσης 3



Εικόνα 4.4 Μενού διαχειριστή

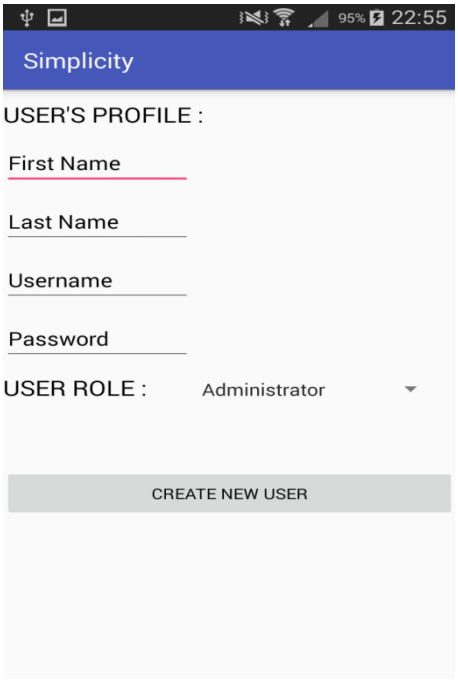
ΒΗΜΑ 2 - ΔΗΜΙΟΥΡΓΙΑ ΝΕΟΥ ΧΡΗΣΤΗ

Στην εικόνα 4.4 φαίνεται το αρχικό μενού του χρήστη adminisistrator μετά από επιτυχή σύνδεση και αφού προηγουμένως έχουν δοθεί σωστά αναγνωριστικά σύνδεσης (εικόνα 4.3).

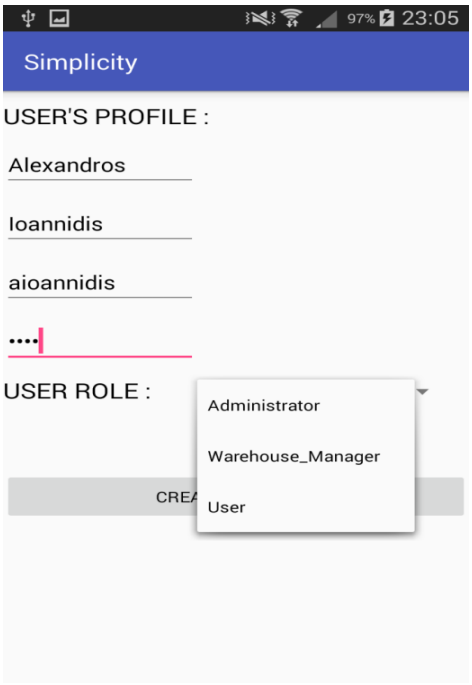
Στην συνέχεια ο χρήστης επιλέγει Create New User πατώντας το αντίστοιχο button με σκοπό να δημιουργήσουμε τον πρώτο χρήστη του συστήματος.

Στην νέα οθόνη ζητάται από το σύστημα να εισαχθούν οι απαραίτητες πληροφορίες για την δημιουργία του χρήστη (εικόνα 4.5) όπως το όνομα του, το επώνυμο του, το όνομα χρήστη του και τον κωδικό ασφαλείας του.

Στην συνέχεια και για να ολοκληρωθεί η διαδικασία δημιουργίας του νέου λογαριασμού επιλέγεται από το drop down menu (εικόνα 4.6 -αντικείμενο τύπου spinner) ο ρόλος που θα έχει στο σύστημα ο νέος χρήστης και ολοκληρώνεται η ενέργεια με το πάτημα του create new user button.



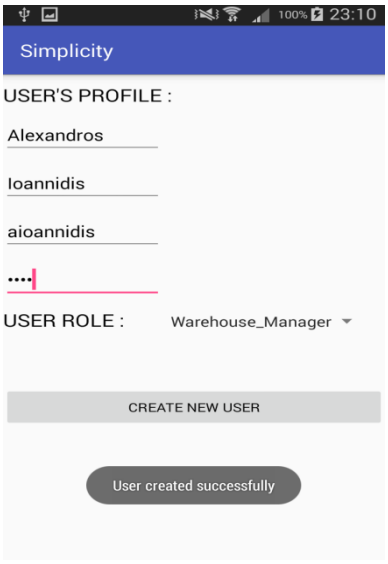
Εικόνα 4.5



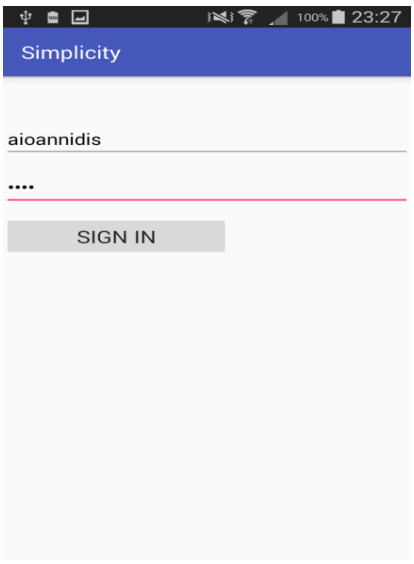
Εικόνα 4.6

Στο πάτημα του button και με την επιτυχημένη εισαγωγή του χρήστη το σύστημα ενημερώνει με ένα μήνυμα ότι η διαδικασία ολοκληρώθηκε επιτυχώς (εικόνα 4.7).

ΒΗΜΑ 3 - ΣΥΝΔΕΣΗ ΝΕΟΥ ΧΡΗΣΤΗ



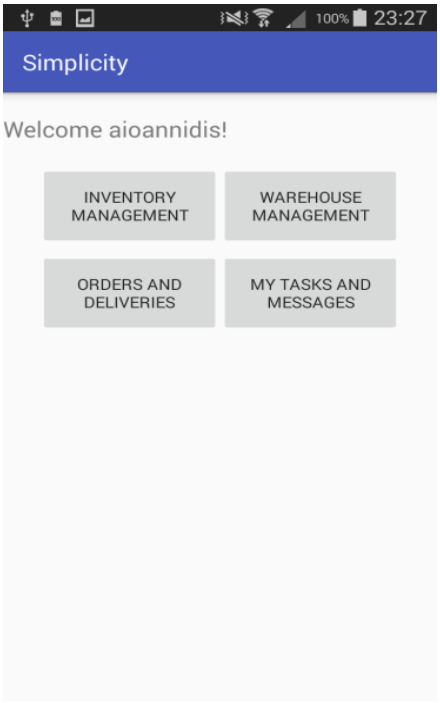
Εικόνα 4.7 Καταχώρηση χρήστη



Εικόνα 4.8 Σύνδεση νέου χρήστη

Αφού έχει πλέον δημιουργηθεί με επιτυχία ο δεύτερος χρήστης του συστήματος με ρόλο Warehouse Manager γίνεται δοκιμή σύνδεσης στην εφαρμογή με τον νέο χρήστη.

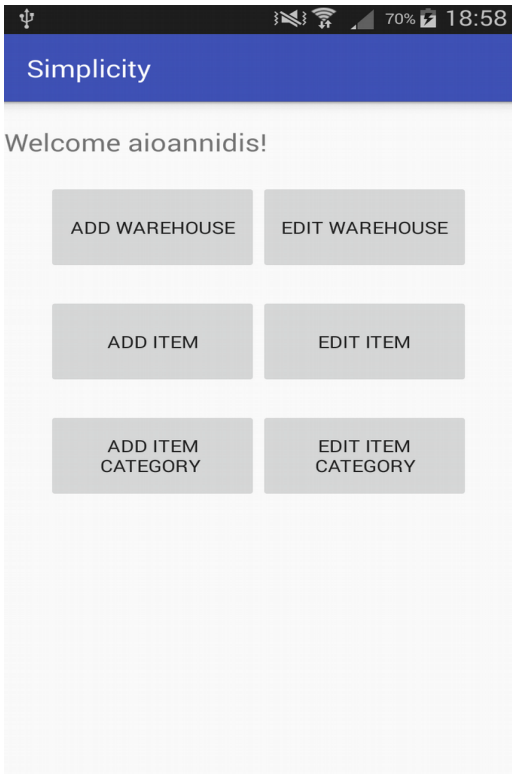
Μετά από επανεκκίνηση της εφαρμογής γίνεται σύνδεση στο σύστημα ως aioannidis που είναι ο νέος χρήστης (εικόνα 4.8).Μετά από την επιτυχημένη σύνδεση ο χρήστης βρίσκεται στην σελίδα με το κεντρικό μενού του (εικόνα 4.9).Κάτω από ένα μήνυμα καλωσορίσματος στο πάνω μέρος της οθόνης ακολουθεί το μενού με τις βασικές επιλογές/κατηγορίες εργασιών που μπορεί να μεταβεί.Αυτές είναι οι inventory management,warehouse management,orders and deliveries και my tasks and messages.



Εικόνα 4.9 Μενού W.Manager

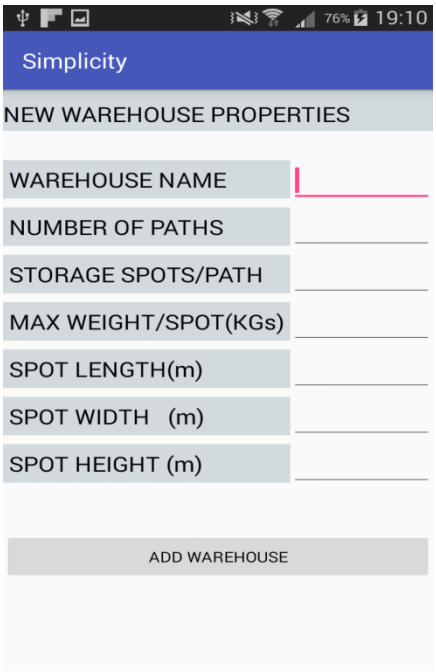
ΒΗΜΑ 4 - ΚΑΤΑΧΩΡΗΣΗ ΝΕΑΣ ΑΠΟΘΗΚΗΣ

Η πρώτη ενέργεια του νέου χρήστη στο σύστημα θα είναι η καταχώρηση μιας νέας αποθήκης έτσι ώστε να υπάρχουν τα δεδομένα (Όνομα,διάδρομος,θέση αποθήκευσης,χαρακτηριστικά αποθηκευτικής θέσης) για να μπορούν αργότερα να αποθηκευθούν τα προϊόντα από μία παραλαληφθείσα παραγγελία. Από το κεντρικό με επιλογή του button με περιγραφή Warehouse Management οδηγείται ο χρήστης στην κεντρική οθόνη της συγκεκριμένης κατηγορίας.

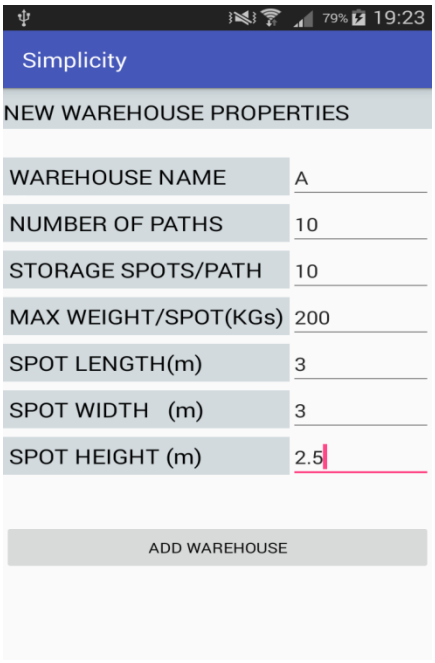


Εικόνα 4.10 Μενού Warehouse Management

Οι επιλογές που δίνονται στην συγκεκριμένη κατηγορία είναι η προσθήκη/καταχώρηση μιας νέας αποθήκης στο σύστημα και η επεξεργασία μιας ήδη υπάρχουσας, η προσθήκη ενός νέου προϊόντος στο σύστημα και η επεξεργασία ενός υπάρχοντος καθώς και η προσθήκη μιας νέας κατηγορίας προϊόντων ή επεξεργασία μιας υπάρχουσας κατηγορίας. Πατώντας το button με περιγραφή 'ADD WAREHOUSE' οδηγείται ο χρήστης στην οθόνη καταχώρησης νέας αποθήκης. Στην οθόνη αυτή (εικόνα 4.11) καλείται να εισάγει ένα όνομα για την αποθήκη που καταχωρηθεί στο σύστημα και με το οποίο θα γίνεται η αναγνώριση της σε όλες τις υπόλοιπες περιπτώσεις που θα χρησιμοποιείται στο σύστημα. Αντίστοιχα, ,καλείται να συμπληρώσει τις τιμές για τα άλλα χαρακτηριστικά της όπως ο αριθμός των διαδρόμων αποθήκευσης, οι αποθηκευτικές θέσεις ανά διάδρομο καθώς επίσης και το μέγιστο βάρος αποθήκευσης ανά θέση μαζί με τις διαστάσεις αυτής (εικόνα 4.12).

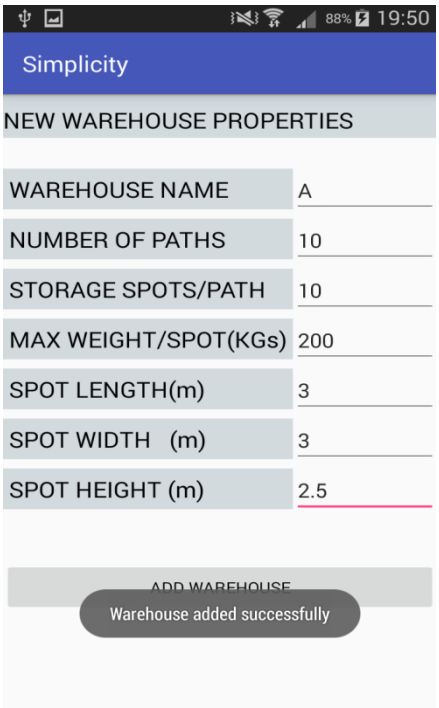


Εικόνα 4.11 Καταχώρηση αποθήκης



Εικόνα 4.12 Καταχώρηση αποθήκης 2

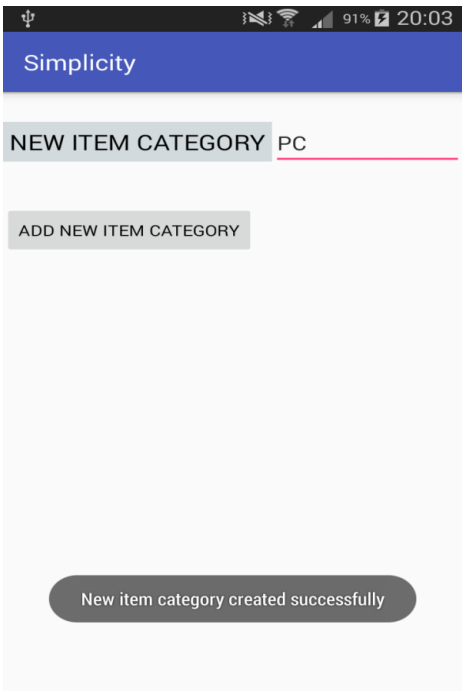
Αφού έχουν συμπληρωθεί όλα τα πεδία ο χρήστης πατάει το add warehouse button και ενημερώνεται με μήνυμα για την επιτυχή καταχώρηση (εικόνα 4.13).



Εικόνα 4.13 Επιτυχής καταχώρηση

ΒΗΜΑ 5 - ΚΑΤΑΧΩΡΗΣΗ ΝΕΩΝ ΚΑΤΗΓΟΡΙΩΝ ΠΡΟΪΟΝΤΩΝ

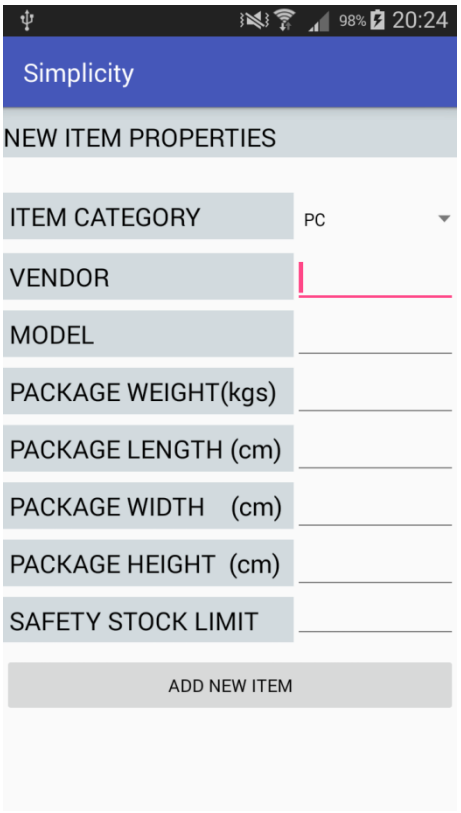
Μετά την καταχώρηση της νέας αποθήκης ο χρήστης επιστρέφει στην προηγούμενη οθόνη με χρήση του back button της συσκευής και από το μενού της Warehouse Management κατηγορίας επιλέγει add item category. Στην συνέχεια καταχωρεί την κατηγορία PC πληκτρολογώντας το όνομα της στον υποδοχέα κειμένου της οθόνης και πατώντας add new item category (εικόνα 4.14).



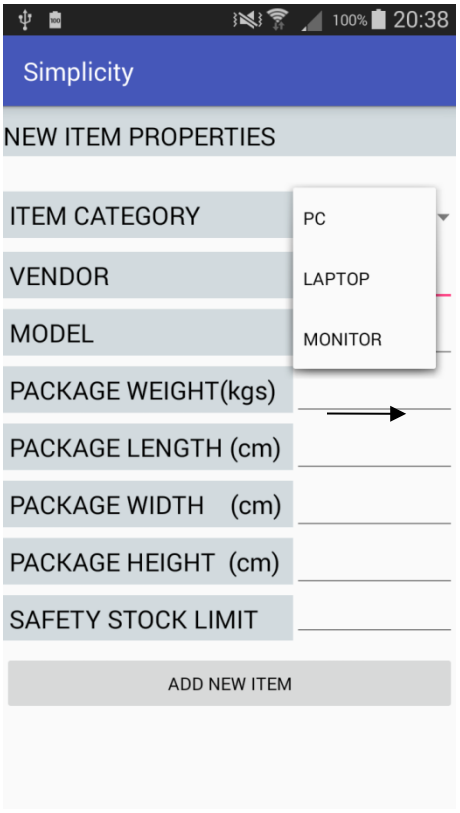
Εικόνα 4.15 Καταχώρηση κατηγορίας προϊόντος

Μετά την επιτυχή καταχώρηση για την οποία ειδοποιείται με αντίστοιχο μήνυμα από την εφαρμογή προχωρά στην καταχώρηση άλλων δύο κατηγοριών (LAPTOP,MONITOR).Στην συνέχεια και αφού έχουν καταχωρηθεί στο σύστημα και η πρώτη αποθήκη αλλά και οι πρώτες κατηγορίες προϊόντων ο χρήστης θα προχωρήσει στην καταχώρηση νέων συγκεκριμένων προϊόντων. Πατώντας το back button και επιλέγοντας add item βρίσκεται στην οθόνη καταχώρησης νέου προϊόντος(εικόνα 4.16).

