



Τεχνολογία Λογισμικού

Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα Ηλεκτρολόγων Μηχανικών και
Μηχανικών Υπολογιστών Α.Π.Θ

8ο Εξάμηνο - Άνοιξη 2018



Robot Bartender for the human customer

Σχεδίαση Συστήματος

Θεοδωρίδου Χριστίνα	christtk@auth.gr
Μπλάννινγκ Φρανκ	frankgou@ece.auth.gr
Παρασκευαΐδης Κωνσταντίνος	konstapf@ece.auth.gr
Πάρναλης-Παλαντζίδης Γιώργος	parnalis@ece.auth.gr

23 Μαΐου 2018

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Φρανκ	13 Μαι 2018	Αρχικό έγγραφο	0.1
Γιώργος	15 Μαι 2018	Συγγραφή κεφαλαίου 2	0.2
Κώστας	17 Μαι 2018	Συγγραφή κεφαλαίου 3	0.3
Χριστίνα	18 Μαι 2018	Συγγραφή κεφαλαίου 4	0.4
Φρανκ	21 Μαι 2018	Διορθώσεις	0.5
Κώστας	22 Μαι 2018	Συγγραφή κεφαλαίου 5	0.6
Φρανκ	23 Μαι 2018	Εισαγωγικό κεφάλαιο	1.0

Μέλη της Ομάδας Ανάπτυξης

Όνομα	Ομάδα	email
Θεοδωρίδου Χριστίνα	Documentation	christtk@auth.gr
Μπλάννινγκ Φρανκ	Frontend Dev	frankgou@ece.auth.gr
Παρασκευαΐδης Κωνσταντίνος	Backend Dev	konstapf@ece.auth.gr
Πάρναλης-Παλαντζίδης Γιώργος		parnalis@ece.auth.gr

Περιεχόμενα

1	Εισαγωγή	4
1.1	Στόχος του εγγράφου	4
1.2	Αντικείμενο του Λογισμικού	4
1.3	Ορισμοί, Ακρωνύμια, Συντομεύσεις	4
1.4	Τυπογραφικές παραδοχές του εγγράφου	5
1.5	Στόχοι Σχεδίασης	5
1.6	Αναγνωστικό κοινό και τρόπος ανάγνωσης	5
1.7	Επισκόπηση Εγγράφου	5
2	Δυναμική Μοντελοποίηση του Συστήματος	7
2.1	Σενάριο Χρήσης 3	7
2.2	Σενάριο Χρήσης 6	7
2.3	Σενάριο Χρήσης 8	9
2.4	Σενάριο Χρήσης 9	9
3	Προτεινόμενη Αρχιτεκτονική Λογισμικού	11
3.1	Αρχιτεκτονική πελάτη-διακομιστή (client-server)	11
3.2	Αρχιτεκτονική Συστήματος Robobar	11
4	Αποδόμηση του Συστήματος	12
4.1	Υποσυστήματα	12
4.1.1	Υποσύστημα LoginHandler	12
4.1.2	Υποσύστημα MakeOrderHandler	12
4.1.3	Orders	13
4.1.4	EmptyBottlesHandler	13
4.1.5	EditDrinksHandler	13
4.1.6	Database	14
4.1.7	Robots	14
4.1.8	Printer	14
4.1.9	GUIs	15
4.2	Απεικόνιση Υλικού / Λογισμικού	15
4.2.1	Client Device (PDA)	15
4.2.2	System server	15
4.2.3	Συνολικό Διάγραμμα Ανάπτυξης	16
4.3	Διαχείριση Μόνιμων Δεδομένων	17
4.4	Έλεγχος πρόσβασης και Ασφάλεια	17
4.5	Γενικός έλεγχος λογισμικού	20
5	Οριακές Συνθήκες	21
5.1	Έναρξη εφαρμογής	21
5.2	Τερματισμός λογισμικού	21
5.3	Σφάλματα Λογισμικού	22
5.4	Διακοπή τροφοδοσίας	23
A	Λίστα Ιχνηλασιμότητας	24

1 Εισαγωγή

1.1 Στόχος του εγγράφου

Ακολουθώντας τα στάδια ανάπτυξης λογισμικού, στα προηγούμενα έγγραφα, το “Έγγραφο Απαιτήσεων Χρηστών” και το “Έγγραφο Απαιτήσεων Λογισμικού”, καθορίστηκαν όσο το δυνατόν πιο εύστοχα οι απαιτήσεις χρηστών και λογισμικού, αντίστοιχα. Στο παρόν έγγραφο σειρά έχει ο αρχιτεκτονικός σχεδιασμός του συστήματος.

Αρχικά σημαντική προϋπόθεση για την σχεδίαση του συστήματος είναι η δυναμική μοντελοποίηση του συστήματος Robobar με τον πλέον λεπτομερή τρόπο. Αυτός είναι η παρουσίαση των ροών που περιγράφουν το πώς οι κλάσεις που αναπτυχτήκαν κατά τη στατική μοντελοποίηση των κλάσεων ικανοποιούν τα σενάρια χρήσης που δημιουργήθηκαν στο έγγραφο απαιτήσεων χρηστών. Η δημιουργία δηλαδή των διαγραμμάτων ροών για την εκτενή περιγραφή της συμπεριφοράς του συστήματος κατά την εκτέλεση των λειτουργιών του αλλά και κατά την αλληλεπίδραση με τους χρήστες και τα εξωτερικά συστήματα.

Επιπλέον η αποδόμηση του συστήματος σε υποσυστήματα αποτελεί πάγια τακτική για την ανάπτυξη ενός λογισμικού και επιτυγχάνεται με την εξής διαδικασία: το υπάρχον σύστημα διαιρείται σε υποσυστήματα, τα οποία αλληλεπιδρούν. Στη συνέχεια επιλέγεται η κατάλληλη αρχιτεκτονική ώστε να έχουμε το καλύτερο δυνατό αποτέλεσμα ως προς την επικοινωνία μεταξύ των υποσυστημάτων αλλά και τη λειτουργικότητα συστήματος.

Τέλος στο παρόν έγγραφο γίνεται μία λεπτομερής ανάλυση της αρχιτεκτονικής του συστήματος Robobar, με γνώμονα τις απαιτήσεις χρηστών και λογισμικού που περιγράφηκαν στα δύο προηγούμενα έγγραφα. Τα δομικά στοιχεία του συστήματος, ο τρόπος που επικοινωνούν μεταξύ τους και τυχόν περιορισμοί που πρέπει να ληφθούν υπόψη, αποτελούν μέρη της ανάλυσης αυτής.

1.2 Αντικείμενο του Λογισμικού

Το λογισμικό Robobar έχει ως αντικείμενο το σύνολο των δραστηριοτήτων που χρειάζονται ώστε να λειτουργήσει ένα μπαρ. Αναλυτικότερα, το λογισμικό RoboBar έχει ως στόχο την γρήγορη και εύκολη εξυπηρέτηση των πελατών ενός μπαρ.

Το σύστημα αλληλεπιδρά με δυο είδη χρηστών τους απλούς χρήστες-εργαζόμενους του μπαρ και των διαχειριστών. Ο χρήστης μπορεί να δεχτεί μια παραγγελία από τους πελάτες του μπαρ από το υπάρχον μενού, όπως επίσης έχει την δυνατότητα να ενημερωθεί ή να επιλέξει την ακύρωση μιας εκκρεμής παραγγελίας ανάλογα με το τι θα ζητηθεί από τους πελάτες. Ο διαχειριστής του συστήματος έχει όλες τις παραπάνω δυνατότητες και επιπλέον μπορεί να επεξεργαστεί το μενού του καταστήματος δηλαδή να προσθέσει ή να αναιρέσει συνταγές και υλικά.

Οι παραπάνω λειτουργίες του συστήματος RoboBar εξυπηρετούνται με χρήση μια βάσης δεδομένων που περιέχει όλα τα απαραίτητα δεδομένα, καθώς και ενός ρομπότ ΝΑΟ το οποίο εκτελεί τις παραγγελίες των πελατών.

1.3 Ορισμοί, Ακρωνύμια, Ξυντομεύσεις

- OA: Ομάδα Ανάπτυξης
- GUI: Graphical User Interface (Γραφική Διεπαφή Χρήστη)
- SQL: Structured Query Language
- MySQL: Σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων
- Client: Πελάτης
- Server: Εξυπηρετητής
- Client--Server: Πελάτης--Εξυπηρετητής
- DB: Βάση Δεδομένων
- UI: User Interface
- UML: Unified Modeling Language
- API: Application Programming Interface

1.4 Τυπογραφικές παραδοχές του εγγράφου

Το κείμενο του παρόντος εγγράφου είναι γραμμένο με γραμματοσειρά Baskerville, μεγέθους 11pt και διάστιχο 1.15. Οι επικεφαλίδες του εγγράφου έχουν μέγεθος 14pt και ο τίτλος κάθε κεφαλαίο 13pt σε γραμματοσειρά Naxos. Οι απαιτήσεις στο κεφάλαιο 2 ονομάζονται και αριθμούνται κατάλληλα και, επίσης, συντάσσονται με την αρμόζουσα μορφή.

1.5 Στόχοι Σχεδίασης

Στόχος της σχεδίασης του συστήματος είναι η ικανοποίηση των τριών διαφορετικών ομάδων φυσικών προσώπων που σχετίζονται με αυτό: του πελάτη, του τελικού χρήστη και του προγραμματιστή

Πελάτης του συστήματος μπορεί να θεωρηθεί ο ιδιοκτήτης μιας εταιρείας ανάπτυξης, διαχείρισης και πώλησης εφαρμογών ή και κάποιο γραφείο ευρέσεως εργασίας. Όπως είναι λογικό, ο πελάτης, επιθυμεί το προϊόν να έχει χαμηλό κόστος, συμβατότητα, , να παρακολουθεί τις λειτουργικές και μη απαιτήσεις που έχουν τεθεί και η ανάπτυξη του να είναι όσο το δυνατόν πιο γρήγορη.

Ο τελικός χρήστης επιθυμεί το σύστημα να είναι χρηστικό, φιλικό προς αυτόν και η διαδικασία εκμάθησής του να είναι απλή, εύκολη και γρήγορη. Ακόμα, θέλει η λειτουργία του συστήματος να είναι ευσταθής, να ανέχεται σφάλματα και συνεπώς να είναι αποδοτική.

Επίσης, ο προγραμματιστής του συστήματος, (ή ο συντηρητής) έχει την απαίτηση, να μην παρουσιάζονται σφάλματα στο σύστημα, να προσαρμόζεται εύκολα και γρήγορα σε μεταβολές, τροποποιήσεις και αναβαθμίσεις λογισμικού και ακόμα ο διαχωρισμός των υποσυστημάτων να είναι σωστός και ορισμένος με σαφήνεια.

1.6 Αναγνωστικό κοινό και τρόπος ανάγνωσης

Το έγγραφο αυτό γράφτηκε για συγκεκριμένες ομάδες ανθρώπων προκειμένου να μελετηθούν και να σχεδιαστούν τα χαρακτηριστικά του συστήματος και στη συνέχεια να γίνει ο προγραμματισμός και η υλοποίηση της εφαρμογής. Οι βασικοί αναγνώστες του συγκεκριμένου εγγράφου θα είναι:

- Προϊστάμενοι καθώς και ορισμένοι αρμόδιοι μηχανικοί λογισμικού των εταιριών ανάπτυξης ανάλογων εφαρμογών
- Προγραμματιστές που θα αναλάβουν τη συγγραφή του κώδικα ο οποίος θα υλοποιεί το σύστημα
- Μηχανικοί λογισμικού που θα αναλάβουν τη συντήρηση του συστήματος.
- Μηχανικοί υλικού που θα αναλάβουν το σχεδιασμό και την εγκατάσταση των συστημάτων που ικανοποιούν τις τεχνολογικές απαιτήσεις του έργου

Αρχικά, είναι σημαντικό να αναφερθεί ότι για την ευκολότερη κατανόηση και αποτελεσματική ανάγνωση του εγγράφου αποτελεί σημαντική προϋπόθεση να έχει προηγηθεί η ανάγνωση των δύο προηγούμενων εγγράφων, αυτού των απαιτήσεων χρηστών και αυτού των απαιτήσεων λογισμικού.

Επιπροσθέτως, η ανάγνωση των κεφαλαίων θα πρέπει να γίνει με τη σειρά που υπάρχουν στο έγγραφο για να μη δημιουργηθούν προβλήματα στην κατανόηση του.

1.7 Επισκόπηση Εγγράφου

- **ΚΕΦΑΛΑΙΟ 1:** Το κεφάλαιο αυτό αποτελεί μια εισαγωγή που περιλαμβάνει γενικές πληροφορίες για το περιεχόμενο του εγγράφου καθώς και για τα κεφάλαια που ακολουθούν.
- **ΚΕΦΑΛΑΙΟ 2:** Στο κεφάλαιο αυτό παρουσιάζεται η δυναμική μοντελοποίηση του συστήματος Robobar
- **ΚΕΦΑΛΑΙΟ 3:** Στο κεφάλαιο αυτό γίνεται αναλυτική περιγραφή και παρουσίαση της αρχιτεκτονικής του συστήματος .
- **ΚΕΦΑΛΑΙΟ 4:** Στο τέταρτο κεφάλαιο γίνεται η αποδόμηση του συστήματος σε υποσυστήματα, σύμφωνα με την αρχιτεκτονική που επιλέχθηκε και περιγράφηκε στο τρίτο κεφάλαιο. Επιπλέον περιλαμβάνει λεπτομερή διαγράμματα της διασύνδεσης μεταξύ των υποσυστημάτων, ώστε να δοθεί στον

αναγνώστη μια πλήρη και παραστατική εικόνα για την αρχιτεκτονική του συστήματος. Τέλος αναλύονται θέματα γενικού ελέγχου, ασφάλειας, πρόσβασης και οριακών συνθηκών στα οποία πρέπει να δοθεί ιδιαίτερη προσοχή.