ΒΗΜΑ 6 - ΚΑΤΑΧΩΡΗΣΗ ΝΕΩΝ ΠΡΟΪΟΝΤΩΝ



Εικόνα 4.16 Καταχώρηση νέου item



Εικόνα 4.17 Καταχώρηση νέου item 2

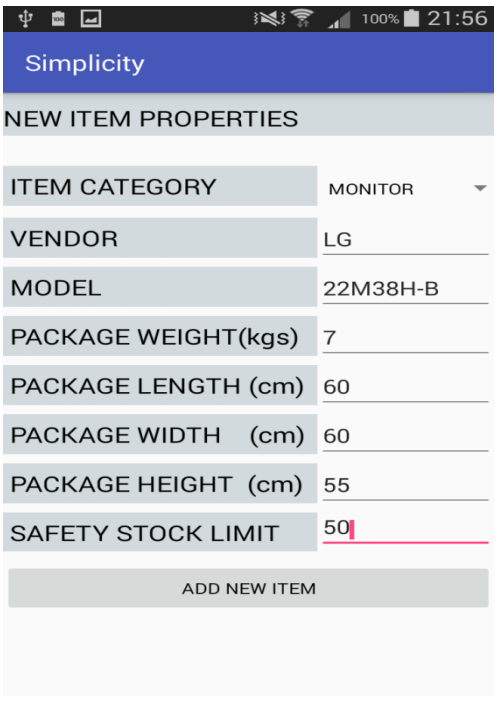
Στην οθόνη καταχώρησης ο χρήστης επιλέγει από το drop down μενού της επιλογής Item Category την κατηγορία προϊόντος.Οι επιλογές που προσφέρονται είναι οι κατηγορίες προϊόντων που καταχωρήθηκαν στο σύστημα στο προηγούμενο βήμα.Τα στοιχεία τα οποία καλείται να συμπληρώσει ο χρήστης και τα οποία συνθέτουν την περιγραφή του προϊόντος είναι η κατηγορία του,ο κατασκευαστής του προϊόντος (Vendor) ,το συγκεκριμένο μοντέλο του κατασκευαστή (Model) όπως επίσης και τα χαρακτηριστικά της συσκευασίας του,δηλαδή το βάρος του πακέτου και οι διαστάσεις του (Package Weight,Package Length,Width,Height).Τέλος,ορίζεται από τον χρήστη και το όριο ασφαλείας του αποθεματικού του προϊόντος στην αποθήκη.Με την επιλογή add new item ολοκληρώνεται η καταχώρηση στο σύστημα.Με την παραπάνω διαδιακασία καταχωρούνται από τον χρήστη τα ακόλουθα προϊόντα των πινάκων 4.1 και 4.2:

Πίνακας 4.1

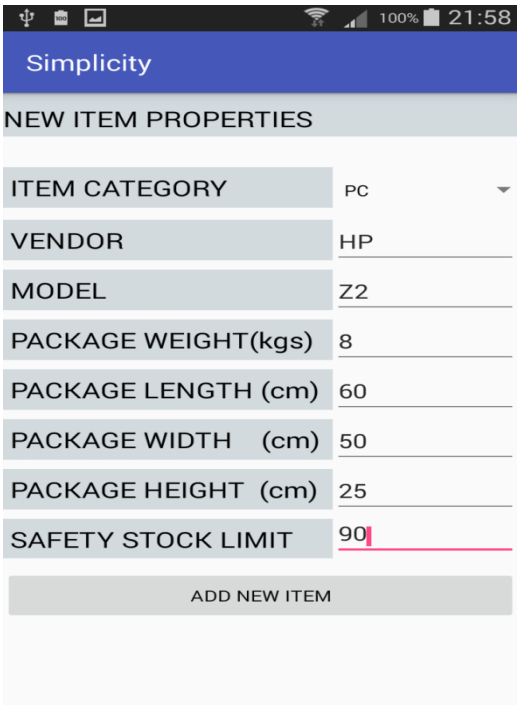
Category	Vendor	Model	Weight	Length	Width	Height	Sf.Stock
PC	DELL	INSPIRON 36-70	5	30	60	60	100
LAPTOP	HP	G-6	4	65	45	20	120
LAPTOP	TURBO-X	STEEL MX II	4,5	55	40	15	70
LAPTOP	DELL	VOSTRO 5000	3,5	50	40	18	80
MONITOR	LG	22M38H-B	7	60	60	55	50

Πίνακας 4.2

Category	Vendor	Model	Weight	Length	Width	Height	Sf.Stock
PC	HP	Z2	8	60	50	25	90
MONITOR	AOC	G2790PX	7,5	64	45	45	60



Εικόνα 4.18 Καταχώρηση item 3

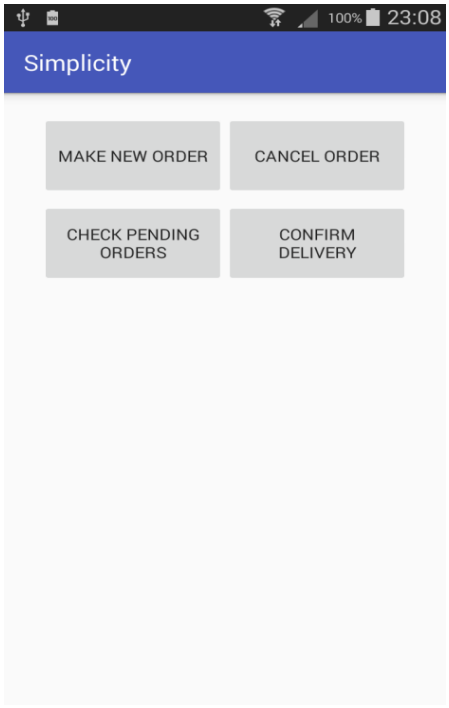


Εικόνα 4.19 Καταχώρηση item 4

Με την καταχώρηση των προϊόντων στο σύστημα ο χρήστης μπορεί να κάνει χρήση και άλλων δυνατοτήτων του συστήματος όπως η πραγματοποίηση παραγγελιών.Στο επόμενο βήμα ο χρήστης επιστρέφει στην κεντρική οθόνη του και από εκεί μεταβαίνει πλεόν στην κατηγορί των orders and deliveries.

ΒΗΜΑ 7 - ΚΑΤΑΧΩΡΗΣΗ ΝΕΩΝ ΠΑΡΑΓΓΕΛΙΩΝ

Έχοντας φτάσει στο σημείο που ο χρήστης έχει καταχωρήσει τις αποθήκες και τα προϊόντα μπορεί να προχωρήσει στην διαδικασία δημιουργίας παραγγελίας για κάποια από τα προϊόντα αυτά.Επιστρέφοντας στην κεντρική οθόνη του και επιλέγοντας την κατηγορία orders and deliveries μεταβαίνει στην οθόνη του μενού επιλογών της κατηγορίας (εικόνα 4.20).

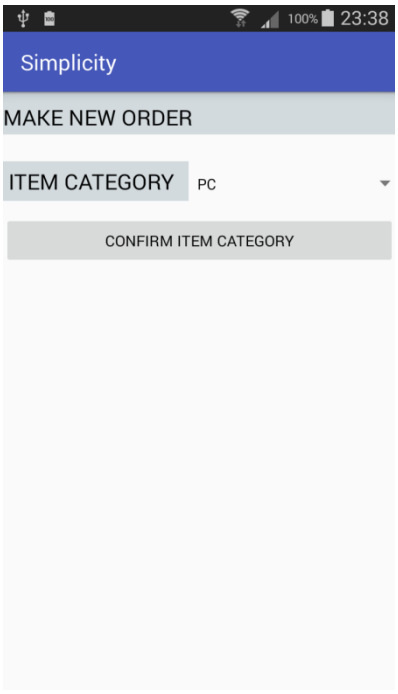


Εικόνα 4.20 Μενού orders

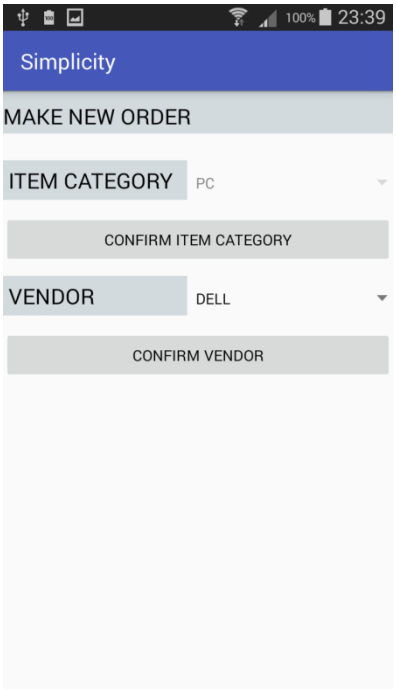
Επιλέγοντας make new order μεταβαίνει στην οθόνη της δημιουργίας νέας παραγγελίας (εικόνα 4.21) όπου καλείται να επιλέξει το προϊόν και την ποσότητα οδηγούμενος από μία σειρά drop down menus τριών spinner objects.Τα drop down menus αυτά οδηγούν τον χρήστη να επιλέξει πρώτα την κατηγορία του προϊόντος,στην συνέχεια τον κατασκευαστή και τέλος το μοντέλο του προϊόντος διευκολύνοντας έτσι την διαδικασία επιλογής του προς παραγγελία προϊόντος (εικόνα 4.22).Έχοντας επιλέξει τα παραπάνω καλείται να δώσει την επιθυμητή ποσότητα στο σύστημα και να καταχωρήσει την παραγγελία.Στο σημείο αυτό ο χρήστης καταχωρεί την ακόλουθη παραγγελία :

Πίνακας 4.3

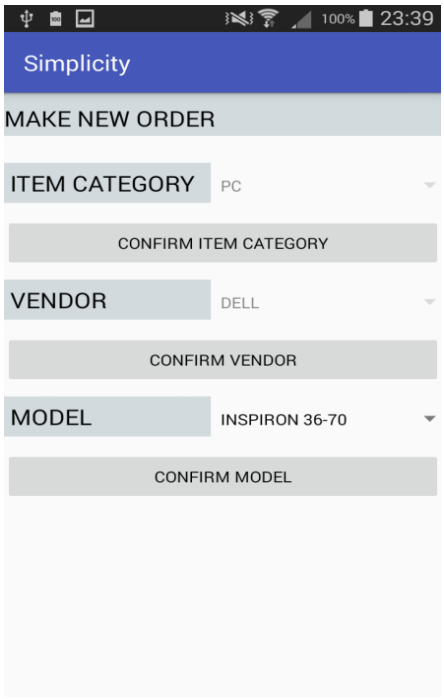
CATEGOTY	VENDOR	MODEL	QUANITY
PC	DELL	INSPIRON 36-70	50
LAPTOP	HP	G-6	70
LAPTOP	DELL	VOSTRO 5000	200
MONITOR	AOC	G2790PX	45



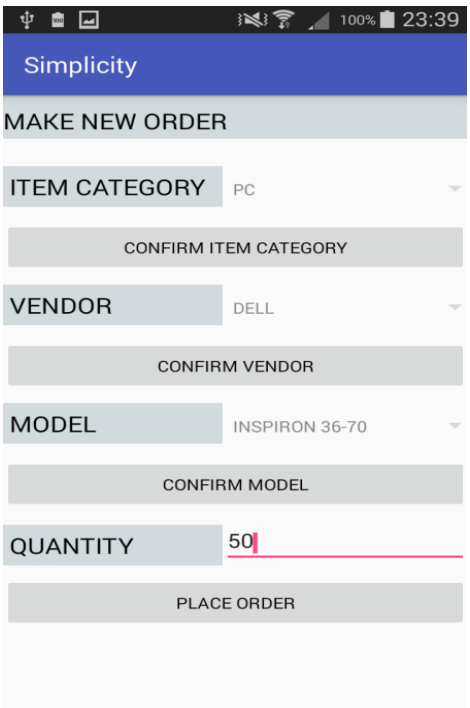
Εικόνα 4.21 Καταχώρηση order



Εικόνα 4.22 Καταχώρηση order 2



Εικόνα 4.23 Καταχώρηση order 3



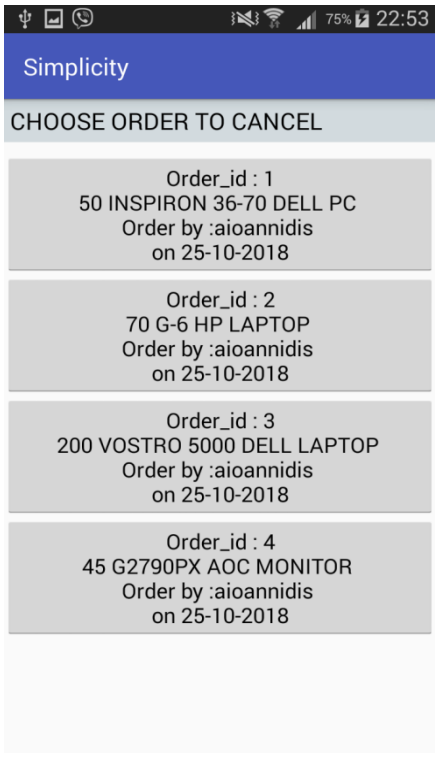
Εικόνα 4.24 Καταχώρηση order 4

ΒΗΜΑ 7 - ΕΛΕΓΧΟΣ ΑΝΟΙΚΤΩΝ ΠΑΡΑΓΓΕΛΙΩΝ ΚΑΙ ΑΚΥΡΩΣΗ ΠΑΡΑΓΓΕΛΙΑΣ

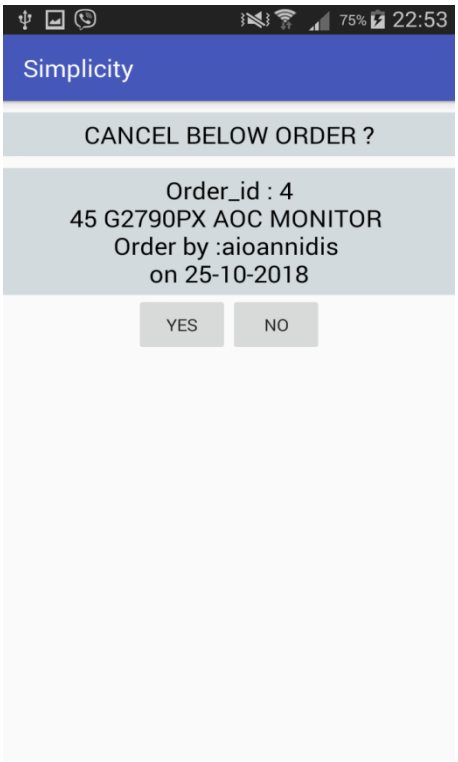
Έχοντας καταχωρήσει ο χρήστης τις παραγγελίες του προηγούμενου βήματος μπορεί πλέον να ελέγξει τις ανοιχτές παραγγελίες του(αυτές που δεν έχουν παραληγθεί ακόμα) ή και να ακυρώσει μια παραγγελία που έχει καταχωρηθεί.Από την κεντρική οθόνη της κατηγορίας orders and deliveries και πατώντας check pending orders ο χρήστης ελέγχει τις παραγγελίες που έχουν καταχωρηθεί στο σύστημα,δεν έχουν ακυρωθεί και δεν έχουν παραληφθεί (εικόνα 4.25).

Εικόνα 4.25 Pending Orders

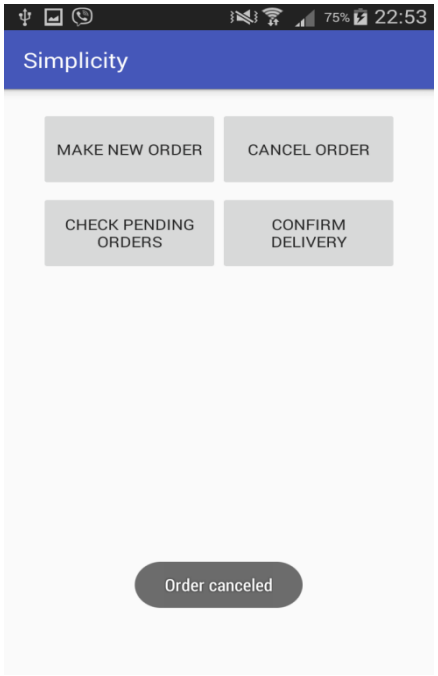
Στην συνέχεια και αφού ο χρήστης έχει καταχωρήσει και ελέγξει ότι υπάρχει στις ανοιχτές παραγγελίες μία παραγγελία που θέλει να ακυρώσει μεταβαίνει από την κεντρική οθόνη του μενού orders and deliveries στην οθόνη ακύρωσης πατώντας cancel order.Εκεί θα βρεί σε μορφή αντικειμένου button κάθε μία από τις ανοιχτές παραγγελίες του (εικόνα 4.26) και πατώντας το αντίστοιχο button θα βρεθεί στην οθόνη ακύρωσης της (4.27).Ο χρήστης εκεί επιλέγει την ακύρωση της τελευταίας παραγγελίας του (Order_id 4) και στην συνέχεια επιστρέφει στην οθόνη των ανοικτών παραγγελιών για να ελέγξει ότι έχει ενημερωθεί και ότι έχει αφαιρεθεί από το συγκεκριμένο report η ακυρωθείσα παραγγελία(4.28).