2 Δυναμική Μοντελοποίηση του Συστήματος

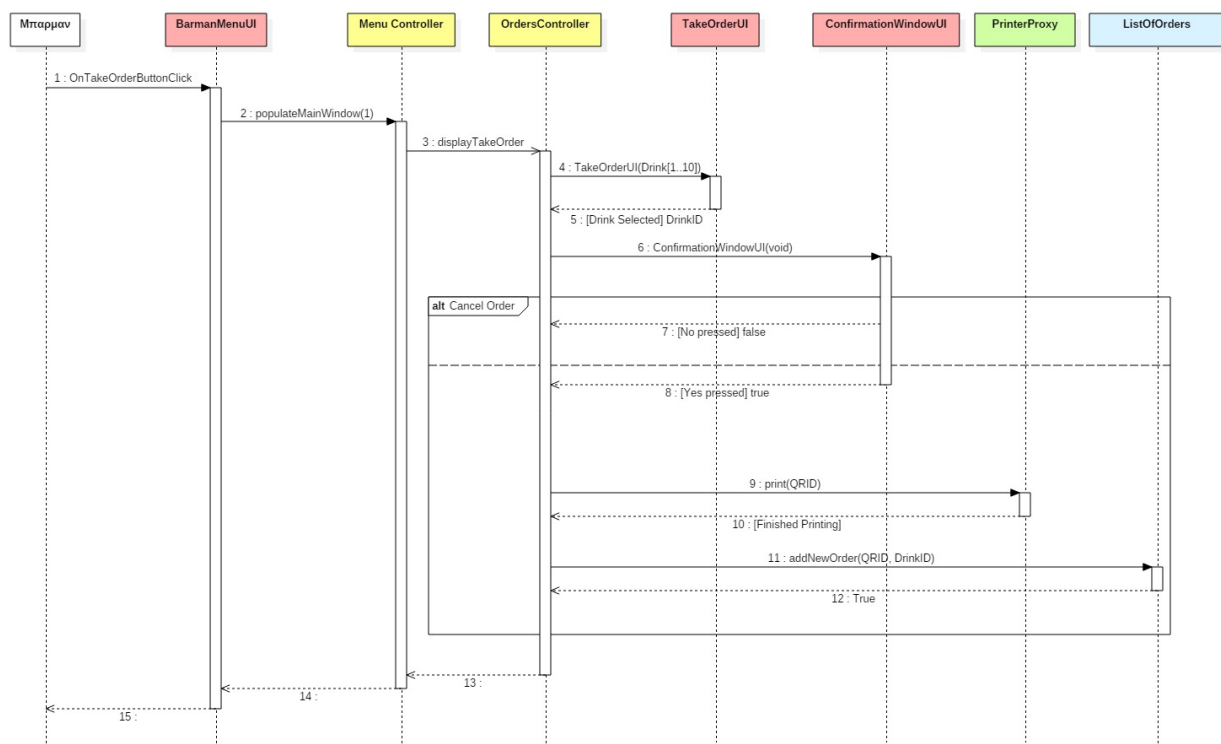
2.1 Σενάριο Χρήσης 3, Παραγγελία ποτού από τον Μπάρμαν

Ορισμός: Σε αυτό το Σενάριο Χρήσης, ο Μπάρμαν επιλέγει το ποτό της παραγγελίας του πελάτη και τελικά λαμβάνει εκτυπωμένο το Qrcode της παραγγελίας και την αποθήκευση της στην λίστα παραγγελιών προς εκτέλεση.

Περιγραφή: Ο Μπάρμαν είναι συνδεδεμένος στο σύστημα. Από το BarmanMenuUI επιλέγει τον σύνδεσμο νέας παραγγελίας, και καλείται η populateMainWindow(1) του ελεγκτή MenuController. Από εκεί ο OrdersController εμφανίζει την TakeOrderUI και αυτή με την σειρά της την SelectDrinkUI. Ο Μπάρμαν επιλέγει το ποτό που του έχει ζητήσει ο πελάτης και πατώντας το κουμπι επιβεβαίωσης, η OnDrinkClick() επιστρέφει στον ελεγκτή το DrinkID του επιθυμητού ποτού.

Με το DrinkID, ο OrdersController καλεί την createOrder και στη συνέχεια γίνεται εμφάνιση του ConfirmationWindowUI, στο οποίο ο Μπάρμαν καλείται να επιλέξει αν θέλει να επιβεβαιώσει την παραγγελία του ποτού. Στην περίπτωση που την απορίψει, η OnNoClick επιστρέφει false στον ελεγκτή, εμφανίζεται το κατάλληλο μήνυμα και ο Μπάρμαν καλείται να ξαναεπιλέξει ένα από τα διαθέσιμα ποτά.

Στην περίπτωση που ο Μπάρμαν πατήσει αποδοχή στην παραγγελία, η OnYesClick επιστρέφει true, ο OrderController εκτελεί τις συναρτήσεις randQR() η οποία δημιουργεί ένα τυχαίο string χαρακτήρων και την stringToQR() η οποία μετατρέπει τους χαρακτήρες σε QR code. Στην συνέχεια καλείται η PrinterProxy, η οποία εκτελεί την print για να εκτυπώσει τον Qrcode και τέλος η addNewOrder προσθέτει το DrinkID στην λίστα του ListofOrders για την μετέπειτα εκτέλεση της κάθε παραγγελίας.



2.2 Σενάριο Χρήσης 6, Παρασκευή ποτού-Παραλαβή ποτού με εισιτήριο

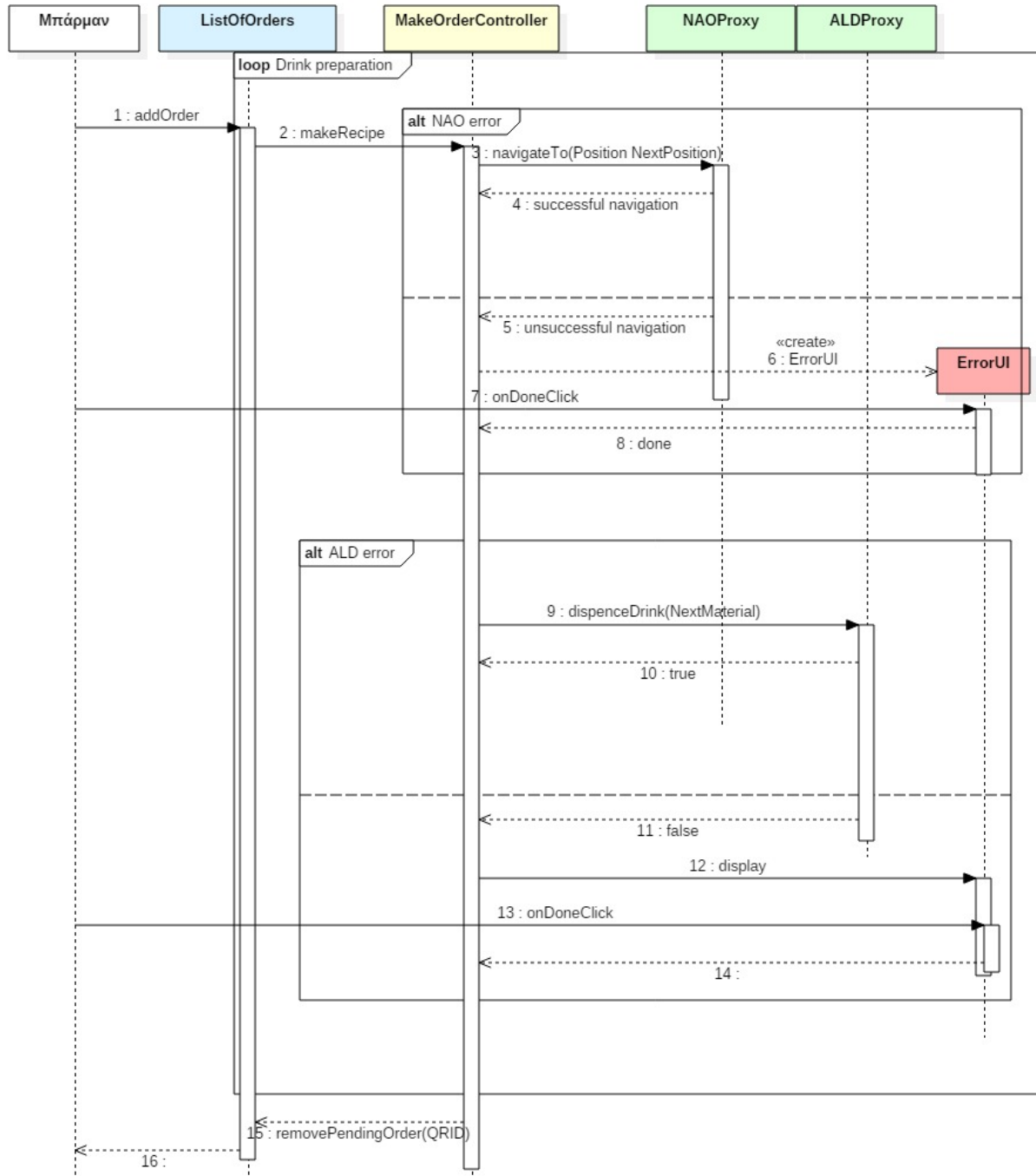
Ορισμός: Όπως αναφέρθηκε και στο πρώτο παραδοτέο, το 6ο Σενάριο Χρήσης περιλαμβάνει την διαδικασία που ακολουθεί το σύστημα δίνοντας εντολές στο NAO για την σωστή κίνηση καθώς και στο ALD για την εξαγωγή της σωστής ποσότητας ποτού.

Περιγραφή: Από την addOrder της ListOfOrders, η τελευταία περιέχει πλέον παραγγελίες. Για την εκτέλεση κάθε παραγγελίας υπάρχει ένας βρόχος, ο οποίος ξεκινά με την κλήση της makeRecipe του ελεγ-

κτή MakeOrderController (η getRecipe έχει εκτελεστεί ήδη και έχουμε λάβει την συνταγή από την βάση δεδομένων των συνταγών).

Στην συνέχεια καλείται η NavigateTo(Position NextPosition) της NAOProxy. Αν υπάρξει κάποιο error στην διαδικασία καλείται η ErrorUI και εμφανίζει στον Μπάρμαν ένα κουτί διαλόγου όπου καλείται να πατήσει “OK”. Του γνωστοποιεί δηλαδή ότι υπήρξε πρόβλημα. Έπειτα καλείται η dispenseDrink(NextMaterial) του ALDProxy κατά την οποία το NAO γεμίζει το ποτήρι με την προκαθορισμένη ποσότητα του κάθε υλικού που περιλαμβάνει η συνταγή. Όπως και στην προηγούμενη περίπτωση, σε περίπτωση σφάλματος καλείται η ErrorUI. Τα δύο αυτά βήματα επαναλαμβάνονται μέχρι να ολοκληρωθεί η συνταγή.

Τέλος έχουμε αφαίρεση της παραγγελίας από την ListOfOrders με την removePendingOrder(QRID).

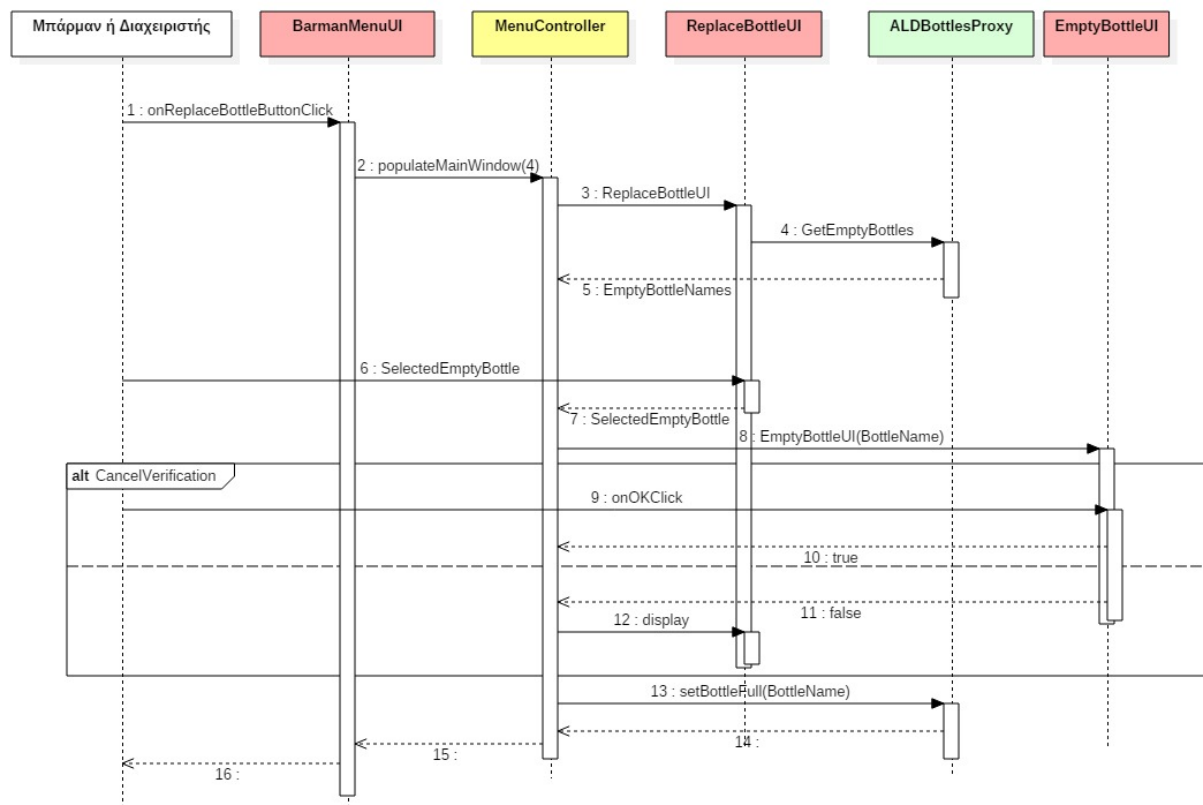


2.3 Σενάριο Χρήσης 8, Επιλογή μπουκαλιού ως γεμάτο

Ορισμός: Στο παρακάτω σενάριο, ο χρήστης (Μπάρμαν ή Διαχειριστής) αφού αντικαταστήσουν ένα μπουκάλι που έχει αδειάσει, επιλέγουν στο σύστημ το μπουκάλι που αντικαταστάθηκε ως γεμάτο.

Περιγραφή: Ο χρήστης είναι συνδεδεμένος στο σύστημα. Από το BarmanMenuUI επιλέγει το κουμπί ενημέρωσης μπουκαλιού και εκτελείται η `onReplaceBottleButtonClick()` η οποία καλεί την `populateMainWindow(4)` της MenuController. Στην συνέχεια εμφανίζεται η ReplaceBottleUI. Εκεί ο χρήστης καλείται να διαλέξει ποιο από τα άδεια μπουκάλια έχει αντικαταστήσει. Οι πληροφορίες για τα άδεια μπουκάλια λαμβάνονται από την `GetEmptyBottles` της ALDBottlesProxy. Όταν επιλεγθεί το μπουκάλι, καλείται η `selectedEmptyBottle()`.

Ο ελεγκτής εμφανίζει την EmptyBottleUI, στην οποία ο χρήστης πρέπει να επιβεβαιώσει την επιλογή του. Σε περίπτωση που επιλέξει “cancel” ξαναεμφανίζεται η λίστα με τα άδεια μπουκάλια. Αν επιλέξει επιβεβαίωση, καλείται η `setBottleFull` της ALDBottlesProxy και ενημερώνεται πως το μπουκάλι δεν είναι πλέον άδειο.