Εικόνα 4.26 Cancel Order



Εικόνα 4.27 Cancel Specific order

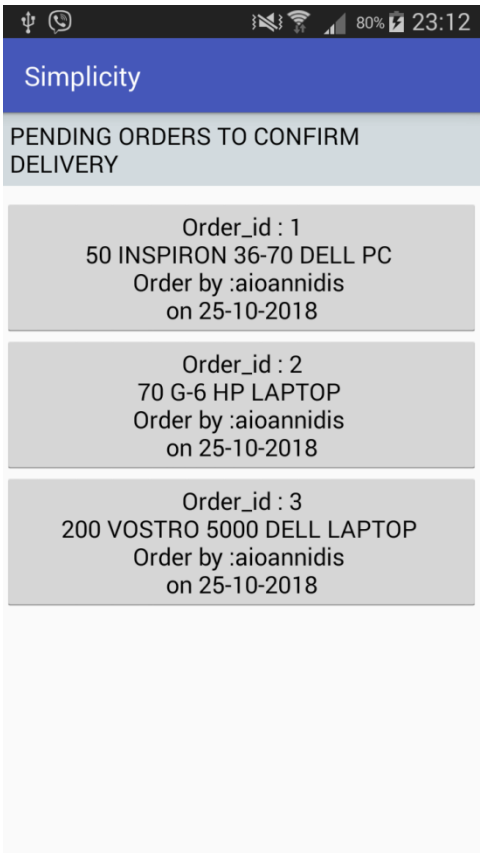


Εικόνα 4.28 Order Canceled

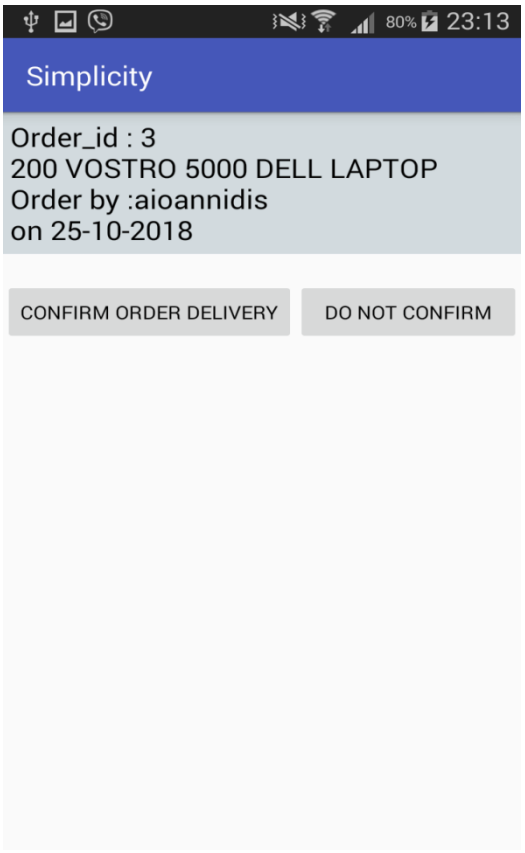
Εικόνα 4.29 Pending Orders

ΒΗΜΑ 8 - ΚΑΤΑΧΩΡΗΣΗ ΠΑΡΑΛΑΒΗΣ ΠΑΡΑΓΓΕΛΙΑΣ

Η καταχώρηση παραλαβής μιας παραγγελίας είναι η τελευταία ενέργεια που μπορεί να πραγματοποιήσει ο χρήστης στην κατηγορία orders and deliveries.Από το κεντρικό μενού της κατηγορίας και επιλέγοντας confirm delivery ο χρήστης βρίσκεται στην οθόνη επιβεβαίωσης παραλαβής παραγγελίας η οποία και του εμφανίζει σε μορφή buttons τις ανοιχτές του παραγγελίες(εικόνα 4.30).Ο χρήστης πατάει το button που αφορά την τελευταία καταχωρημένη ανοιχτή παραγγελία (order_id 3) και μεταβαίνει στην οθόνη επιβεβαίωσης παραλαβής της συγκεκριμένης παραγγελίας (εικόνα 4.31).Εκεί επιλέγει confirm order delivery (μπορεί επιλέγοντας do not confirm να μεταβεί στην αρχική οθόνη) και στην συνέχεια καλείται να επιβεβαιώσει την αποθήκη,τον διάδρομο και την θέση αποθήκευσης (εικόνες 4.30,4.31,4.32,4.33).



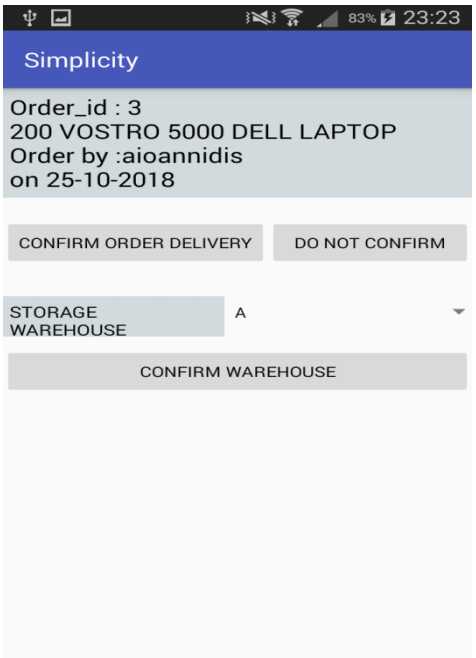
Εικόνα 4.30 Confirm Delivery



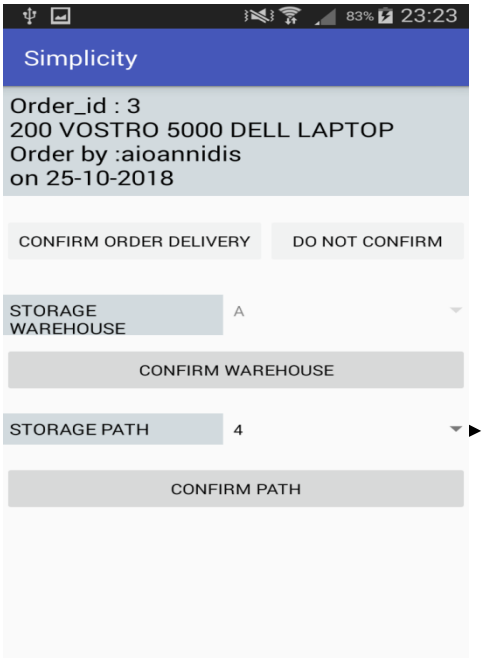
Εικόνα 4.31 Confirm Delivery 2

Αφού επιβεβαιώσει την θέση αποθήκευσης ο χρήστης καλείται να καταχωρήσει την ποσότητα των προϊόντων που θα αποθηκευθούν στην συγκεκριμένη θέση.Στο συγκεκριμένο σημείο εκτελούνται μια σειρά από ελέγχους στο σύστημα και στην συνέχεια ο χρήστης ενημερώνεται για την καταχώρηση της παραλαβής ή για τα πιθανά σφάλματα τα οποία προκύπτουν από τα δεδομένα καταχώρησης του.Οι έλεγχοι που πραγματοποιούνται στο σύστημα έχουν να κάνουν

με τον αριθμό των προϊόντων και την εγκυρότητα αυτού σε σχέση με τον αριθμό της παραγγελίας,τον έλεγχο επιτρεπόμενου βάρους αποθήκευσης στην θέση αποθήκευσης και τον έλεγχο διαθεσιμότητας από πλευράς όγκου αποθήκευσης.

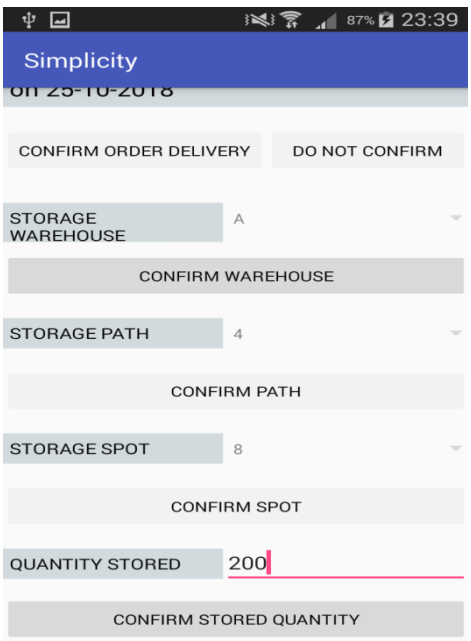


Εικόνα 4.32 Confirm Specific Delivery



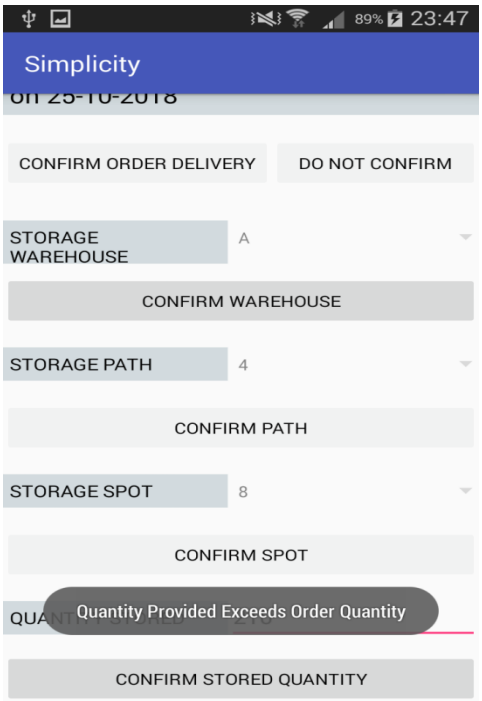
Εικόνα 4.33 Confirm Specific Delivery 2

Εικόνα 4.34 Confirm Delivery Spot

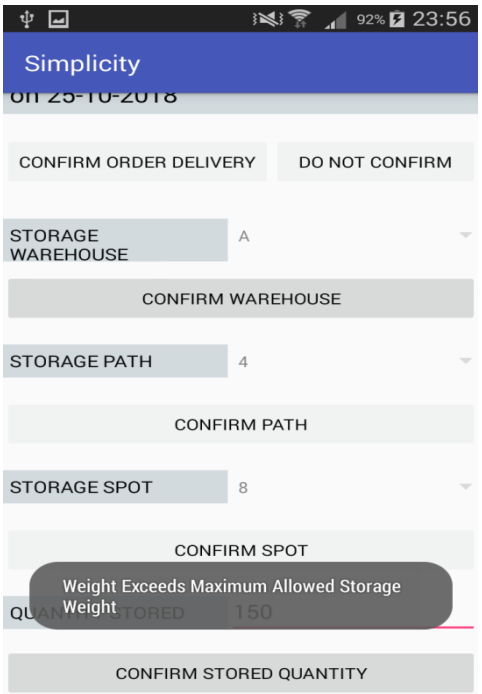


Εικόνα 4.35 Confirm Delivery Quantity

Ο χρήστης καταχωρεί αρχικά ποσότητα αποθήκευσης 210.Το σύστημα τον ενημερώνει με μήνυμα ότι ο αριθμός που καταχωρήθηκε είναι μεγαλύτερος από την ποσότητα της παραγγελίας. Στην συνέχεια ο χρήστης θα καταχωρήσει ποσότητα 0 για να ενημερωθεί ότι έχει δηλώσει μη έγκυρο αριθμό προϊόντων προς αποθήκευση. Έπειτα ο χρήστης καταχωρεί ποσότητα αποθήκευσης 150 για να ενημερωθεί από το σύστημα ότι το βάρος των 150 Laptops ξεπερνά το ανώτατο όριο βάρους της συγκεκριμένης θέσης αποθήκευσης.

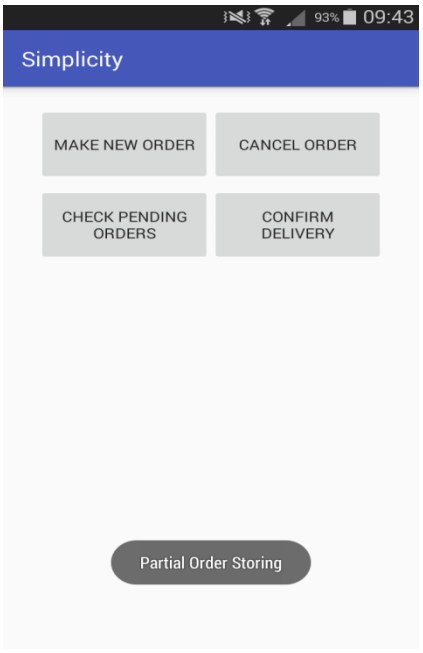


Εικόνα 4.36 Έλεγχος ποσότητας



Εικόνα 4.37 Έλεγχος Βάρους

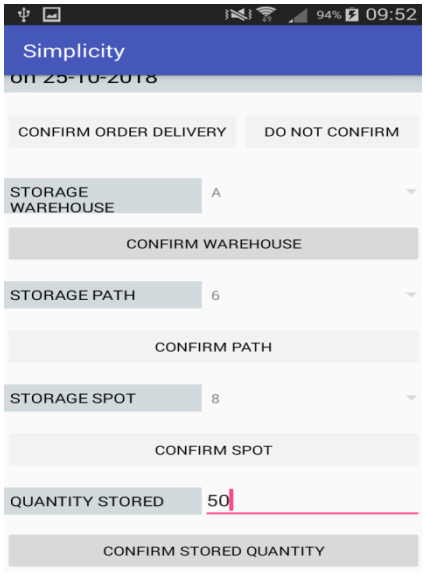
Τέλος ο χρήστης καταχωρεί αριθμό αποθήκευσης 50 laptop ο οποίος και είναι αποδεκτός από το σύστημα.Λόγω του ότι ο αριθμός των προϊόντων που αποθηκεύτηκε ήταν μικρότερος από τον αριθμό των προϊόντων της παραγγελίας ενημερώνεται με μήνυμα ότι αποθηκεύτηκε τμήμα της παραγγελίας ενώ αυτόματα το σύστημα στις ανοιχτές παραγγελίες έχει αφαιρέσει από την αναμενόμενη ποσότητα τα προϊόντα που αποθηκεύτηκαν (εικόνες.38, 4.39) .



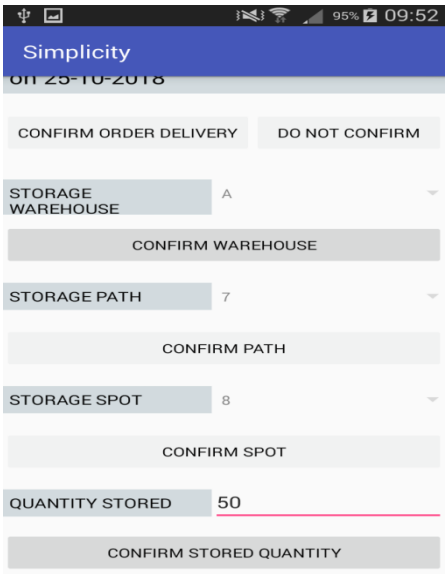
Εικόνα 4.38 Order Storing

Εικόνα 4.39 Pending Orders

Στην συνέχεια ο χρήστης επιστρέφοντας στην οθόνη confirm delivery συνεχίζει την καταχώρηση παραλαβής παραγγελίας για τα εναπομείναντα τεμάχια,τα οποία και αποθηκεύει αντίστοιχα στους διαδρόμους 5,6,7 στην θέση 8 κάθε διαδρόμου.



Εικόνα 4.40 Delivery Confirmation

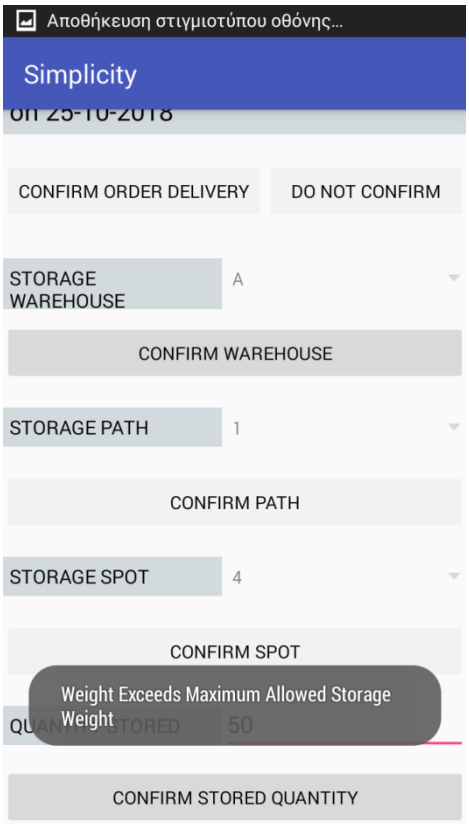


Εικόνα 4.41 Delivery Confirmation 2

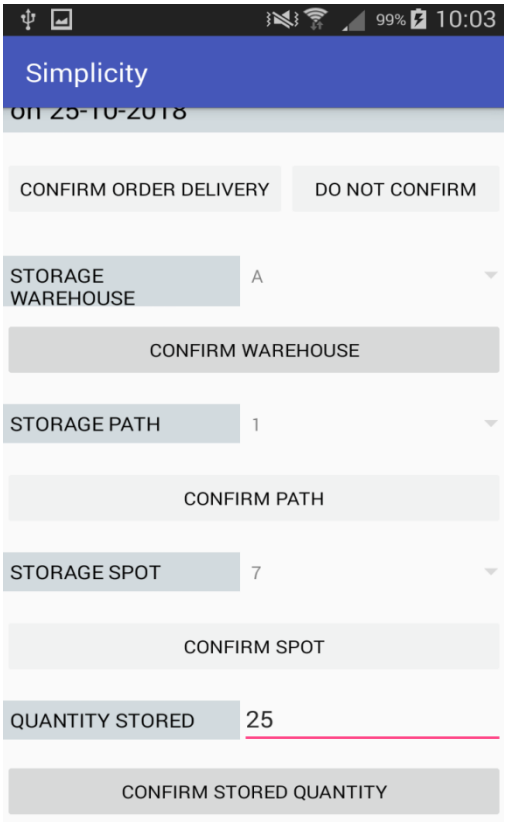
Μετά την καταχώρηση και του τελευταίου υπολοίπου ο χρήστης ενημερώνεται με μήνυμα ότι η συναλλαγή καταχωρήθηκε επιτυχώς.

Στην συνέχεια ο χρήστης προχωρά και στην επιβεβαιώση παραλαβής της πρώτης παραγγελίας των 50 PCs του κατασκευαστή DELL και μοντέλου INSPIRON 36-70.

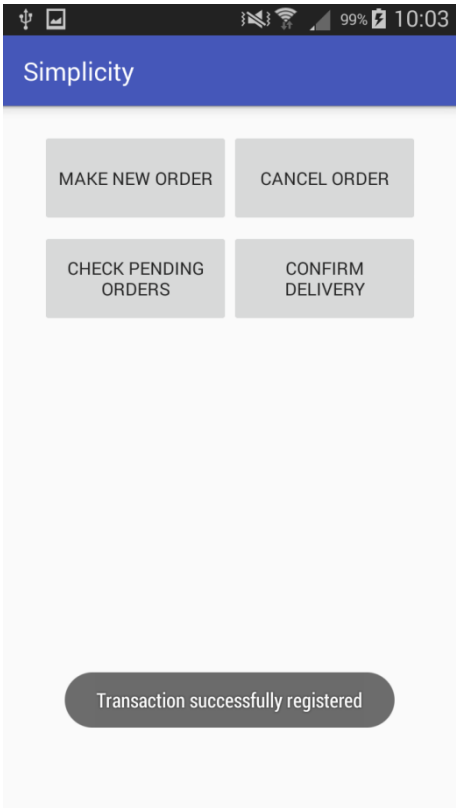
Έπειτα και από αυτήν την καταχώρηση ελέγχει ότι οι ανοιχτές παραγγελίες έχουν ενημερωθεί ανάλογα και επιστρέφει στην οθόνη με το κεντρικό μενού του χρήστη.