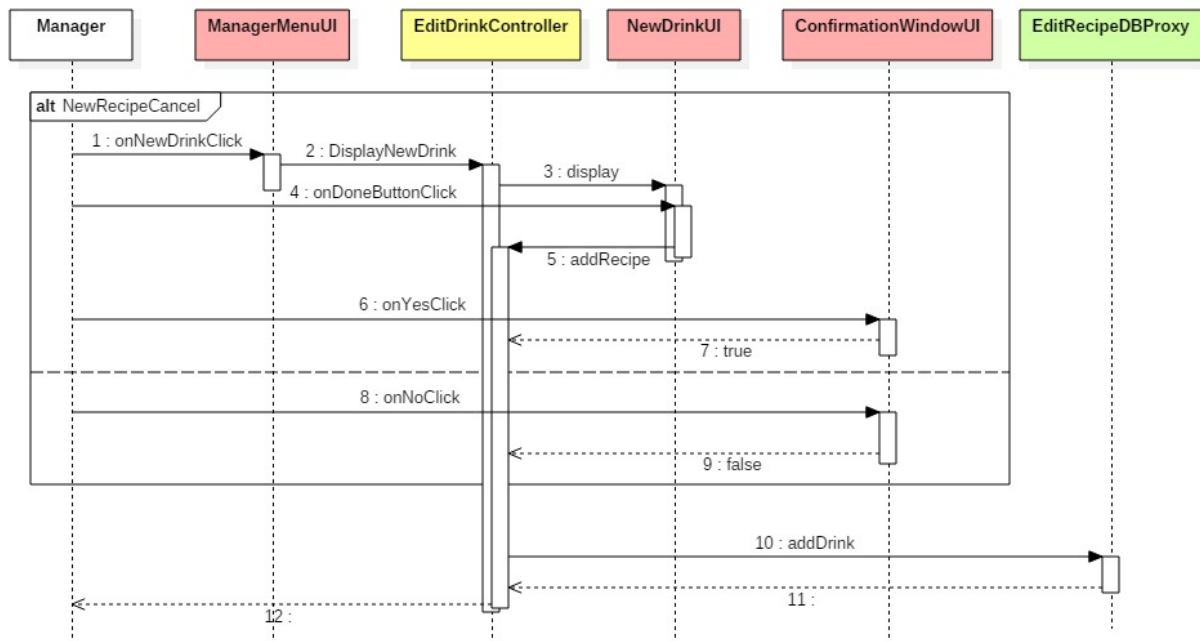
2.4 Σενάριο Χρήσης 9, Προσθήκη συνταγής

Ορισμός: Ο Διαχειριστής προσθέτει μια καινούρια συνταγή στην λίστα των διαθέσιμων συνταγών, επιλέγοντας το είδος και την ποσότητα των ποτών που περιέχουν.

Περιγραφή: Ο Διαχειριστής είναι συνδεδεμένος στο σύστημα. Στην συνέχεια πατάει το κουμπί για την δημιουργία καινούριας συνταγής ποτού με το οποίο καλείται η `display()` του NewDrinkUI. Συμπληρώνει τα απαραίτητα στοιχεία του καινούριου ποτού στο NewDrinkUI, δηλαδή το DrinkName για το όνομα του ποτού, διαλέγει από ποιά μπουκάλια περιέχει ποτό (Bottles: dropdown[0..10]) και την ποσότητα ποτού που θα έχει από κάθε μπουκάλι (Quantity: textbox[0..10]).

Με το πάτημα του κουμπιού `onDoneButtonClick()` εκτελεί την `addRecipe()` του EditDrinkController. Στην συνέχεια καλείται η ConfirmationWindowUI και ο χρήστης πατάει τον σύνδεσμο αποδοχής της καινούριας συνταγής. Έτσι, ο ελεγκτής λαμβάνει true από την ConfirmationWindowUI και με την σειρά του καλεί την `addDrink` του EditRecipeDBProxy.

Σε περίπτωση που δεν πατήσει το κουμπί αποδοχής αλλά το κουμπί cancel, η ConfirmationWindowUI επιστρέφει false στον ελεγκτή και εμφανίζεται μήνυμα όπου ενημερώνει τον χρήστη ότι επιλέγει την έξοδο από την φόρμα.



3 Προτεινόμενη Αρχιτεκτονική Λογισμικού

3.1 Αρχιτεκτονική πελάτη-διακομιστή (client-server)

Το πρότυπο αρχιτεκτονικής λογισμικού πελάτη-διακομιστή αποτελεί μία συνήθη μέθοδο ανάπτυξης λογισμικού. Πρόκειται για μία αρχιτεκτονική η οποία αποτελείται από έναν ή περισσότερους διακομιστές (servers) οι οποίοι παρέχουν ορισμένες υπηρεσίες και από τον πελάτη(client),ο οποίος χρησιμοποιεί τις υπηρεσίες που παρέχουν οι servers.

Στην εν λόγω αρχιτεκτονική, ο πελάτης χρησιμοποιεί τις υπηρεσίες του λογισμικού μέσω των γραφικών διεπαφών δημιουργώντας αιτήσεις για πληροφορίες οι οποίες στέλνονται στο διακομιστή. Ο διακομιστής ελέγχει το αίτημα και στέλνει πίσω τις κατάλληλες πληροφορίες. Ένας διακομιστής έχει την δυνατότητα να δέχεται ταυτόχρονα αιτήματα από πολλούς πελάτες. Τα δεδομένα διατηρούνται σε ένα κεντρικό σύστημα ανεξάρτητο από πού χρησιμοποιούν οι πελάτες. Ένας πελάτης γνωρίζει τη διεπαφή του διακομιστή, ενώ το αντίστροφο δεν είναι απαραίτητο. Επιπλέον, το σύστημα που χρησιμοποιεί ο πελάτης δεν χρειάζεται να είναι ισχυρό αφού οι κύριες λειτουργίες εκτελούνται στο διακομιστή. Η αρχιτεκτονική αυτή παρέχει ασφάλεια και έλεγχο του δικτύου μέσω της κεντρικής διαχείρισης των διακομιστών.

Η αρχιτεκτονική πελάτη-διακομιστή είναι ουσιαστικά μια αρχιτεκτονική 2 επίπεδων. Η αρχιτεκτονική επίπεδων είναι μια τεχνική που μας βοηθάει να μειώσουμε την πολυπλοκότητα και να επιτύχουμε χαμηλή σύζευξη. Μέσω της αρχιτεκτονικής επίπεδων μπορούμε να διαμοιράσουμε τις βασικές λειτουργίες του συστήματος σε ξεχωριστά επίπεδα κατάλληλα ιεραρχημένα, έτσι ώστε τα χαμηλότερα επίπεδα να μην έχουν γνώση των υψηλότερων και να μην εξαρτώνται από αυτά, ενώ τα υψηλότερα επίπεδα να εξαρτώνται από τα χαμηλότερα

3.2 Αρχιτεκτονική Συστήματος Robobar

Με βάση την παραπάνω ανάλυση, το λογισμικό RoboBar θα ακολουθήσει ανάπτυξη αρχιτεκτονικής Client/Server. Όπως αναφέρθηκε προηγούμενος η αρχιτεκτονική αυτή αποτελείται από 2 επίπεδα.

Το πρώτο επίπεδο είναι ο server στον οποίο θα βρίσκεται εγκατεστημένο το σύστημα όπως επίσης θα βρίσκονται αποθηκευμένες και οι βάσεις δεδομένων του συστήματός μας. Κάθε ενέργεια του συστήματος που αφορά στους διακομιστές συμβαίνει σε αυτό το επίπεδο.

Το δεύτερο επίπεδο (Client Tier) θα περιλαμβάνει όλες τις διεπαφές και τις γραφικές διεπαφές που θα προβάλλονται στον εκάστοτε χρήστη και μέσω των οποίων αυτός θα έχει πρόσβαση στο σύστημα.

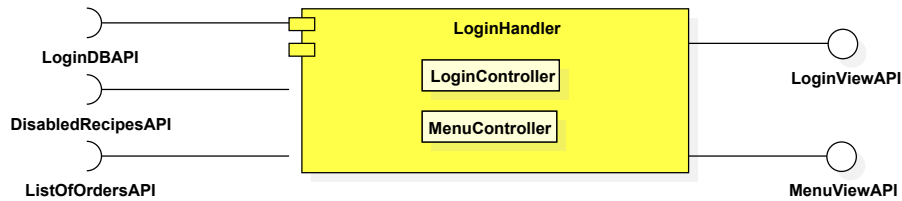
Μέσω της συσκευής τους, οι χρήστες θα δίνουν τα απαραίτητα δεδομένα για να εισέλθουν στο σύστημα και να κάνουν τις ενέργειες που χρειάζεται ώστε να χρησιμοποιήσουν τις λειτουργίες του συστήματος.