Εικόνα 4.42 Έλεγχος Βάρους



Εικόνα 4.43 Έλεγχος ποσότητας

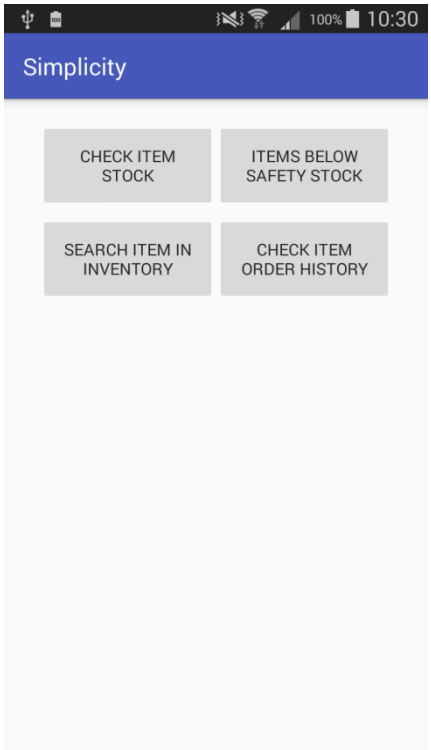


Εικόνα 4.44 Καταχώρηση transaction

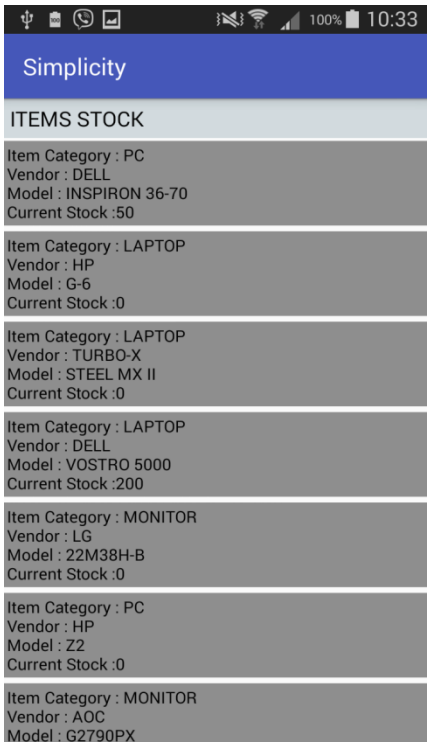
Εικόνα 4.45 Pending Orders

ΒΗΜΑ 9 - ΕΛΕΓΧΟΣ ΑΠΟΘΕΜΑΤΙΚΟΥ ΠΡΟΪΟΝΤΩΝ

Έχοντας ολοκληρώσει τον κύκλο από την καταχώρηση της παραγγελίας μέχρι και την καταχώρηση της παραλαβής της ο χρήστης μπορεί πλέον να μεταφερθεί για να ελέγξει τις δυνατότητες του μενού της κατηγορίας Inventory Management.Από την κεντρική οθόνη του χρήστη και επιλέγοντας inventory management ο χρήστης μεταφέρεται στην οθόνη της κατηγορίας (εικόνα 4.46).Οι επιλογές που του δίνονται εκεί είναι ο έλεγχος του αποθεματικού όλων των προϊόντων τα οποία είναι καταχωρημένα στο σύστημα,ο έλεγχος των προϊόντων που βρίσκονται κάτω από το όριο που έχει οριστεί ως αποθεματικό ασφαλείας κατά την καταχώρηση κάθε προϊόντος με παράλληλη δυνατότητα πραγματοποίησης της απαραίτητης παραγγελίας ώστε να καλυφθεί η διαφορά μέχρι το όριο ασφαλείας,ο έλεγχος των θέσεων αποθήκευσης όπου βρίσκεται συγκεκριμένο προϊόν καθώς επίσης και ο έλεγχος του ιστορικού παραγγελιών ενός συγκεκριμένου προϊόντος.



Εικόνα 4.46 Διαχείριση προϊόντος



Εικόνα 4.47 Αποθεματικό προϊόντος

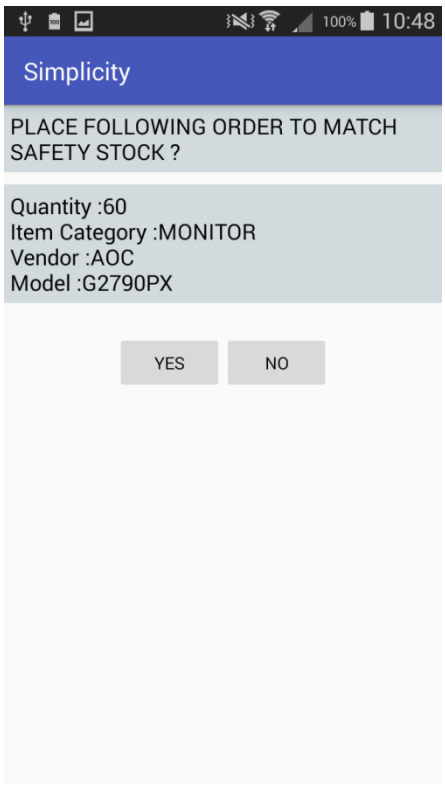
Ο χρήστης επιλέγει check item history και μεταβαίνει στην οθόνη όπου εμφανίζεται μία αναφορά με τα αποθεματικά όλων των διαθέσιμων προϊόντων (εικόνα 4.47).

Εκεί θα δει αναλυτικά για κάθε προϊόν την κατηγορία του,τον κατασκευαστή του,το μοντέλο του και το διαθέσιμο αποθεματικό.Στην λίστα θα εμφανιστούνόλα τα καταχωρημένα προϊόντα,ακόμα και εκείνα για τα οποία δεν έχει καταχωρηθεί ποτέ παραλαβή κάποιας παραγγελίας (αυτά θα εμφανιστούν με αποθεματικό 0).

ΒΗΜΑ 10 - ΕΛΕΓΧΟΣ ΠΡΟΪΟΝΤΩΝ ΜΕ ΑΠΟΘΕΜΑΤΙΚΟ ΚΑΤΩ ΤΟΥ ΟΡΙΟΥ ΑΣΦΑΛΕΙΑΣ

Έπειτα επιστρέφοντας στην κεντρική οθόμη της κατηγορίας ο χρήστης επιλέγει items below safety stock και μεταβαίνει στην αντίστοιχη οθόνη.Εκεί μπορεί να δει όλα τα προϊόντα των οποίων το αποθεματικό βρίσκεται κάτω από το όριο ασφαλείας σε μορφή button objects το καθένα εκ των οποίων φέρει σαν κείμενο την περιγραφή του προϊόντος,το διαθέσιμο αποθεματικό και το όριο αποθεματικού ασφαλείας (εικόνα 4.48).Πα΄τωντας πάνω στο αντίστοιχο button μεταβαίνει σε νέα οθόνη όπου του δίνεται η επιλογή καταχήρησης παραγγελίας της διαφοράς των προϊόντων μέχρι το απαραίτητο όριο (εικόνα 4.49).

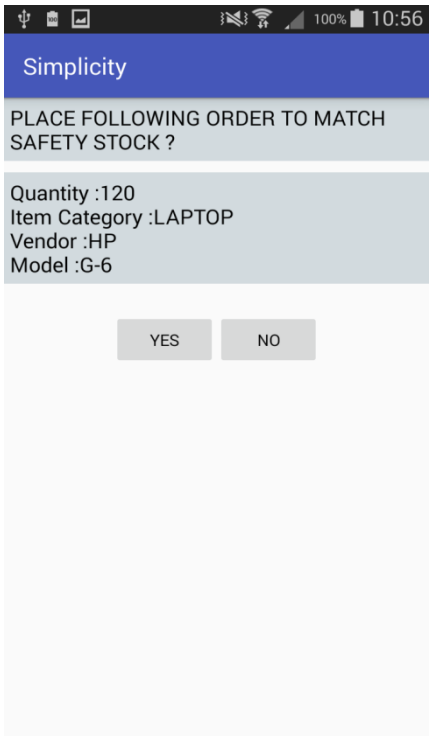
Εικόνα 4.48 Item Stock



Εικόνα 4.49 Safety Stock order

Ο χρήστης καταχωρεί τις παραγγελίες ασφαλείας για τα προϊόντα AOC Monitor G2790PX και HP Laptop G-6. Στην συνεχεια μεταβαίνει στην αναφορά ανοιχτών παραγγελιών μέσω της κατηγορίας orders and deliveries και του button check pending orders όπου επιβεβαιώνει ότι οι παραγγελίες έχουν καταχωρηθεί στο σύστημα (εικόναι 4.51).

Έτσι,και μέσω αυτής της λειτουργίας μπορεί ο χρήστης πολύ εύκολα και αυτοματοποιημένα να ελέγξει ανά πάσα στιγμή ότι δεν βρίσκεται κάτω από το ασφαλές αποθεματικό όριο κάθε για κανένα προϊόν το οποίο φιλοξενεί στην αποθήκη του.



Εικόνα 4.50 Safety Order 2

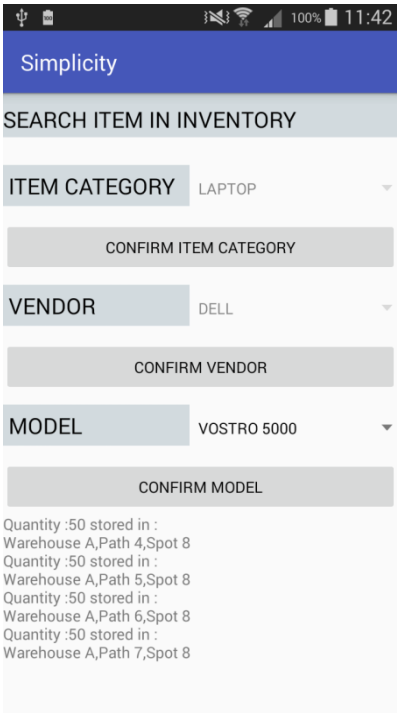
Εικόνα 4.51 Pending Orders

ΒΗΜΑ 10 - ΕΛΕΓΧΟΣ ΑΠΟΘΕΜΑΤΙΚΟΥ ΚΑΙ ΣΗΜΕΙΟΥ ΑΠΟΘΗΚΕΥΣΗΣ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΠΡΟΪΟΝΤΟΣ

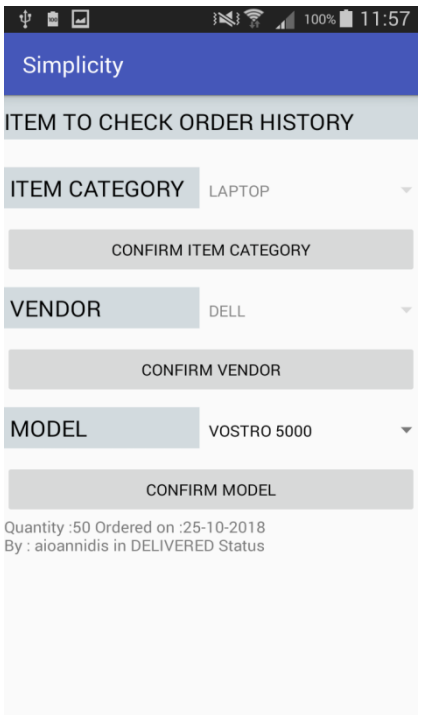
Συνεχίζοντας τον έλεγχο των δυνατοτήτων του συστήματος ο χρήστης από το κεντρικό μενού της κατηγορίας inventory management επιλέγει search item in inventory.

Εκεί καλείται να επιβεβαιώσει σε τρία στάδια (επιβεβαίωση κατηγορίας προϊόντος,επιβεβαίωση κατασκευαστή προϊόντος και επιβεβαίωση μοντέλου) το προϊόν προς αναζήτηση.Αφού παράσχει τις απαραίτητες πληροφορίες το σύστημα τον ενημερώνει για το σύνολο των αποθηκευτικών θέσεων που υπάρχει το προϊόν καθώς και τον αριθμό των τεμαχίων ανά θέση.Ο χρήστης δίνει ως παραμέτρους στο σύστημα την κατηγορία LAPTOP,τον κατασκευαστή DELL και το μοντέλο Vostro 5000.Με την επιβεβαίωση του μοντέλου προς αναζήτηση τα αποτελέσματα εμφανίζονται στην οθόνη (εικόνα 4.52).Έτσι,το σύστημα μπορεί να παρέχει στον χρήστη πληροφορίες σε πραγματικό χρόνο για την τοποθεσία του προϊόντος που θέλει και τον ακριβή αριθμό των τεμαχίων ανά θέση σε όλους τους διαθέσιμους αποθηκευτικούς του χώρους.

ΒΗΜΑ 11 - ΕΛΕΓΧΟΣ ΙΣΤΟΡΙΚΟΥ ΠΑΡΑΓΓΕΛΙΩΝ ΣΥΓΚΕΚΡΙΜΕΝΟΥ ΠΡΟΪΟΝΤΟΣ



Εικόνα 4.52 Specific Item Stock



Εικόνα 4.53 Specific Item Stock 2

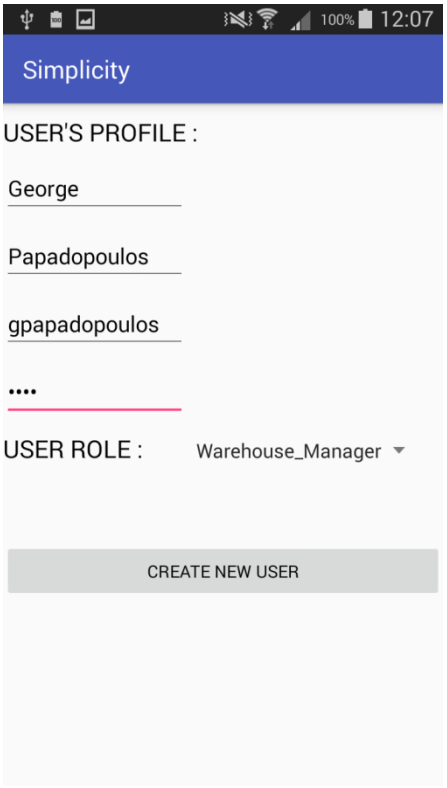
Τέλος,μετά τον έλεγχο αναζήτησης σε συγκεκριμένο προϊόν το σύστημα δίνει στον χρήστη την δυνατότητα ενημέρωσης σχετικά με το ιστορικό των παραγγελιών ενός προϊόντος.Για να λάβει την αναφορά αυτή,ο χρήστης μεταβαίνει στο κεντρικό μενού της κατηγορίας inventory

management και επιλέγει check item order history.Από εκεί οδηγείται στην αντίστοιχη οθόνη όπου επιβεβαιώνοντας και πάλι σε τρία στάδια το προϊόν του ενδιαφέροντός του λαμβάνει στην οθόνη τα αποτελέσματα.Εκεί μπορεί να δει τον αριθμό των τεμαχίων που παραγγέλθηκαν, τον χρήστη που καταχώρησε την παραγγελία,την ημερομηνία αυτής καθώς επίσης και το status της παραγγελίας (delivered,pending,canceled) (εικόνα 4.53).

ΒΗΜΑ 12 - ΚΑΤΗΓΟΡΙΑ ΕΠΙΚΟΙΝΩΝΙΑΣ ΧΡΗΣΤΩΝ

Για τον έλεγχο της τελευταίας κατηγορίας του κεντρικού menu my tasks and messages θα χρειαστεί η δημιουργία στο σύστημα ενός δεύτερου χρήστη με τον οποίο ο ήδη υπάρχον θα έχει επικοινωνία.

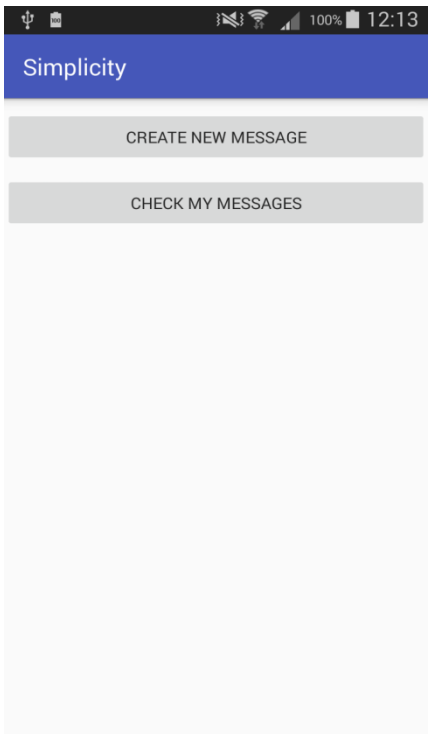
Προς αυτό,πραγματοποιείται σύδεση στο σύστημα με τον χρήστη administrator ο οποίος και δημιουργεί λογαριασμό για τον χρήστη gpapadopoulos (εικόνα 4.54) .



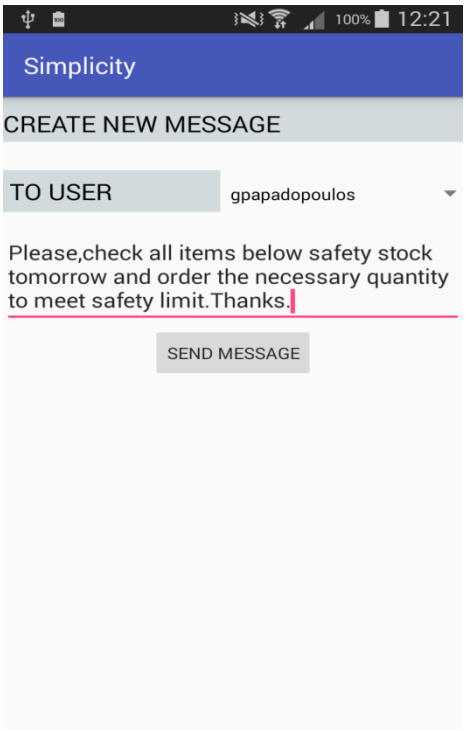
Εικόνα 4.54 New User

Στην συνέχεια ο χρήστης συνδέεται στο σύστημα ως aioannidis και μεταβαίνει στην κατηγορία my tasks and messages.Εκεί αφού επιλέγει create new message και μεταβαίνει στην οθόνη δημιουργίας του νέου μηνύματος.Επιλέγει τον χρήστη που θέλει να λάβει το μήνυμα (gpapadopoulos) από το drop down menu ενός spinner object (δεν εμφανίζονται οι χρήστες που

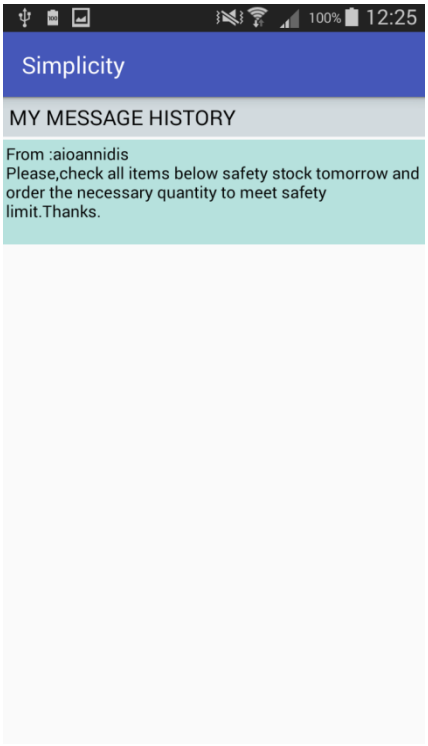
έχουν ρόλο administrator) και πληκτρολογεί το μήνυμα προς αποστολή.Τέλος,πατάει στο button send message και ενημερλωνεται μήνυμα για την επιτυχή αποστολή ενώ παράλληλα το σύστημα τον μεταφέρει στην προηγούμενη οθόνη.Πλέον,ο χρήστης αποσυνδέεται από τον λογαριασμό aioannidis,συνδέεται με τον χρήστη gpapadopoulos και μεταβαίνει στην κατηγορία my tasks and messages όπου ελέγχει τα μηνύματα του (check my messages) για ναδει το μήνυμα από τον χρήστη aioannidis (εικόνα 4.57).Αφού βλέπει το μήνυμα επιστρέφει στην προηγούμενη οθόνη όπου μέσω της επιλογής create new message στέλνει απαντητικό μήνυμα στον χρήστη aioannidis.Τέλος,ο χρήστης αποσυνδέεται και συνδέεται ξανά ως aioannidis όπου ελέγχοντας τα μηνύματα του μπορεί να δει την συνομιλία με τον gpapadopoulos (εικόνα 4.58).



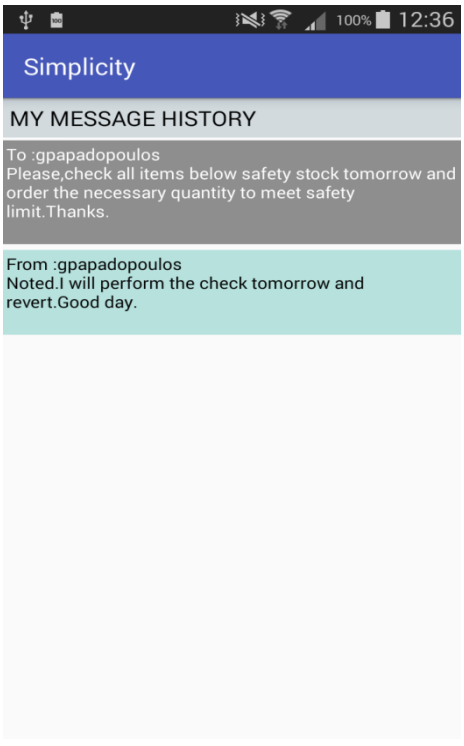
Εικόνα 4.55 Message Μενού



Εικόνα 4.56 New Message

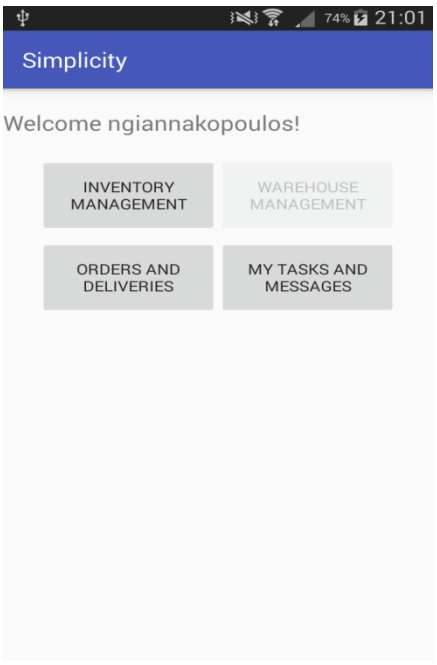


Εικόνα 4.57 Message Box

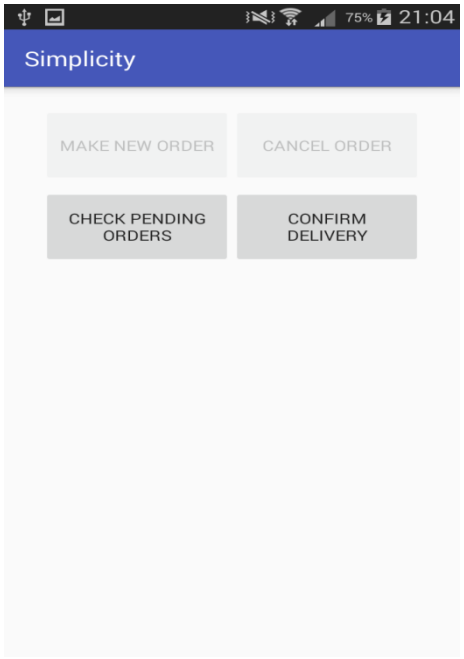


Εικόνα 4.58 Message Box 2

ΒΗΜΑ 13 - ΕΛΕΓΧΟΣ ΠΕΡΙΟΡΙΣΜΟΥ ΔΥΝΑΤΟΤΗΤΩΝ ΑΠΛΟΥ ΧΡΗΣΤΗ



Εικόνα 4.59 User μενού



Εικόνα 4.60 User orders μενού

Περιορισμένες δυνατότητες απλού χρήστη κατά την σύνδεση του στο κεντρικό μενού και στην κεντρική οθόνη διαχείρισης παραγγελιών.

4.2 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό πραγματοποιήθηκε μέσα από ένα δεδομένο σενάριο ελέγχου μια παρουσίαση των δυνατοτήτων του συστήματος και των επιλογών που προσφέρονται στους χρήστες του. Συγκεκριμένα, ξεκινώντας από ένα άδειο σύστημα με μόνη εγγραφή στην βάση δεδομένων τον αρχικό χρήστη administrator δημιουργήθηκε ο πρώτος χρήστης του συστήματος με username aioannidis και ρόλο χρήστη warehouse manager.

Ο εν λόγω χρήστης και μετά την σύνδεση του στο σύστημα ελέγχει τις δυνατότητες της κατηγοριας warehouse management όπου και καταχωρεί στο σύστημα μία νέα αποθήκη δίνοντας τιμές στα μεγέθη που θα την χαρακτηρίζουν (όνομα, διάδρομοι αποθήκευσης,αποθηκευτικές θέσεις ανά διάδρομο, μέγιστο βάρος αποθήκευσης ανά θέση, διαστάσεις θέσης αποθήκευσης),καταχωρεί νέες κατηγορίες προϊόντων στο σύστημα και τέλος νέα προϊόντα δίνοντας τιμές επίσης στα μεγέθη που θα τα χαρακτηρίζουν (κατηγορία προϊόντος, κατασκευαστής, μοντέλο, βάρος, διαστάσεις).

Έπειτα, έχοντας τροφοδοτήσει το σύστημα με χώρους αποθήκευσης και προϊόντα προς διαχείριση, ο χρήστης ελέγχει τις δυνατότητες της κατηγορίας orders and deliveries.Στην διαδικασία αυτή ο χρήστης καταχωρεί νέες παραγγελίες προϊόντων που έχει προηγουμένως καταχωρήσει στο σύστημα, ελέγχει τις ανοιχτές παραγγελίες του και ακυρώνει κάποιες από αυτές. Τέλος, επιβεβαιώνει την παραλαβή κάποιων εκ των ανοιχτών παραγγελιών και καταχωρεί στο σύστημα τις θέσεις αποθήκευσης του κάθε προϊόντος.

Στην συνέχεια, μεταβαίνει στην κατηγορία inventrory management όπου ελέγχει το αποθεματικό διαθέσιμο όλων των προϊόντων του, ελέγχει μέσα από μια οθόνη αναφοράς τα προϊόντα που βρίσκονται σε αποθεματικό κάτω του ορίου ασφαλείας που εχει οριστεί κατά την καταχώρηση του προϊόντος και παράλληλα μέσα από δυνατότητα επιλογής που του δίνει το σύστημα καταχωρεί τις απαραίτητες παραγγελίες ώστε να φτάσει στο επιθυμητό όριο ασφαλείας.

Τέλος, κλείνοντας τον κύκλο αξιολόγησης του συστήματος ελέγχει τις δυνατότητες της κατηγορίας my tasks and messages όπου και έχει επικοινωνία μέσω μηνύματος με άλλον χρήστη του συστήματος στον οποίο αναθέτει μία εργασία για την επόμενη μέρα.

ΚΕΦΑΛΑΙΟ 5 ΤΕΧΝΙΚΑ ΖΗΤΗΜΑΤΑ ΑΝΑΠΤΥΞΗΣ

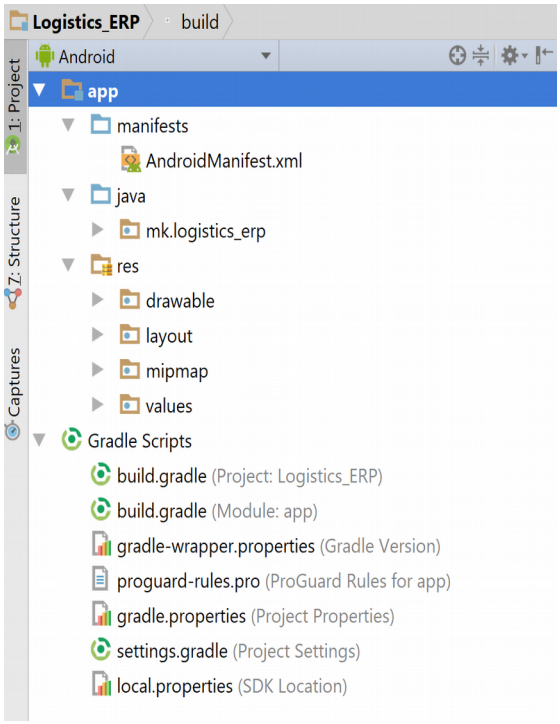
5.1 Τεχνικά ζητήματα ανάπτυξης κώδικα εφαρμογής

ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ ANDROID STUDIO

Όπως αναφέρθηκε στο κεφάλαιο της ανάλυσης των απαιτήσεων και του σχεδιασμού του συστήματος η εφαρμογή που αποτελεί το προϊόν της παρούσας διπλωματικής εργασίας είναι μία εφαρμογή σχεδιασμένη για κινητές συσκευές με λειτουργικό σύστημα Android OS.Για την ανάπτυξη μιας τέτοιας εφαρμογής υπάρχουν αρκετά εργαλεία τα οποία υποστηρίζουν είτε εκ φύσεως(Android Studio) είτε μέσω κάποιου patch προγράμματος την ανάπτυξη Android εφαρμογής (Netbeans, Eclipse,IntelliJ,Visual Studio - Xamarin).Με δεδομένο ότι η μητέρα εταιρεία του λειτουργικού συστήματος Android, δηλαδή η Google, διαθέτει ελεύθερο προς χρήση δικό της περιβάλλον ανάπτυξης Android εφαρμογών, το Android Studio, επιλέχθηκε αυτό ως IDE (Integrated Development Environment) λόγω της βελτιστοποιησης του για την συγκεκριμένη διαδικασία,το γεγονός ότι αποτελεί την επίσημη πλατφόρμα ανάπτυξης και λόγω των χαρακτηριστικών ανάπτυξης που διαθέτει. Στα χαρακτηριστικά αυτά ανήκει ο σχεδιαστής user interface με visual χαρακτηριστικά, δηλαδή χρήση έτοιμων και παραμετροποιήσιμων αντικειμένων τα οποία μπορούν να τοποθετηθούν σε ένα υπάρχον interface και το περιβάλλον να δημιουργήσει αυτόματα τον κώδικα (στην παρούσα εφαρμογή ωστόσο προτιμήθηκε η δημιουργία του user interface να γίνει μέσω κώδικα μόνο λόγω ανάγκης για αυστηρότερο αποτέλεσμα σε αποστάσεις και ομοιομορφία),ο ενσωματωμένος αναλυτής APK, μέσω του οποίου μπορεί να γίνει έλεγχος του τελικού Android Package Kit και βελτιστοποίηση του μέσω του ελέγχου του μεγέθους των βιβλιοθηκών και των resource αρχείων που συνθέτουν την εφαρμογή, ο πολύ αποτελεσματικός code editor που διαθέτει με χαρακτηριστικά όπως η αυτόματη συμπλήρωση κειμένου να λειτουργούν σε πάρα πολύ ανεπτυγμένο επίπεδο και τα συστήματα παρακολούθησης της εκτέλεσης της εφαρμογής σε πραγματικό χρόνο. Το Android Studio υποστηρίζει SDK (Software Development Kit) και NDK (Native Development Kit) για ανάπτυξη της εφαρμογής σε Java με ενσωματωμένο κώδικα από γλώσσες C και C++. Τέλος, διαθέτει καταγραφή πραγματικού χρόνου για κατανάλωση πόρων μνήμης, δικτύου κλπ.

ΔΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ANDROID STUDIO PROJECT

Τα δομικά στοιχεία ενός Android Studio Project γίνονται διακριτά με την εκκίνηση του και το άνοιγμα ενός νέου ή υπάρχοντος project.Στην κεντρική οθόνη εργασίας και στην αριστερή πλευρά δίνεται μία λίστα επιλογής σε μορφή drop down μενού για την απεικόνιση των βασικών αρχείων. Η προεπιλεγμένη απεικόνιση είναι η Android.Εκεί διακρίνουμε τις τέσσερις βασικές κατηγορίες αρχείων που συνθέτουν την εφαρμογή. Αυτές είναι τα *αρχεία manifest*,τα *αρχεία Java*,τα *αρχεία resources* και τα *αρχεία Gradle Scripts*.



Εικόνα 5.1 Android Studio Project

ΑΡΧΕΙΟ MANIFEST ΚΑΙ ΚΩΔΙΚΑΣ ΑΡΧΕΙΟΥ

Στα αρχεία manifest δημιουργείται το AndroidManifest.xml αρχείο κατά την δημιουργία του app project.Το αρχείο αυτό περιγράφει πολύ ουσιώδεις πληροφορίες σχετικά με την εφαρμογή στα εργαλεία ανάπτυξης του Android Studio,στο λειτουργικό σύστημα Android και στο Google Play (την υπηρεσία διανομής ψηφιακού υλικού και εφαρμογών της Google).Μεταξύ άλλων το αρχείο manifest χρειάζεται για να δηλώσει τα ακόλουθα :

1. Το όνομα πακέτου της εφαρμογής (package name) το οποίο συνήθως ταυτίζεται με την ιδιότητα namespace του κώδικα. Τα εργαλεία ανάπτυξης Android το χρησιμοποιούν αυτό για να καθορίσουν την τοποθεσία των οντοτήτων του κώδικα όταν συνθέτουν την εφαρμογή. Όταν συνθέτουν το τελικό πακέτο της εφαρμογής (apk) τα εργαλεία αυτά αντικαθιστούν αυτήν την τιμή με το Application ID των Gradle αρχείων, το οποίο και χρησιμοποιείται σαν το μοναδικό αναγνωριστικό κλειδί κάθε εφαρμογής τόσο στο σύστημα ανάπτυξης όσο και στο Google Play.

2. Τα συνθετικά συστατικά (components) της εφαρμογής τα οποία περιλαμβάνουν τις κατηγορίες των activities,των services,των broadcast receivers και τους παροχείς περιεχομένου. Κάθε συστατικό πρέπει να ορίζει κάποιες βασικές ιδιότητες όπως το της Java κλάσης του.
3. Τα δικαιώματα (permissions) που χρειάζεται η εφαρμογή για να έχει πρόσβαση σε προστατευμένα τμήματα του συστήματος ή άλλων εφαρμογών (π.χ. internet access,πρόσβαση στις αποθηκευμένες επαφές, στα μηνύματα κλπ)
4. Τα χαρακτηριστικά υλικού και λογισμικού που απαιτεί η εφαρμογή έτσι ώστε να μπορεί να εγκατασταθεί σε μία συσκευή, το οποίο επηρεάζει το ποιες συσκευές μπορούν να εγκαταστήσουν την εφαρμογή μέσω του Google Play.

Ο κώδικας του αρχείου AndroidManifest.xml της εφαρμογής :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"


  package="mk.logistics_erp">
  <uses-permission android:name="android.permission.INTERNET"/>
  <application
    android:allowBackup="true"
    android:icon="@drawable/erp_1"
    android:label="Simplicity"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".log_in">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
    <activity android:name=".user_home_screen">
```

```
</activity>
<activity android:name=".admin_home_screen">
</activity>
<activity android:name=".create_new_user">
</activity>
<activity android:name=".add_warehouse">
</activity>
<activity android:name=".warehouse_management">
</activity>
<activity android:name=".edit_warehouse">
</activity>
<activity android:name=".add_new_item_category">
</activity>
<activity android:name=".add_new_item">
</activity>
<activity android:name=".orders_and_deliveries_home">
</activity>
<activity android:name=".make_new_order">
</activity>
<activity android:name=".cancel_order">
</activity>
<activity android:name=".cancel_selected_order">
</activity>
<activity android:name=".check_pending_orders">
</activity>
<activity android:name=".confirm_delivery">
</activity>
<activity android:name=".confirm_selected_order_delivery">
</activity>
<activity android:name=".inventory_management_home">
</activity>
```