4 Αποδόμηση του Συστήματος

4.1 Υποσυστήματα

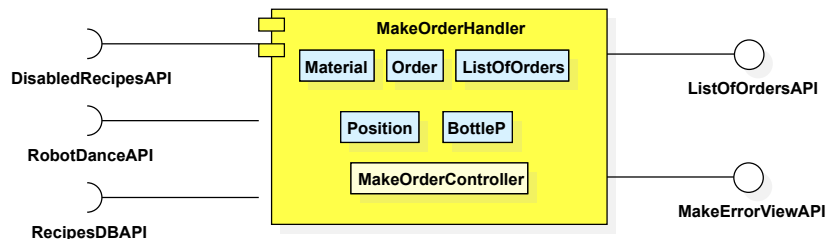
Πρόκειται για το διαχωρισμό του συστήματος σε επιμέρους υποσυστήματα και τμήματα, τα οποία φέρουν ξεχωριστές ευθύνες και λειτουργίες. Στην παράγραφο που ακολουθεί ορίζονται τα υποσυστήματα με τα τμήματα που εμπεριέχουν και αναλύεται η λειτουργικότητα καθενός από αυτά.

4.1.1 Υποσύστημα LoginHandler



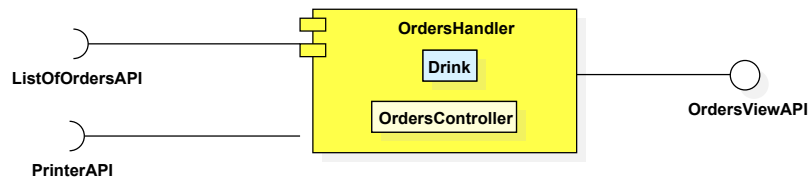
Το παρόν υποσύστημα ελέγχει τα στοιχεία εισόδου των χρηστών και εμφανίζει τα κατάλληλα μενού για την εκπλήρωση των διάφορων επιτρεπόμενων εργασιών. Απαιτεί τις διεπαφές LoginDBAPI από το υποσύστημα databases για την εξακρίβωση των στοιχείων σύνδεσης και την ανάγνωση της λίστας διαθέσιμων ποτών για παρασκευή, DisabledRecipesAPI από το υποσύστημα EmptyBottles για την ανάγνωση της λίστας μη διαθέσιμων ποτών για παρασκευή λόγω άδειων μπουκαλιών, και ListOfOrdersAPI από το υποσύστημα Orders για την μεταφορά εντολών έναρξης διαδικασιών όπου αφορούν τις παραγγελίες. Παρέχει την διεπαφή LoginViewAPI για την εμφάνιση των GUI όπου αφορούν την είσοδο του Μπάρμαν και του Διαχειριστή, όπως και την MenuViewAPI για την εμφάνιση του Background γραφικού, όπως και του δεξιού μενού.

4.1.2 Υποσύστημα MakeOrderHandler



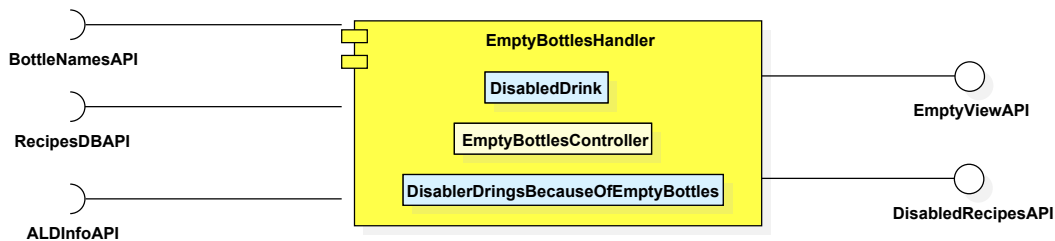
Το παρόν υποσύστημα διαβάζει μια λίστα από εκκρεμής παραγγελίες, και τις εκτελεί μία μία. Σε περίπτωση λάθους, ενημερώνει τον χρήστη για τις ενέργειες που πρέπει να προβεί ώστε να συνεχιστεί η παραγωγή. Απαιτεί τις διεπαφές RecipesDBAPI από το υποσύστημα Databases για την ανάγνωση συνταγής για την παρασκευή ενός ποτού, RobotDanceAPI από το υποσύστημα robots για την επικοινωνία με τα εξωτερικά συστήματα ρομπότ με στόχο την μετακίνηση ποτηριών, εξαγωγή ποτού με την κατάλληλη σειρά και τέλος την παράδοση του ποτού στον πελάτη όπου το παρήγγειλε, και DisabledRecipesAPI από το υποσύστημα EmptyBottles για την ανάγνωση της λίστας μη διαθέσιμων ποτών για παρασκευή λόγω άδειων μπουκαλιών. Παρέχει την διεπαφή ListOfOrdersAPI όπου επιτρέπει την προσθαφαίρεση παραγγελιών από την λίστα εκκρεμών παραγγελιών και MakeErrorViewAPI για την εμφάνιση κατάλληλου μηνύματος σφάλματος όταν αναγνωριστεί κάποιο πρόβλημα με την παρασκευή του ποτού.

4.1.3 Orders



Το παρόν υποσύστημα εμφανίζει στον χρήστη τις κατάλληλες γραφικές διεπαφές ώστε να προστεθεί μια νέα παραγγελία στην λίστα εκκρεμών παραγγελιών του MakeOrderHandler, να διαγραφεί ή να ενημερωθεί. Απαιτεί την διεπαφή ListOfOrdersAPI από το υποσύστημα MakeOrderHandler για την προσθαφαίρεση παραγγελιών βάσει των επιλογών του Μπάρμαν, και την διεπαφή PrinterAPI από το υποσύστημα Printer για την εκτύπωση του αποδεικτικού του πελάτη για την παραλαβή του ποτού του. Παρέχει την διεπαφή OrdersViewAPI για την εμφάνιση των GUIs που αφορούν την προσθαφαίρεση παραγγελιών.

4.1.4 EmptyBottlesHandler



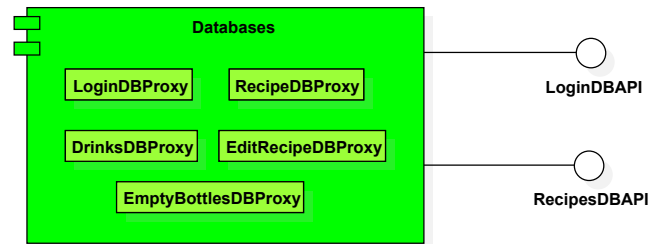
Το παρόν υποσύστημα διαχειρίζεται το έκτακτο άδειασμα ενός μπουκαλιού με την ενημέρωση εσωτερικών λιστών ποιων μπουκαλιών είναι άδειων και ποιων ποτών δεν είναι δυνατό να παρασκευαστούν λόγω αυτής της έλλειψης, όπως και την ενημέρωση του χρήστη για αυτήν την έλλειψη. Απαιτεί τις διεπαφές RecipesDBAPI από το υποσύστημα Databases για την ανάγνωση δεδομένων των ποτών και εγκατεστημένων μπουκαλιών, και ALDInfoAPI από το υποσύστημα Robots για τον έλεγχο αν υπάρχει κάποιο νέο άδειο μπουκάλι. Παρέχει τις διεπαφές DisabledRecipesAPI όπου προσφέρει μια λίστα με ποτά όπου δεν μπορούν αυτή την στιγμή να παρασκευαστούν λόγω ενός άδειου μπουκαλιού και EmptyViewAPI για την εμφάνιση των παραθύρων ενημέρωσης άδειου μπουκαλιού και αντικατάστασης μπουκαλιού.