```
        <activity android:name=".check_item_stock">
        </activity>
        <activity android:name=".items_below_safety_stock">
        </activity>
        <activity android:name=".place_safety_order">
        </activity>
        <activity android:name=".search_item_in_inventory">
        </activity>
        <activity android:name=".check_item_order_history">
        </activity>
        <activity android:name=".tasks_and_messages">
        </activity>
        <activity android:name=".create_new_message">
        </activity>
        <activity android:name=".check_my_messages">
        </activity>
    </application>
</manifest>
```

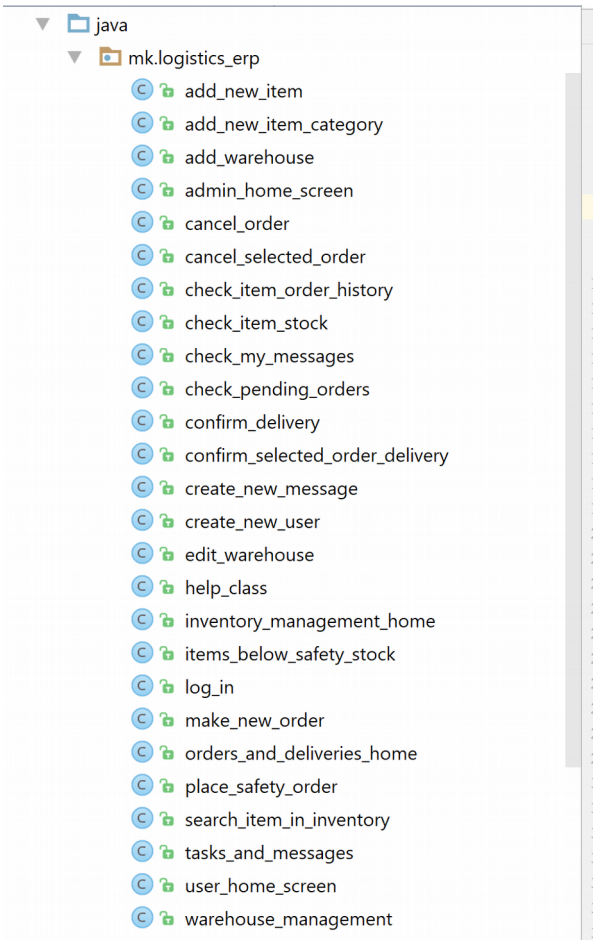
ΤΑ ΑΡΧΕΙΑ JAVA ΚΑΙ RESOURCES ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Το δεύτερο βασικό συστατικό της εφαρμογής και το ουσιαστικότερο καθώς περιέχει όλη την πληροφορία για την λογική της εφαρμογής και την επεξεργασία της πληροφορίας είναι τα αρχεία Java.Κάθε οθόνη που μπορεί να εμφανιστεί στην εφαρμογή αποτελεί ένα ξεχωριστό συστατικό της που ονομάζεται activity.Κάθε activity αποτελείται από την σύνθεση του αρχείου κώδικα της Java και των αντίστοιχων αρχείων resources που περιγράφουν το user interface του activity.Τα αρχεία resources που φέρουν τον xml κώδικα για την σύνθεση του user interface ενός activity ονομάζονται layouts και είναι αρχεία τύπου xml. Για κάθε ένα αρχείο layout υπάρχει και το αντίστοιχο .class αρχείο που αντιστοιχεί στον back_end Java κώδικα του. Εκτός των αρχείων (κλάσεων Java) που αντιστοιχούν σε layout xml αρχεία μπορεί να υπάρχουν και άλλα αρχεία κλάσεων που χρησιμοποιούνται στο πρόγραμμα για διαφορετικούς σκοπούς από την διαχείριση του περιεχομένου μιας οθόνης.

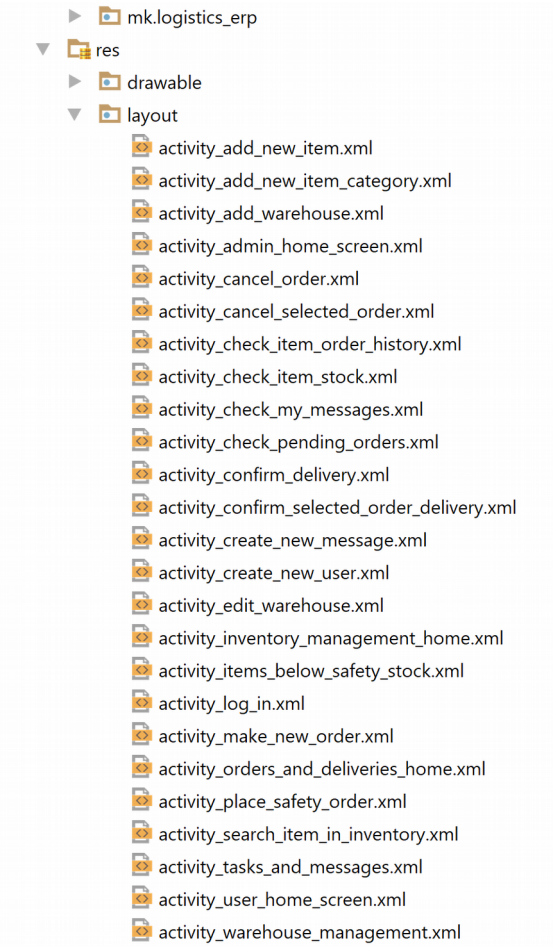
Τα αρχεία resources αποτελούνται από τις κατηγορίες layout (αναφέρθηκαν προηγουμένως),drawable (γραφικά αρχεία resources που μπορούν να εμφανιστούν σε κάποια οθόνη απεικόνισης είτε μέσω εντολών Java είτε μέσω κάποιας ιδιότητας xml στα χαρακτηριστικά

ενός User Interface στοιχείου), mipmap (αρχεία εικόνων που χρησιμοποιούνται για την εικόνα εκκίνησης της εφαρμογής) και values (αρχεία xml που περιέχουν resources κειμένου, χρωμάτων, στυλ και τα οποία μπορούν να καλούνται στο πρόγραμμα με το όνομα τους και να αλλάζουν τιμές κεντρικά από ένα xml αρχείο).

Στις παρακάτω εικόνες φαίνεται η αντιστοίχιση των αρχείων user interface resources (.xml) με τα αρχεία του back_end τμήματος της εφαρμογής. Για κάθε ένα activity που υπάρχει στο πρόγραμμα έχει δημιουργηθεί και το αντίστοιχο .class αρχείο.



Εικόνα 5.2 Java κλάσεις



Εικόνα 5.3 Layout αρχεία

ΤΑ ΑΡΧΕΙΑ GRADLE ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Τα Gradle Scripts της εφαρμογής περιέχουν όλες τος απαραίτητες οδηγίες για την διαδικασία δημιουργίας ενός τελικού παραδοτέου αρχείου εγκατάστασης (από την παραγωγή των τελικών αρχείων κώδικα, στην διαδικασία του compile,την διενέργεια δοκιμαστικής εκτέλεσης, της ενσωμάτωσης της εφαρμογής στα απαραίτητα αρχεία κλπ.).Βασικές πληροφορίες που αναφέρονται μπορεί να είναι οι βιβλιοθήκες που ενσωματώνονται στην εφαρμογή, τα χαμηλότερα επίπεδα API που μπορεί να λειτουργήσει η εφαρμογή (παλαιότερη υποστηριζόμενη έκδοση του Android OS) κ.α.

ΔΟΜΗ USER INTERFACE ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ LAYOUT

Αναφέρθηκε προηγουμένως ότι η δημιουργία ενός Activity συνιστάται στην δημιουργία ενός resource layout αρχείου και του αντίστοιχου .class Java αρχείου.Το layout αρχείο όπως επισημάνθηκε προηγουμένως είναι ένα αρχείο .xml στο οποίο καταχωρούνται τα απαραίτητα δομικά συστατικά ενός user interface.Τα αντικείμενα τα οποία μπορούν να εμφανιστούν στην οθόνη της κινητής συσκευής είναι όλα αντικείμενα της κλάσεως View.Τα βασικότερα views που χρησιμοποιήθηκαν για την παρούσα εργασία είναι αντικείμενα τύπου buttons. Δηλαδή, κουμπιά τα οποία ο χρήστης μπορεί να πατήσει και να έχει αλληλεπίδραση με την εφαρμογή, μπορεί να είναι αντικείμενα τύπου textview τα οποία εμφανίζουν ένα κείμενο στην οθόνη με τον τρόπο που περιγράφεται στις ιδιότητες του xml στοιχείου που συνθέτει το tewtview, αντικείμενα τύπου edit text τα οποία είναι υποδοχείς κειμένου και στα οποία ο χρήστης μπορεί να πληκτρολογήσει και να εισάγει πληροφορία για να αλληλεπιδράσει με το σύστημα και αντικείμενα τύπου spinners τα οποία χρησιμοποιήθηκαν στη δημιουργία της εφαρμογής για να αναπαραχθούν drop-down λίστες επιλογής.

Άρα, κάθε ένα αντικείμενο που εμφανίζεται στην οθόνη αποτελεί ένα αντικείμενο της κλάσεως View και περιγράφεται με ένα xml στοιχείο στο layout αρχείο του activity που εμφανίζεται. Στο στοιχείο αυτό δίνονται τιμές στις απαραίτητες ιδιότητες και σε όσες επιπλέον επιθυμεί ο χρήστης ώστε να περιγραφεί με τον κατάλληλο τρόπο. Οι βασικές ιδιότητες που πρέπει να έχει ένα View είναι η android:layout_width και η android:layout_height και οι τιμές που μπορούν να πάρουν είναι είτε τιμές συγκεκριμένου μεγέθους (π.χ. 20dp) είτε η τιμή wrap_content ,που σημαίνει ότι θα καταλάβει τόσο τμήμα της διάστασης όσο του χρειάζεται για να περικλύσει το περιεχόμενο του, είτε η match_parent που σημαίνει ότι θα καταλάβει όλη τη διαθέσιμη διάσταση του πατρικού του (parent) view. Άλλες ιδιότητες που συνθέτουν ένα στοιχείο ui είναι η ιδιότητα id,η οποία δίνει στο στοιχείο ένα μοναδικό αναγνωριστικό όνομα έτσι ώστε να μπορεί να κληθεί από τον κώδικα της Java για να εκτελέσει την όποια δοθείσα εντολή, οι ιδιότητες text,textsize,textcolor που αφορούν το κείμενο το οποίο θα αναγράφεται στο αντίστοιχο view,οι ιδιότητες margin και padding που αφορούν την απόσταση από άλλα views στην οθόνη, η ιδιότητα background που δείχνει το χρώμα του φόντου του αντικειμένου κ.α Συγκεκριμένα για τα button views χρησιμοποιείται η ιδιότητα onClick η οποία περιγράφει την μέθοδο που θα κληθεί από τον κώδικα της Java που αντιστοιχεί στο συγκεκριμένο layout σε περίπτωση που ο χρήστης πατήσει την οθόνη στα πλαίσια που καλύπτει με τις διαστάσεις του το button.

Ακολουθεί ενδεικτικά ο κώδικας που περιγράφει ένα view κάθε κατηγορίας :

Το textview που εμφανίζει το κείμενο "USER'S PROFILE" στην οθόνη δημιουργίας νέου χρήστη.

```
<TextView
    android:id="@+id/create_new_user_textview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="USER'S PROFILE :"
    android:textSize="20dp"
    android:textColor="@color/black">
</TextView>
```

Το edittext που υποδέχεται την πληκτρολόγηση του ονόματος χρήστη κατά την διαδικασία σύνδεσης στο σύστημα. Η ιδιότητα hint έχει τιμή το κείμενο που θα εμφανίζεται σαν βοήθεια προς τον χρήστη για να κατανοήσει την χρήση του αντίστοιχου view.

```
<EditText
    android:id="@+id/log_in_username_edittext"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Username"
    android:textColor="@color/black"
    android:textColorHint="@color/black"
    android:layout_marginTop="50dp">
</EditText>
```

Το button καταχώρησης νέας παραγγελίας στην κατηγορία orders and deliveries.Η ιδιότητα onClick δείχνει την μέθοδο που θα κληθεί στο πάτημα του.

```
<Button

    android:id="@+id/add_new_item_place_order_button"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="PLACE ORDER"
    android:textColor="@color/black"
    android:onClick="place_order">
</Button>
```

Το spinner επιλογής αποθήκης στην οθόνη παραμετροποίησης μιας ήδη υπάρχουσας.

```
<Spinner
    [redacted]:id="@+id/edit_warehouse_warehouse_choice_spinner"
    [redacted]:layout_width="wrap_content"
    [redacted]:layout_height="match_parent"
    [redacted]:layout_marginTop="20dp">
</Spinner>
```

Μία άλλη κατηγορία View που είναι απαραίτητη για την δημιουργία ενός layout αρχείου ειναι τα ViewGroups.Τα ViewGroups είναι views τα οποία έχουν την ιδιότητα να ενσωματώνουν άλλα views στο εσωτερικό τους. Οι βασικές κατηγορίες των ViewGroups που συνθέτουν ένα android layout είναι τα linearlayout,relativelayout,tablelayout,griddayout,scrollview κ.α. Για την παρούσα εργασία τα ViewGroups που χρησιμοποιήθηκαν για να ενσωματώσουν τα άλλα views είναι τα linearlayout,tablelayout και scrollview.Κάθε ViewGroup έχει τον δικό του χαρακτηριστικό τρόπο απεικόνισης των αντικειμένων στην οθόνη.Το LinearLayout και ανάλογα με την τιμή της ιδιότητας android:orientation η οποία μπορεί να πάρει τιμές horizontal ή vertical αναπαριστά τα περικλειόμενα views κάθετα το ένα μετά το άλλο ή οριζόντια το ένα μετά το άλλο. Το tableLayout αντίστοιχα εμφανίζει τα views σε μορφή πίνακα με κάθε σειρά να έχει ένα αριθμό στηλών όπου κάθε στήλη είναι ένα View.Στην συνέχεια ακολουθεί ο κώδικας που περιγράφει ένα Linear και ένα TableLayout.

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
</LinearLayout>

<TableLayout
                android:id="@+id/my_tablelayout"
                android:layout_width="match_parent"
                android:layout_height="275dp"
                android:layout_marginTop="10dp"
                android:stretchColumns="1">
    <TableRow>

</TableRow>

</TableLayout>
```

Στο εσωτερικό του στοιχείου tablerow περιλαμβάνονται τα views που συνθέτουν μια σειρά του tableLayout.

Παρακάτω παρατίθεται ο κώδικας που συνθέτει το layout resource αρχείο του activity_add_new_item που είναι η οθόνη καταχώρησης νέου προϊόντος καθώς και η επεξήγηση του :

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fillViewport="true">
    <LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
```

```
        android:id="@+id/add_new_item_title_textview"
        android:layout_width="match_parent"
        android:layout_height="40dp"
        android:paddingTop="10dp"
        android:text="NEW ITEM PROPERTIES"
        android:textSize="20dp"
        android:textColor="@color/black"
        android:background="@color/light_grey">
    </TextView>
    <TableLayout
        android:id="@+id/add_new_item_tablelayout"
        android:layout_width="match_parent"
        android:layout_height="360dp"
        android:layout_marginTop="25dp"
        android:stretchColumns="1">
        <TableRow>
            <TextView
                android:id="@+id/item_category_textview"
                android:padding="5dp"
                android:text="ITEM CATEGORY"
                android:textColor="@color/black"
                android:textSize="20dp"
                android:background="@color/light_grey">
            </TextView>
            <Spinner
                android:id="@+id/item_cat_choice_spinner">
            </Spinner>
        </TableRow>
        <TableRow>
            <TextView
                android:id="@+id/add_new_item_vendor_textview"
                android:padding="5dp"
```

```
                android:text="VENDOR"
                android:textColor="@color/black"
                android:textSize="20dp"
                android:background="@color/light_grey">
            </TextView>
            <EditText
                android:id="@+id/item_vendor_edittext"
                android:inputType="text"
                android:maxLength="30">


    </EditText>
        </TableRow>
        <TableRow>
            <TextView
                android:id="@+id/add_new_item_model_textview"
                android:padding="5dp"
                android:text="MODEL"
                android:textColor="@color/black"
                android:textSize="20dp"
                android:background="@color/light_grey">
            </TextView>
            <EditText
                android:id="@+id/item_model_edittext"
                android:inputType="text"
                android:maxLength="30">
            </EditText>
        </TableRow>
        <TableRow>
            <TextView
                android:id="@+id/add_new_item_weight_textview"
                android:padding="5dp"
                android:text="PACKAGE WEIGHT(kgs)"
```

```
            android:textColor="@color/black"
            android:textSize="20dp"
            android:background="@color/light_grey">
        </TextView>
        <EditText
            android:id="@+id/add_new_item_weight_edittext"
            android:inputType="numberDecimal">
        </EditText>
    </TableRow>
    <TableRow>
        <TextView
            android:id="@+id/add_new_item_length"
            android:padding="5dp"
            android:text="PACKAGE LENGTH (cm)"
            android:textColor="@color/black"
            android:textSize="20dp"
            android:background="@color/light_grey">
        </TextView>
        <EditText
            android:id="@+id/add_new_item_length_edittext"
            android:inputType="numberDecimal">
        </EditText>
    </TableRow>
    <TableRow>
        <TextView
            android:id="@+id/add_new_item_width_textview"
            android:padding="5dp"
            android:text="PACKAGE WIDTH    (cm)"
            android:textColor="@color/black"
            android:textSize="20dp"
            android:background="@color/light_grey">
        </TextView>
```

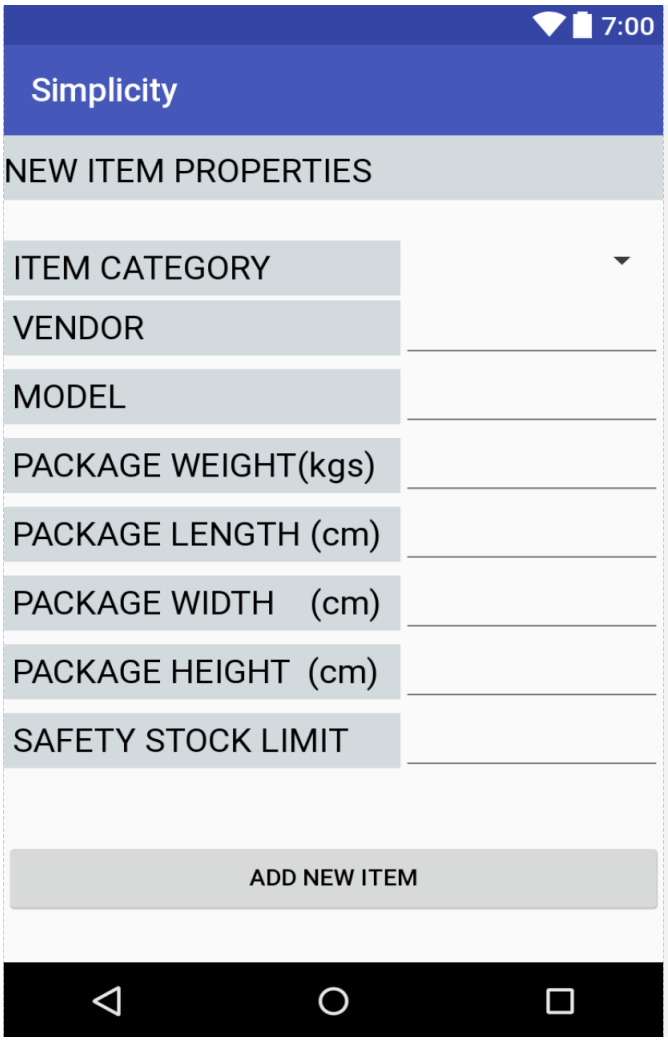
```
            <EditText
                android:id="@+id/add_new_item_width_edittext"
                android:inputType="numberDecimal">
            </EditText>
        </TableRow>
        <TableRow>
            <TextView
                android:id="@+id/add_new_item_height_textview"
                android:padding="5dp"
                android:text="PACKAGE HEIGHT  (cm) "
                android:textColor="@color/black"
                android:textSize="20dp"
                android:background="@color/light_grey">
            </TextView>
            <EditText
                android:id="@+id/add_new_item_height_edittext"
                android:inputType="numberDecimal">
            </EditText>
        </TableRow>
        <TableRow>
            <TextView
              android:id="@+id/add_new_item_safety_stock_textview"
                android:padding="5dp"
                android:text="SAFETY STOCK LIMIT "
                android:textColor="@color/black"
                android:textSize="20dp"
                android:background="@color/light_grey">
            </TextView>
            <EditText
android:id="@+id/add_new_item_safety_limit_edittext"
                android:inputType="number">
            </EditText>
```

```
            </TableRow>
        </TableLayout>
        <Button
            android:id="@+id/add_new_item_submit_button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:text="ADD NEW ITEM"
            android:textColor="@color/black"
            android:onClick="add_new_item_action">
        </Button>
    </LinearLayout>
</ScrollView>
```

Στο συγκεκριμένο Layout χρησιμοποιήθηκε μία σύνθετη δομή ενσωμάτωσης των views καθώς υπάρχουν αρκετά εμφολευμένα ομοειδή αντικείμενα. Αρχικά ορίζεται ένα ViewGroup τύπου Scrollview το οποίο θα ενσωματώνει όλα τα υπόλοιπα Views.Το scrollview έχει την ιδιότητα να δημιουργεί μπάρα κύλισης στην οθόνη έτσι ώστε να εμφανίζονται τα αντικείμενα που δεν θα μπορούσαν να φανούν στις διαστάσεις της οθόνης της κινητής συσκευής λόγω έλλειψης χώρου.Είναι το parent view όλων των υπολοίπων κι όλα τα υπόλοιπα θεωρούνται children views του scrollview.Στην συνέχεια ορίζεται ένα νέο ViewGroup τύπου LinearLayout με προσανατολισμό κάθετο.Αυτό σημαίνει ότι όλα τα άλλα views θα ενσωματωθούν σε αυτό και θα εμφανίζονται στην οθόνη σε κάθετη απεικόνιση, δηλαδή το ένα κάτω από το άλλο, και λόγω του ότι όλα περιλαμβάνονται στο parent viewgroup scrollview όσα δεν χωρούν για να εμφανιστούν στην οθόνη θα εμφανίζονται με κύλιση της οθόνης αφής της συσκευής. Σαν πρώτο view μέσα στο linearLayout δημιουργείται ένα textview με id "add_new_item_title_textview" οποίο εμφανίζει στην οθόνη το μήνυμα "NEW ITEM PROPERTIES" μέσω της ιδιότητας android:text.Από τις ιδιότητες textColor και textSize καθορίζονται το μέγεθος της γραμματοσειράς και το χρώμα των γραμμάτων. Η ιδιότητα android:background καθορίζει το χρώμα του φόντου του view ενώ οι τιμές android:margin και android:padding καθορίζουν τις αποστάσεις από τα άλλα views καθώς και το χώρο που θα καταλάβει το textview πέραν του μηνύματος που θα φέρει. Το δεύτερο από τα τρία συνολικά views που είναι children views του LinearLayout είναι ένα άλλο ViewGroup,ένα tableLayout.Το tableLayout είναι ένα ViewGroup το οποίο περιέχει οκτώ σειρές που δηλώνονται με το στοιχείο tablerow.Η πρώτη από τις οκτώ γραμμές περιέχει δύο views,ένα textview και ένα spinner,ενώ τα άλλα επτά tablerow στοιχεία περιέχουν δύο views το καθένα, ένα textview και ένα edittext.Το κάθε textview χρησιμοποιείται για να εμφανίσει στην οθόνη ένα μήνυμα και ακριβώς δίπλα του υπάρχει το αντίστοιχο edittext αντικείμενο στο οποίο ο χρήστης μπορεί να καταχωρήσει τις τιμές για κάθε ιδιότητα του νέου προϊόντος. Τέλος, το τρίτο στοιχείο που αποτελεί children view του LinearLayout είναι ένα αντικείμενο τύπου button με id add_new_item_submit_button το οποίο όταν πατηθεί από τον χρήστη οδηγεί στην κλήση της

μεθόδου add_new_item_action ώστε να καταχωρηθεί το νέο προϊόν στο σύστημα.

Το αποτέλεσμα του παραπάνω κώδικα στην οθόνη είναι η παρακάτω εικόνα:

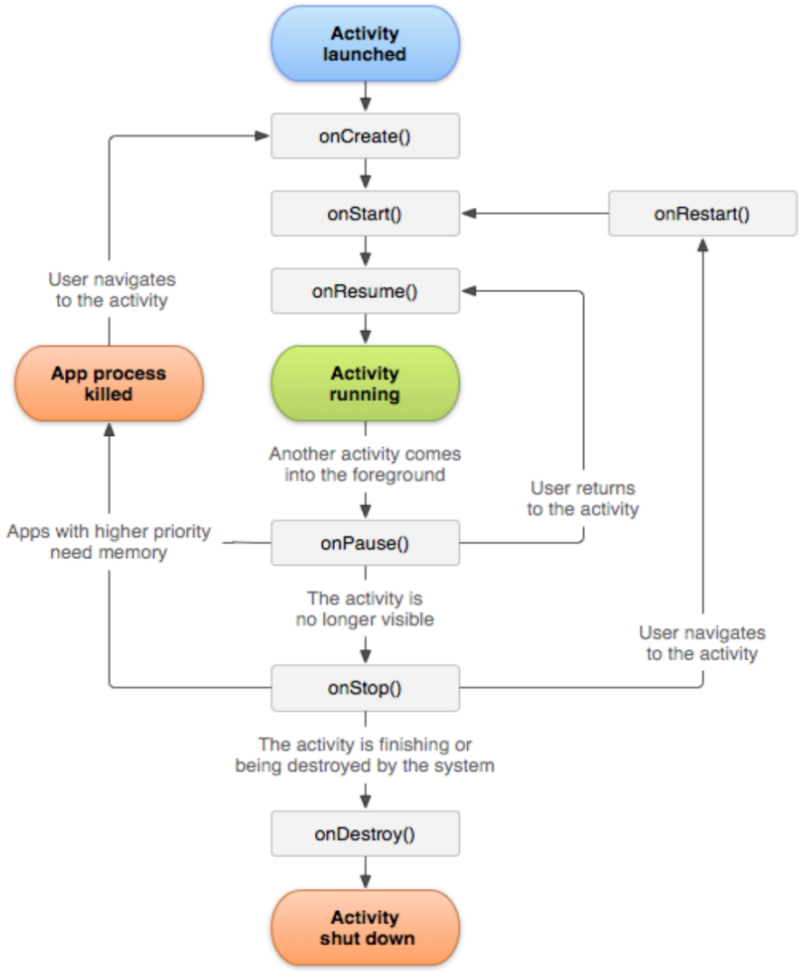


Εικόνα 5.4 Αποτέλεσμα xml κώδικα

ΚΥΚΛΟΣ ΖΩΗΣ ΕΝΟΣ ACTIVITY

Όπως αναφέρθηκε προηγουμένως κάθε μία οθόνη που εμφανίζεται σε μία εφαρμογή αποτελεί ένα ξεχωριστό αντικείμενο activity το οποίο έναν δικό του κύκλο ζωής μέσα στον κώδικα της Java που αντιστοιχεί στο συγκεκριμένο αντικείμενο.

Τα βασικά στάδια της ζωής ενός activity εκφράζονται μέσα από συγκεκριμένες μεθόδους στον κώδικα της Java με βασικότερες τις onCreate,onPause και onDestroy.Η σημαντικότερη όλων είναι η μέθοδος oncreate που είναι η μέθοδος που εκτελείται κατά την δημιουργία του αντικειμένου, δηλαδή κατά την διαδικασία αρχικοποίησης της οθόνης.Η βασικότερη της διαδικασία είναι η αναγνώριση του αρχείου xml που καταχωρείται ο κώδικας για το user interface κομμάτι και η αλληλεπίδραση του με αυτό (UI inflation).Στον κώδικα της Java επίσης γίνεται η αναγνώριση των views αντικειμένων που έχουν δηλωθεί στο resource αρχείο και η ανάθεση τους σε μεταβλητές αντίστοιχου τύπου έτσι ώστε να γίνει εφικτή η χρήση τους προγραμματιστικά.



Εικόνα 5.5 Activity Life Cycle

JAVA ΚΛΑΣΗ ΚΑΙ USER INTERFACE

Όπως αναφέρθηκε προηγουμένως κάθε activity διαχειρίζεται στον κώδικα από ένα java αρχείο, μια κλάση, το οποίο έχει συγκεκριμένα χαρακτηριστικά. Τα βασικά και χαρακτηριστικά όλων των κλάσεων που αντιστοιχούν σε activities της εφαρμογής είναι ότι είναι υποκλάσεις της AppCompatActivity κλάσης που σημαίνει ότι κληρονομούν τις μεθόδους και τις ιδιότητες της. Η συγκεκριμένη κλάση χρησιμοποιείται για την δημιουργία activities που περιέχουν ένα actionbar,δηλαδή το πάνω τμήμα της οθόνης σε μπλε φόντο όπου φαίνεται στην παρούσα εργασία το όνομα της εφαρμογής. Επιπλέον κοινό χαρακτηριστικό των κλάσεων αυτής της κατηγορίας είναι ότι χρησιμοποιούν την δική τους εκδοχή της μεθόδου onCreate σε σχέση με την κλάση από την οποία κληρονομούν τις μεθόδους τους.Όπως αναφέρθηκε προηγουμένως η onCreate μέθοδος είναι πάρα πολύ σημαντική γιατί υπάρχει ο κώδικας που θα εκτελεστεί κατά την δημιουργία του activity και ο οποίος είναι διαφορετικός για κάθε ένα. Η διαφοροποίηση της onCreate μεθόδου σε σχέση με αυτήν που κληρονομεί η κάθε κλάση από την υπερκλάση της γίνεται με χρήση ενός σχολίου κατανοητού από τον compiler του περιβάλλοντος.Το σχόλιο αυτό είναι το @override .Μετά το σχόλιο ακολουθεί η δήλωση της διαφοροποιημένης σε σχέση με την κληρονομηθείσα μεθόδου. Απαραίτητη εντολή στην αρχή του κώδικα της κάθε κλάσης που είναι υποκλάση της AppCompatActivity είναι η setContentView μέσω της οποίας το πρόγραμμα λέει στην εφαρμογή ποιο είναι το αρχειο με το user inteface που πρέπει να φορτωθεί στην μνήμη. Στον κώδικα όπως αναφέρθηκε υπάρχει η δυνατότητα για αλληλεπίδραση με τα αντίστοιχα view objects που συνθέτουν το user interface μιας εφαρμογής. Για να γίνει αυτό πρέπει τα αντικείμενα αυτά να ανατεθούν σε μεταβλητές αντίστοιχες με τον τύπο τους (button,textview,edittext,spinner) και μέσω αυτών των μεταβλητών να ελεγθούν. Πριν την δήλωση της onCreate μεθόδου δηλώνονται σαν μεταβλητές κλάσης μια σειρά από μεταβλητές του τύπου των αντικειμένων του user interface που πρέπει να ελεγχθούν, π.χ. buttons και textviews. Έπειτα, στον κώδικα της onCreate μεθόδου γίνεται ανάθεση των αντικειμένων του user interface στις μεταβλητές αυτές και έτσι μπορεί πλέον να πραγματοποιηθεί ο έλεγχος τους μέσω του κώδικα καθώς τα αντικείμενα αυτά έχουν μεθόδους που μπορεί να καλέσει το πρόγραμμα για να επηρεάσει τις ιδιότητες τους.

Παράδειγμα κλάσης AppCompatActivity (αντιστοιχεί στο user interface της κεντρικής οθόνης του χρήστη administrator)

```
package mk.logistics_erp;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import org.w3c.dom.Text;
```

```
public class admin_home_screen extends AppCompatActivity {
    Button   new_user_button,deactivate_user_button,edit_user_button;
    TextView welcome_textview;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_home_screen);
        //Assign Views to View Variables
                                                        new_user_button =
(Button)findViewById(R.id.admin_home_screen_new_user_button);
                                                    deactivate_user_button =
(Button)findViewById(R.id.admin_home_screen_deactivate_user_button);
        edit_user_button = (Button)findViewById(R.id.admin_home_screen_edit_user_button);
                                                    welcome_textview =
(TextView)findViewById(R.id.admin_home_screen_welcome_textview);
        //Set welcome text
        welcome_textview.setText("Welcome " + log_in.current_user + "!");
        welcome_textview.setTextSize(20);
    }
    public void create_new_user (View v){
        Intent intent = new Intent(this, create_new_user.class);
        startActivity(intent);
    }
    public void deactivate_user (View v){
        Intent intent = new Intent(this, deactivate_user.class);
        startActivity(intent);
    }
    public void edit_user (View v){
        Intent intent = new Intent(this, edit_user.class);
        startActivity(intent);
    }
```

}

ΕΠΙΚΟΙΝΩΝΙΑ ΕΦΑΡΜΟΓΗΣ ΜΕ WEB SERVER

Όπως καταγράφηκε στο κεφάλαιο 3 η αρχιτεκτονική της εφαρμογής είναι τέτοια ώστε δεν μπορεί το επίπεδο εφαρμογής (application layer) να επικοινωνήσει άμεσα με την βάση δεδομένων αλλά πρέπει να μεσολαβήσει το επίπεδο του application server.Ο application server είναι αυτός που δέχεται τα αιτήματα για επικοινωνία με την βάση δεδομένων και τις αντίστοιχες παραμέτρους και επιστρέφει στην εφαρμογή τα αποτελέσματα της επικοινωνίας αυτής. Η επικοινωνία της εφαρμογής με τον application server γίνεται μέσω της κλήσης php αρχείων που φιλοξενούνται σε αυτόν και είναι διαφορετικά για κάθε απαιτούμενη ενέργεια στο σύστημα. Η δυνατότητα αυτή, δηλαδή η επεξεργασία δεδομένων στο μη εμφανές τμήμα της εφαρμογής και η απεικόνισή τους στο user interface γίνεται με την χρήση της βοηθητικής κλάσης AsyncTask.Η AsyncTask κλάση επιτρέπει την σωστή διαχείριση του νήματος του user interface.Η χρήση της AsyncTask κλάσης στην εφαρμογή έγινε με την βοηθητική βιβλιοθήκη com.kosalgeek.asynctask η οποία περιλαμβάνει το interface AsyncResponse το οποίο χρησιμοποιεί η κλάση που θέλει να επικοινωνήσει με τον application server και της επιτρέπει να καλέσει το php αρχείο δηλώνοντας την τοποθεσία του και καλώντας την μέθοδο execute ενός αντικειμένου της κλάσεως PostResponseAsyncTask που ορίζεται στην ίδια βοηθητική βιβλιοθήκη. Στην συνέχεια και μέσω της μεθόδου processFinish διαχειρίζεται το αποτέλεσμα που επιστρέφει στην εφαρμογή ο application server.

Παράδειγμα κώδικα που κάνει χρήση της παραπάνω διαδικασίας επικοινωνίας με τον application server (καταχώρηση νέας παραγγελίας) :

```
package mk.logistics_erp;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.kosalgeek.asynctask.AsyncResponse;
```

```
import com.kosalgeek.asynctask.PostResponseAsyncTask;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.HashMap;
import static mk.logistics_erp.log_in.web_server;

public class make_new_order extends AppCompatActivity implements AsyncResponse {
    Spinner item_cat_choice_spinner,vendor_choice_spinner,model_choice_spinner;
                                                                    Button
confirm_item_cat_button,confirm_vendor_button,confirm_model_button,place_order_bu
tton;
    TextView vendor_textview,model_textview,quantity_textview;
    EditText quantity_edittext;



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_make_new_order);
                        //Toast.makeText(this,    orders_and_deliveries_home.items_table[0][2],
Toast.LENGTH_LONG).show();
        item_cat_choice_spinner = (Spinner)findViewById(R.id.item_cat_choice_spinner);
                                                    confirm_item_cat_button      =
(Button)findViewById(R.id.add_new_item_confirm_item_categoty_button);
        vendor_choice_spinner = (Spinner)findViewById(R.id.vendor_choice_spinner);
        vendor_choice_spinner.setEnabled(false);
        vendor_choice_spinner.setVisibility(View.GONE);
                                                confirm_vendor_button      =      (Button)
findViewById(R.id.add_new_item_confirm_vendor_button);
        confirm_vendor_button.setEnabled(false);
        confirm_vendor_button.setVisibility(View.GONE);
        model_choice_spinner = (Spinner)findViewById(R.id.model_choice_spinner) ;
        model_choice_spinner.setEnabled(false);
```

```
        model_choice_spinner.setVisibility(View.GONE);
                                                    confirm_model_button =
(Button)findViewById(R.id.add_new_item_confirm_model_button) ;
        confirm_model_button.setEnabled(false);
        confirm_model_button.setVisibility(View.GONE);
        vendor_textview = (TextView)findViewById(R.id.item_vendor_textview);
        vendor_textview.setVisibility(View.GONE);
        model_textview = (TextView)findViewById(R.id.item_model_textview) ;
        model_textview.setVisibility(View.GONE);
        quantity_textview = (TextView)findViewById(R.id.item_quantity_textview);
        quantity_textview.setVisibility(View.GONE);
        place_order_button = (Button)findViewById(R.id.add_new_item_place_order_button);
        place_order_button.setVisibility(View.GONE);
        quantity_edittext = (EditText)findViewById(R.id.item_quantity_edittext);
        quantity_edittext.setEnabled(false);
        quantity_edittext.setVisibility(View.GONE);
        //Set Spinner adapter
                                        ArrayAdapter<String> adapter3 = new
ArrayAdapter<String>(this,R.layout.support_simple_spinner_dropdown_item,orders_and_d
eliveries_home.item_categories );
         adapter3.setDropDownViewResource(R.layout.support_simple_spinner_dropdown_ite
m);
        item_cat_choice_spinner.setAdapter(adapter3);
    }


    public void confirm_item_category (View v){
        String item_cat = item_cat_choice_spinner.getSelectedItem().toString();
        help_class hlp = new help_class();
        String [] vnds = hlp.find_vendor(item_cat);
        item_cat_choice_spinner.setEnabled(false);
                                        ArrayAdapter<String> adapter4 = new
ArrayAdapter<String>(this,R.layout.support_simple_spinner_dropdown_item, vnds);
      adapter4.setDropDownViewResource(R.layout.support_simple_spinner_dropdown_item)
```

```
;
        vendor_choice_spinner.setAdapter(adapter4);
        vendor_choice_spinner.setVisibility(View.VISIBLE);
        vendor_choice_spinner.setEnabled(true);
        vendor_textview.setVisibility(View.VISIBLE);
        confirm_vendor_button.setEnabled(true);
        confirm_vendor_button.setVisibility(View.VISIBLE);
    }

    public void confirm_vendor (View v){
        String item_vendor = vendor_choice_spinner.getSelectedItem().toString();
        help_class hlp = new help_class();
        String models[] = hlp.find_model(item_vendor);
        vendor_choice_spinner.setEnabled(false);
                                                ArrayAdapter<String>      adapter5      =      new
ArrayAdapter<String>(this,R.layout.support_simple_spinner_dropdown_item, models);
         adapter5.setDropDownViewResource(R.layout.support_simple_spinner_dropdown_ite
m);
        model_choice_spinner.setAdapter(adapter5);
        model_textview.setVisibility(View.VISIBLE);
        model_choice_spinner.setEnabled(true);
        model_choice_spinner.setVisibility(View.VISIBLE);
        confirm_model_button.setEnabled(true);
        confirm_model_button.setVisibility(View.VISIBLE);
    }

    public void confirm_model (View v) {
        model_choice_spinner.setEnabled(false);
        quantity_textview.setVisibility(View.VISIBLE);
        quantity_edittext.setEnabled(true);
        quantity_edittext.setVisibility(View.VISIBLE);
        place_order_button.setEnabled(true);
```

```
        place_order_button.setVisibility(View.VISIBLE);
    }

    public void place_order (View v){
        String warehouse,PHP_Url;
        item_cat_choice_spinner.setEnabled(true);
        vendor_choice_spinner.setEnabled(true);
        model_choice_spinner.setEnabled(true);
        HashMap PostData = new HashMap();
        PostData.put("PLACED_BY",log_in.current_user);
            PostData.put("ITEM_CATEGORY",item_cat_choice_spinner.getSelectedItem().toStri
ng());
        PostData.put("ITEM_VENDOR",vendor_choice_spinner.getSelectedItem().toString());
        PostData.put("ITEM_MODEL",model_choice_spinner.getSelectedItem().toString());
        PostData.put("QUANTITY",quantity_edittext.getText().toString());
        PostResponseAsyncTask task = new PostResponseAsyncTask(this,PostData);
        PHP_Url = web_server + "/10_place_order.php";
        task.execute(PHP_Url);
    }

    @Override
    public void processFinish (String result){
        Toast.makeText(this, result, Toast.LENGTH_LONG).show();
    }
}
```

Αναλύοντας τα βασικά σημεία του κώδικα στην συγκεκριμένη κλάση θα πρέπει να επισημανθούν :

1. Το πρώτο τμήμα της όπου με την εντολή import ενσωματώνονται στο πρόγραμμα όλες οι απαραίτητες βιβλιοθήκες.
2. Η δήλωση της κλάσης (class declaration) όπου ενημερώνεται ο compiler ότι η κλάση είναι υποκλάση της AppCompatActivity και ότι χρησιμοποιεί το interface AsyncResponse.

3. Η δήλωση των μεταβλητών κλάσης που αφορούν τα views τα οποία υπάρχουν στο layout αρχείο και με τα οποία θέλει το πρόγραμμα να αλληλεπίδραση.
4. Ο κώδικας της onCreate μεθόδου όπου αρχικοποιείται το user interace της εφαρμογής μέσω της εντολής setContentView και στην οποία δίνεται ως παράμετρος το όνομα του layout αρχείου που αντιστοιχεί στο συγκεκριμένο activity.
5. Το τμήμα της onCreate μεθόδου όπου τα αντικείμενα views του user interface γίνονται ανάθεση στις αντίστοιχες μεταβλητές ώστε να χρησιμοποιηθούν από το πρόγραμμα.
6. Οι μέθοδοι confirm_item_category, confirm_vendor, confirm_model που περιγράφου την λειτουργία του κάθε αντίστοιχου button αντικειμένου στην οθόνη του χρήστη.
7. Η μέθοδος place_order η οποί καλείται στο πάτημα του button place order στο user interface και η οποία καλεί το αντίστοιχο αρχείο στον application server συλλέγοντας από τα views τις απαραίτητες πληροφορίες που πρέπει να σταλούν ως παράμετροι.
8. Η μέθοδος processFinish η οποία διαχειρίζεται το αποτέλεσμα που επιστρέφεται στο πρόγραμμα από τον application server.

ΕΠΙΚΟΙΝΩΝΙΑ WEB SERVER ΜΕ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Όπως αναφέρθηκε στην προηγούμενη ενότητα η επικοινωνία της εφαρμογής με τον web server γίνεται διότι δεν υπάρχει απ' ευθείας πρόσβαση της στην βάση δεδομένων. Ο web server αντίθετα επικοινωνεί με την βάση δεδομένων και μπορεί να εκτελέσει σε αυτήν τα απαραίτητα CRUD operations (Create,Read,Update,Delete).Για να γίνει αυτό υπάρχει ένα php αρχείο το οποίο φέρει τα στοιχεία σύνδεσης με την βάση δεδομένων και το οποίο καλείται στα υπόλοιπα php αρχεία που θέλουν να επικοινωνήσουν με την βάση.

Το αρχείο είναι το 1_sql_connection_variables.php .