4.1.5 EditDrinksHandler



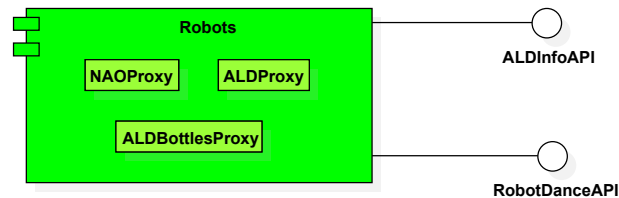
Το παρόν υποσύστημα εμφανίζει στον χρήστη τις απαραίτητες γραφικές διεπαφές για την προσθαφαίρεση συνταγών ποτών όπου μπορούν να παραγγελθούν (και παρασκευαστούν) όπως και την προσθήκη νέων μπουκαλιών για την παρασκευή νέων συνταγών. Απαιτεί την διεπαφή RecipesDBAPI από το υποσύστημα Databases για την ανταλλαγή πληροφοριών των υπαρχουσών και νέων συνταγών ποτών. Παρέχει τις διεπαφές BottleNamesAPI για την ενημέρωση των ονομάτων των εγκατεστημένων μπουκαλιών και EditDrinksViewAPI για την εμφάνιση των GUIs που αφορούν την προσθαφαίρεση συνταγών ποτών και μπουκαλιών από το σύστημα.

4.1.6 Database



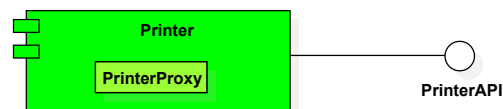
Το παρόν υποσύστημα, παρέχει τις απαραίτητες διασυνδέσεις για την επικοινωνία των επιθυμητών δεδομένων από το σύστημα βάσης δεδομένων και τα υποσυστήματα του δεύτερου επιπέδου. Δεν απαιτεί κάποια διεπαφή, και παρέχει τις διεπαφές LoginDBAPI για τον έλεγχο των στοιχείων εισόδου και την RecipesDBAPI για την ανάγνωση και εγγραφή συνταγών ποτών και εγκατεστημένων μπουκαλιών.

4.1.7 Robots



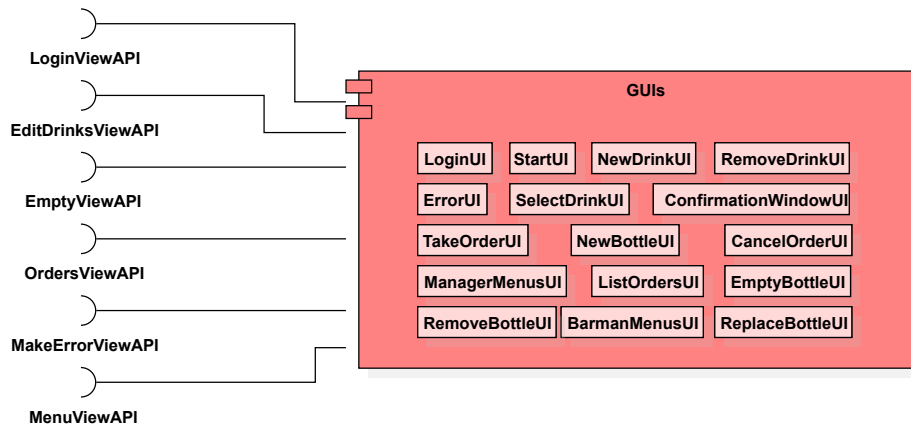
Το παρόν υποσύστημα παρέχει τις απαραίτητες διασυνδέσεις για την επικοινωνία μιας γκάμας από εντολές στα ρομπότ εξωτερικά συστήματα όπου είναι συμβατό. Η επικοινωνία είναι αμφίδρομη, επιτρέποντας και την ανάκτηση πληροφοριών από τα εξωτερικά συστήματα αυτά. Δεν απαιτεί κάποια διεπαφή, και παρέχει τις διεπαφές RobotDanceAPI για την επικοινωνία με τα εξωτερικά συστήματα ρομπότ με στόχο την μετακίνηση ποτηριών, εξαγωγή ποτού και την χρήση της διαθέσιμης κάμερας, και ALDInfoAPI για την ενημέρωση της κατάστασης των μπουκαλιών όπου είναι εγκατεστημένα στο ALD.

4.1.8 Printer



Το παρόν υποσύστημα, πολύ απλά παρέχει την δυνατότητα εκτύπωσης μιας εικόνας στον διαθέσιμο εκτυπωτή. Δεν απαιτεί κάποια διεπαφή, και παρέχει μια διεπαφή PrinterAPI για την εκτύπωση του αποδεικτικού παραγγελίας.

4.1.9 GUIs



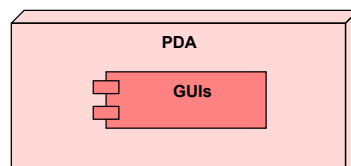
Το παρόν υποσύστημα παρέχει μια ενοποιημένη διασύνδεση με το γραφικό περιβάλλον του συστήματος προς όλα τα υπόλοιπα υποσυστήματα. Απαιτεί τις διεπαφές LoginViewAPI, MenuViewAPI, MakeErrorViewAPI, OrdersViewAPI, EmptyViewAPI, και EditDrinksViewAPI για την αμφίδρομη ανταλλαγή εντολών εμφάνισης γραφικών διεπαφών και πραγματοποίησης ενεργειών όπου προέρχονται από την είσοδο του Μπάρμαν και του Διαχειριστή.

4.2 Απεικόνιση Υλικού / Λογισμικού

Στη συνέχεια θα παρουσιαστούν με τα κατάλληλα διαγράμματα UML η απεικόνιση Υλικού/Λογισμικού. Πιο συγκεκριμένα, σκοπός αυτής της παραγράφου είναι η κατανόηση του τρόπου σύνδεσης των τμημάτων του συστήματος, καθώς και το πως συσχετίζονται τα τμήματα αυτά με το υλικό του συστήματος.

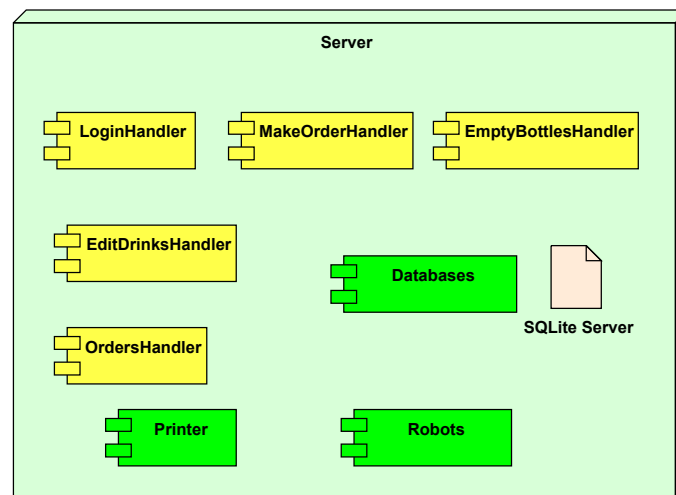
4.2.1 Client Device (PDA)

Όλη η αλληλεπίδραση με τον Μπάρμαν και τον Διαχειριστή πραγματοποιείται μέσω των γραφικών διεπαφών όπου εμφανίζονται σε μια συσκευή PDA. Με αυτόν τον τρόπο αυτοί οι χρήστες έχουν την δυνατότητα να είναι σε κίνηση χωρίς να δεσμεύονται σε ένα σημείο. Η επικοινωνία μεταξύ του client και του server γίνεται μέσω εντολών HTTPS οπότε υπάρχει προϋπόθεση να είναι συνδεδεμένα στο ίδιο δίκτυο. Η γραφικές αυτές διεπαφές είναι δυνατό να εμφανιστούν σε οποιαδήποτε συσκευή μέσω του web browser της συμπεριλαμβανομένου του μηχανήματος όπου είναι εγκατεστημένο το server side κομμάτι του συστήματος. Οπότε αν και οι διεπαφές είναι βελτιστοποιημένες για μικρή οθόνη αφής όπως έχει αναφερθεί στο έγγραφο απαιτήσεων χρηστών, υπάρχει και η δυνατότητα το σύστημα να λειτουργήσει εξολοκλήρου σε ένα μηχάνημα.