```
<?php
$servername = "192.168.1.16:3306";
$username = "root";
$password = "@dm1n";
$dbname = "simplicity_2";
?>
```

Τα επόμενα αρχεία που θέλουν να εκτελέσουν μια ενέργεια σε σχέση με την βάση δεδομένων απλά ενσωματώνουν στον κώδικά τους το αρχείο αυτό όπως το 4_add_warehouse_add_warehouse.php του οποίου ο κώδικας ακολουθεί :

```
<?php
```

```
// Import Connection Variables
require '1_sql_connection_variables.php';
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
// Perform Authentication Check
if ( isset($_POST['Warehouse_Name']) && isset($_POST['Paths_Number']) && isset($_POST['Spots_Number']) && isset($_POST['Spot_max_Weight'])
        && isset($_POST['Spot_Length']) && isset($_POST['Spot_Width']) && isset($_POST['Spot_Height'])) {
        $Warehouse_Name  = $_POST['Warehouse_Name'];
        $Paths_Number_temp   = $_POST['Paths_Number'];
        $Spots_Number_temp   = $_POST['Spots_Number'];
        $Spot_Length_temp    = $_POST['Spot_Length'];
        $Spot_Width_temp     = $_POST['Spot_Width'];
        $Spot_Height_temp    = $_POST['Spot_Height'];
        $Spot_max_Weight_temp    = $_POST['Spot_max_Weight'];
        //temp variables to final (strings to int,double)
        $Paths_Number    = intval($Paths_Number_temp);
        $Spots_Number    = intval($Spots_Number_temp);
        $Spot_max_Weight = floatval($Spot_max_Weight_temp);
        $Spot_Length     = floatval($Spot_Length_temp);
        $Spot_Width      = floatval($Spot_Width_temp);
        $Spot_Height     = floatval($Spot_Height_temp );
        //Form Query
        $sql_query = "INSERT INTO WAREHOUSES (WAREHOUSE_NAME, NUM_OF_PATHS, SPOTS_PER_PATH, MAX_SPOT_WEIGHT, SPOT_LENGTH, SPOT_WIDTH,SPOT_HEIGHT) VALUES ('$Warehouse_Name','$Paths_Number','$Spots_Number','$Spot_max_Weight','$Spot_Length','$Spot_Width','$Spot_Height')";
        if ($conn->query($sql_query)){
                echo "Warehouse added successfully";
        } else {
                echo "Warehouse was not added";
        }
```

```
}
//Close Connection
$conn->close();
?>
```

Με την εντολή " require '1_sql_connection_variables.php'; " το αρχείο ενσωματώνει τα στοιχεία σύνδεσης. Στην συνέχεια δημιουργεί ένα αντικείμενο conn το οποίο πραγματοποιεί τη σύνδεση στην βάση δεδομένων και μέσω του οποίου εκτελούνται οι απαραίτητες εντολές ή επιστρέφει μήνυμα αποτυχίας σύνδεσης. Το αποτέλεσμα το οποίο επιστρέφεται στο πρόγραμμα γίνεται μέσω της εντολής echo που θα εκτελεστεί.

5.2 Τεχνικά ζητήματα απόδοσης βάσης δεδομένων

Επιπλέον της δημιουργίας της βάσης δεδομένων και της εφαρμογής αυτής καθεαυτής έγιναν οι απαραίτητες ενέργειες για την βελτιστοποίηση της απόδοσης της βάσης δεδομένων. Η απόδοση της βάσης δεδομένων στα ερωτήματα που αποστέλλονται από τον application server δεν διαφοροποιείται σε μετρήσιμο βαθμό όσο τα δεδομένα τα υπάρχουν σε αυτήν είναι τόσα ώστε να καταλαμβάνουν χώρο μέχρι και κάποιων εκατοντάδων σελίδων στον φυσικό δίσκο, όμως μπορεί να προκληθεί καθυστέρηση όταν αυτά γίνουν εκατοντάδες ή χιλιάδες σελίδες. Για τον λόγο αυτό και σαν μέτρο πρόληψης αυτής της ενδεχόμενης καθυστέρησης όταν το σύστημα θα έχει μεγαλώσει αρκετά δημιουργήθηκαν στην βάση δεδομένων οι κατάλληλοι indexes (δείκτες) οι οποίοι αποτελούν ξεχωριστές οντότητες της βάσης, καταλαμβάνουν χώρο στον δίσκο πέραν των δεδομένων των πινάκων και στην πράξη αυτό που κάνουν είναι να ταξινομούν τα δεδομένα με βάση τις στήλες που ο χρήστης έχει ορίσει ώστε στα αντίστοιχα ερωτήματα που χρησιμοποιούν αυτές τις στήλες να είναι πολύ πιο άμεση η απόκριση του συστήματος διαχείρισης της βάσης. Οι κατάλληλοι δείκτες προκύπτουν από τα ερωτήματα που πραγματοποιούνται στην βάση και στα πλαίσια της εφαρμογής κρίθηκε σκόπιμο να δημιουργηθούν οι ακόλουθοι :

1. Username στον πίνακα users και στην στήλη username
2. Comp_users στον πίνακα users που αποτελεί σύνθετο index στις στήλες (username,password και flag_active)
3. Wrh_name στον πίνακα warehouses και στην στήλη warehouse_name
4. Mdl στον πίνακα orders και στην στήλη item_model
5. Mdl στον πίνακα items και στην στήλη model

Τέλος, έπειτα από την διαδικασία δημιουργίας των κατάλληλων indexes δημιουργήθηκε κατάλληλη διαδικασία συντήρησης της βάσης δεδομένων μέσω της εντολής analyze table η οποία δημιουργεί τα απαραίτητα στατιστικά χρήσης του πίνακα και η οποία εκτελείται με περιοδικότητα που ορίζεται από την εργασία του task_scheduler των windows που την εκτελεί.

5.3 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό αναλύθηκαν τα βασικά ζητήματα ανάπτυξης της εφαρμογής σε σχέση με τον κώδικα της και το βασικό μέτρο πρόληψης για την καλή απόδοση της λειτουργίας του συστήματος διαχείρισης της βάσης δεδομένων.

Όσον αφορά τον κώδικα ανάπτυξης καταγράφηκε η βασική δομή ενός project του εργαλείου ανάπτυξης που χρησιμοποιήθηκε (Android Studio).Αυτή ήταν η σύνθεση τεσσάρων κατηγοριών αρχείων : αρχεία manifest,αρχεία java,αρχεία resource και αρχεία gradle.Στην συνέχεια, καταγράφηκε ο κώδικας του manifest αρχείου της εφαρμογής και αναφέρθηκε η σχέση εξάρτησης μεταξύ των αρχείων layout που χρησιμοποιούνται για την δημιουργία του user interface τμήματος της εφαρμογής και των java κλάσεων που τα διαχειρίζονται. Καταγράφηκαν τα βασικά σημεία του κώδικα που είναι κοινά στις υποκλάσεις της κατηγορίας AppCompatActivity. Αυτά ήταν η onCreate μέθοδος και η κλήση του αρχείου layout ως πηγή του user interface,η ανάθεση των views σε μεταβλητές αντίστοιχου τύπου (buttons,textviews,edittexts,spinners,linearlayouts),η κλήση των php αρχείων του web server που πραγματοποιούν την επικοινωνία με την βάση δεδομένων μέσω του interface AsyncResponce και η επεξεργασία του αποτελέσματος που δέχεται η εφαρμογή σαν απάντηση από τον application server μέσω της μεθόδου processfinish.Κατόπιν αυτών των αναλύσεων δόθηκε παράδειγμα κώδικα μιας κλάσης που αντιστοιχούσε στην οθόνη (activity) της καταχώρησης νέας παραγγελίας. Στην συνέχεια, εξετάστηκε η μέθοδος επικοινωνίας του application server με την βάση δεδομένων και δόθηκε παράδειγμα κώδικα το οποίο εκτελεί καταχώρηση στην βάση νέας αποθήκης. Τέλος, αναφέρθηκαν μέτρα πρόληψης για την απόδοση της βάσης δεδομένων σε ενδεχόμενο μελλοντικό άλμα στην τάξη μεγέθους των δεδομένων της εφαρμογής.

ΚΕΦΑΛΑΙΟ 6 ΣΥΜΠΕΡΑΣΜΑΤΑ

6.1 Σύνοψη εργασίας και συμπεράσματα

Στο σημείο αυτό και με σύντομο τρόπο θα επιχειρηθεί μία σύνοψη της παρούσας εργασίας και καταγραφή της συνεισφοράς της στους τομείς που μπορεί να επηρεάσει.

Το προϊόν της εργασίας είναι ένα πληροφοριακό σύστημα ERP (enterprise resource planning), mobile τεχνολογίας για κινητές συσκευές Android και στοχευμένο στις ανάγκες του κλάδου της εφοδιαστικής αλυσίδας. Η συνεισφορά του έγκειται στην ύπαρξη ενός συστήματος που καλύπτει τις βασικές λειτουργικές ανάγκες μιας εταιρείας του χώρου με τρόπο όσο το δυνατόν απλούστερο, ώστε να βοηθήσει προσωπικό τεχνολογικά μη εξοικειωμένο να ενταχθεί στην ψηφιακή εποχή της καταγραφής των επιχειρησιακών συναλλαγών. Το κοινό, στο οποίο στοχεύει το παρών προϊόν είναι οι μικρομεσαίες επιχειρήσεις του κλάδου (SMEs) και οι ανάγκες οι οποίες καλύπτονται από το σύστημα είναι η πληροφορία σε πραγματικό χρόνο διαθεσιμότητας αποθεματικού προϊόντος, η καταχώρηση και ο έλεγχος σε πραγματικό χρόνο της κατάστασης των παραγγελιών, η καταχώρηση σε πραγματικό χρόνο παραλαβής παραγγελίας, με παράλληλη αυτόματη ενημέρωση των αποθεμάτων, ο αυτοματοποιημένος έλεγχος για προϊόντα με αποθεματικό κάτω του ορισθέντος ορίου ασφαλείας με παράλληλη δυνατότητα άμεσης παραγγελίας της διαφοράς έως του ορίου ασφαλείας, η δυνατότητα αναζήτησης συγκεκριμένου προϊόντος και ενημέρωσης του χρήστη για την διαθεσιμότητα του σε ορισμένα σημεία αποθήκευσης, η αναζήτηση ιστορικού παραγγελιών συγκεκριμένου προϊόντος και η επικοινωνία των χρηστών σε πραγματικό χρόνο. Πλεονέκτημα της εφαρμογής θεωρείται η ίδια η φύση της καθώς το είδος των συσκευών που υποστηρίζει (κινητές συσκευές) απελευθερώνει τον χρήστη από τον περιορισμό της πρόσβασης στο πληροφοριακό σύστημα μόνο από τον σταθμό εργασίας δίνοντας του ελευθερία κίνησης σε έναν χώρο στον οποίο του είναι απαραίτητη καθώς η μετακίνηση και ο έλεγχος επί του σημείου ενδιαφέροντος σε μια επιχείρηση του κλάδου αυτού είναι αναγκαίος. Για την ολοκλήρωση της εργασίας απαιτήθηκε ο συνδυασμός πολλών τεχνολογιών όπως η Java,η PHP,η XML,η SQL,ο Apache Web Server,το σύστημα διαχείρισης βάσεων δεδομένων MySQL και το περιβάλλον ανάπτυξης κινητών εφαρμογών Android Studio της εταιρείας Google αποτελώντας όλες τεχνολογία αιχμής στον κλάδο τους διασφαλίζοντας ότι η εφαρμογή δημιουργήθηκε με τα αρτιότερα τεχνολογικά διαθέσιμα μέσα.

6.2 Προτάσεις για μελλοντικές επεκτάσεις

Με δεδομένο ότι η παρούσα εφαρμογή καλύπτει τις βασικές ανάγκες μιας επιχείρησης του χώρου και ως εκ τούτου στοχεύει σε ένα μικρομεσαίο επιχειρησιακό περιβάλλον, του οποίου οι ανάγκες δεν είναι σε πολύ υψηλό βαθμό ανεπτυγμένες, οι προτάσεις για επεκτάσεις της παρούσας εργασίας αφορούν την επέκταση των καλυπτόμενων αναγκών του κλάδου με σκοπό

την επέκταση του επιχειρηματικού κοινού, που μπορεί να εξυπηρετήσει. Για αυτό, προτείνεται η επέκταση σε άλλες κατηγορίες προγραμμάτων, η επέκταση των δυνατοτήτων των υπαρχόντων κατηγοριών και η ενσωμάτωση νέων τεχνολογικών δυνατοτήτων στο σύστημα. Πιο συγκεκριμένα, ως προς την επέκταση των δυνατοτήτων των υπαρχόντων κατηγοριών του συστήματος προτείνεται η δημιουργία ροής εργασίας (workflow) με ανάθεση εργασιών από εξουσιοδοτημένο σε ρόλο χρήστη σε άλλον χρήστη του συστήματος με δυνατότητα του δεύτερου επιβεβαίωσης του δικού του τμήματος της εργασίας και παράλληλη ανάθεση σε τρίτο χρήστη του επόμενου βήματος μέχρι την τελική ολοκλήρωση της αρχικά δοθείσας εργασίας προς εκτέλεση και τελικά ενημέρωση του εκκινούν τα την διαδικασία. Επιπλέον, προτείνεται η επέκταση της δυνατότητας επικοινωνίας πραγματικού χρόνου με δυνατότητα αποστολής ηχητικού μηνύματος ή αποστολής αρχείου (διαχείριση αρχείων). Ως προς την επέκταση της εφαρμογής σε νέες κατηγορίες προτείνεται η ενσωμάτωση στο σύστημα της κατηγορίας πληρωμών, ώστε να ολοκληρωθεί η διαδικασία παραλαβής προϊόντος, χωρίς να απαιτείται περεταίρω διαδικασία εκτός συστήματος, η επέκταση της εφαρμογής στην κατηγορία του Plant Maintenance ώστε να μπορεί μέσω του συστήματος να ελεγχθεί η συντήρηση και ο κύκλος εργασιών που αφορούν την καλή κατάσταση μιας εγκατάστασης, καθώς και η ενσωμάτωση στο σύστημα κατηγορίας διαχείρισης αρχείων (Document Control), έτσι ώστε να μπορεί ο εργαζόμενος να έχει πρόσβαση ανά πάσα στιγμή σε έγγραφα οδηγιών, εταιρικών διαδικασιών και εγγράφων προς συμπλήρωση, κατά την εκτέλεση μιας εργασίας. Τέλος, ως προς το κομμάτι της ενσωμάτωσης νέων τεχνολογιών στο σύστημα προτείνεται η υλοποίηση δυνατότητας σκαναρίσματος QR ετικετών με παράλληλη ενημέρωση των αποθεματικών κατά την παραλαβή προϊόντων και η υλοποίηση δυνατότητας γραφικής απεικόνισης των θέσεων αποθήκευσης μιας αποθήκης, με παράλληλα διαθέσιμη πληροφορία για τα δεδομένα των προϊόντων που είναι αποθηκευμένα εκεί. Με τις προαναφερθείσες προσθήκες η εφαρμογή μπορεί να αποτελέσει ενναλακτική πρόταση πληροφοριακού συστήματος για μεγαλύτερο εύρος επιχειρήσεων, με μεγαλύτερο αριθμό εργαζομένων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Concepts in Enterprise Resource Planning,2013 ,Ellen F. Monk ,Bret J. Wagner

2. The Impact of Enterprise Resource Planning (ERP) Systems Implementation on Business Performance,2013 ,Nishad Nawaz

3. SaaS versus on-premise ERP, 2012 ,Jonathan Gross

4. ERP in the Cloud.Is it Ready?Are you? , 2013 , Booz and Company

5. Shared leadership in enterprise resource planning and human resource management system implementation ,2013 , Julia E. Hoch , James H. Dulebohn b

6. Oracle Accelerate for midsize companies,2011

7. ERP Demystified, 2008 , Alexis Leon

8. Enterprise Resource Planning Systems, 2000 ,Daniel O'Leary

9. ERP system implementation: benefits and economic effectiveness , 2013 , Andrejs Tambovcevs & Tatjana Tambovceva

10. How Does Financial Reporting Quality Relate to Investment Efficiency? Journal of Accounting and Economics, 2009 , MIT Open Access Articles

11. Risks in Enterprise cloud computing: The perspective of IT experts, 2013 , University of Sheffield

12. Adopting Cloud ERP in Small and Medium Enterprises:Benefits and Challenges, 2016 ,R. Meganathan , P. Jeyanthi

13. Delivering Software as a Service, 2007 , A.Dubey

14. Above the clouds:A Berkeley View of Cloud Computing, 2009

15. The Evolution of Enterprise Resource Planning Systems ,Teodor Beleţ , Anca Alexandra Purcărea ,2017

16. The Impact of Company Size on factors influencing EROP Adoption in Ireland,Paul Lindopp ,2015