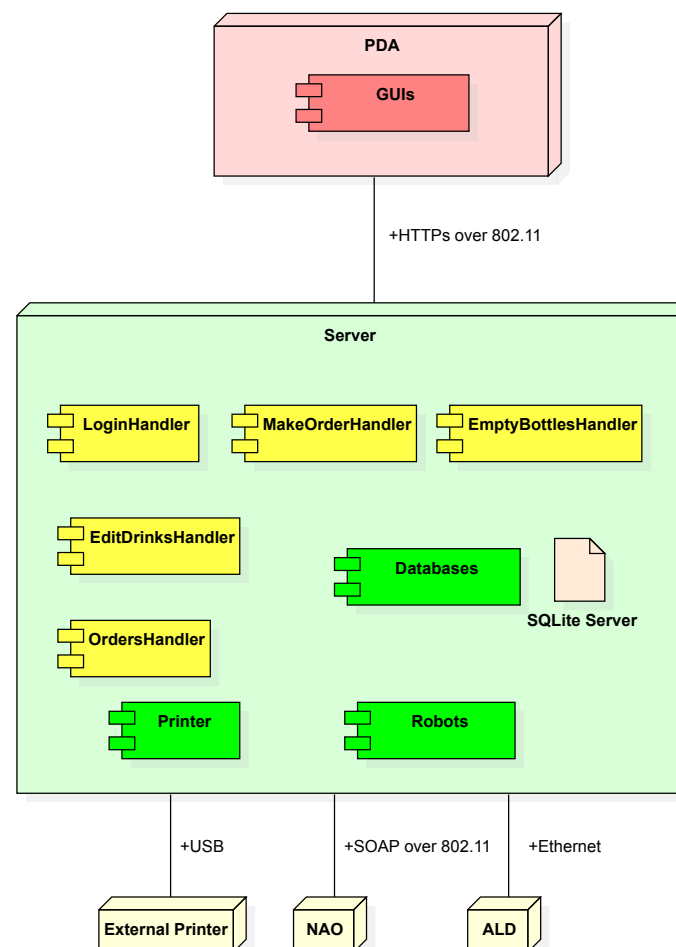
4.2.2 System server

Όλα τα υπόλοιπα υποσυστήματα τρέχουν σε ένα μηχάνημα server όπου βρίσκεται στον χώρο του μπαρ. Είναι συνδεδεμένο με το τοπικό δίκτυο του χώρου, όπου πρέπει να περιέχει και ένα access point για να είναι εφικτή η επικοινωνία με τα PDAs και το NAO. Επίσης, είναι συνδεδεμένο με τον εκτυπωτή αποδείξεων μέσω USB. Η βάση δεδομένων όπου περιέχει τα login στοιχεία όπως και τις συνταγές λόγω του μικρού της μεγέθους θα τρέχει ως service στο ίδιο το μηχάνημα.



4.2.3 Συνολικό Διάγραμμα Ανάπτυξης

Οι παραπάνω δύο κόμβοι συνενώνονται για να δημιουργήσουν το σύστημα Robobar. Όπως αναφέρθηκε, ο κύριος διάλογος επικοινωνίας είναι ένα τοπικό δίκτυο στον χώρο όπου διαθέτει τουλάχιστον ένα access point. Με αυτόν τον τρόπο, το PDA και το εξωτερικό σύστημα NAO αποκτούν ασύρματη πρόσβαση στο δίκτυο όπου εμπεριέχεται ο server και το ALD εξωτερικό σύστημα. Η σύνδεση του εκτυπωτή επιλέχτηκε να γίνει με USB για λόγους συμβατότητας για την πλειοψηφία των μοντέλων θερμικών εκτυπωτών.



Τέλος, παρουσιάζονται παρακάτω οι προτεινόμενες απαιτήσεις των μηχανημάτων του συστήματος

PDA	
Model	iPad mini 4
Capacity	base 128GB
Connectivity	WiFi only

Server	
Model	Mac mini
CPU	2.6GHz dual-core Intel Core i5
RAM	8GB LPDDR3 SDRAM
Storage	256GB PCIe-based Flash Storage

Printer	
Model	Epson C31CA85084
Maximum Print Width	2.83"
QRCode Support	YES
USB Support	YES

4.3 Διαχείριση Μόνιμων Δεδομένων

Η διαχείριση μόνιμων δεδομένων αφορά τα δεδομένα που είναι απαραίτητα για την λειτουργία της εφαρμογής μας και τον τρόπο με τον οποίο αυτά αποθηκεύονται και προσπελάζονται.

Πιο συγκεκριμένα, αυτό υλοποιείται με την χρήση βάσεων δεδομένων λόγω της εύκολης ανάκτησης δεδομένων μέσω αναζήτησης κατ' απαίτηση. Το σύστημά μας επικοινωνεί με 4 εξωτερικές βάσεις δεδομένων μέσω κατάλληλων κλάσεων διεπαφών. Ειδικότερα επικοινωνεί με τις εξής βάσεις δεδομένων:

- **LoginDBProxy:** Η κλάση αυτή επικοινωνεί με την βάση δεδομένων των χρηστών. Σε αυτήν βρίσκονται τα στοιχεία των δύο διαφορετικών χρηστών που υποστηρίζει το σύστημα και χρησιμοποιείται για την αντιστοίχιση κάθε χρήστη στο ανάλογο μενού μετά την σύνδεσή του στο σύστημα.
- **RecipeDBProxy:** Η κλάση αυτή χρησιμοποιείται για την επικοινωνία του συστήματος με την βάση δεδομένων των ποτών που είναι διαθέσιμα και χρησιμοποιείται από το εξωτερικό σύστημα ΝΑΟ όταν αυτό πάρει εντολή για την παρασκευή ενός ποτού.
- **DrinksDBProxy:** Η κλάση αυτή χρησιμοποιείται για την επικοινωνία του συστήματος με την βάση δεδομένων των μπουκαλιών που είναι διαθέσιμα κάθε δεδομένη στιγμή και χρησιμοποιείται από το εξωτερικό σύστημα ΝΑΟ για την παρασκευή των ποτών αλλά και από τους χρήστες για έλεγχο διαθεσιμότητας.
- **EmptyBottlesDBProxy:** Η κλάση αυτή χρησιμοποιείται για την επικοινωνία του συστήματος με την βάση δεδομένων των άδειων μπουκαλιών. Χρησιμοποιείται από τον διαχειριστή έτσι ώστε να ενημερώνεται όταν κάποιο μπουκάλι έχει αδειάσει έτσι ώστε να προχωρήσει στην αντικατάστασή του.
- **EditRecipeDBProxy:** Η κλάση αυτή χρησιμοποιείται για την επικοινωνία του συστήματος με την βάση δεδομένων των συνταγών των ποτών. Χρησιμοποιείται από το εξωτερικό σύστημα ΝΑΟ έτσι ώστε να γνωρίζει πώς να φτιάξει κάθε ποτό, από τον μπάρμαν έτσι ώστε να βλέπει τις διαθέσιμες και να μπορεί να πάρει παραγγελία και από τον διαχειριστή ο οποίος επιπλέον μπορεί να προσθαφαιρέσει συνταγές κατά βούληση.

4.4 Έλεγχος πρόσβασης και Ασφάλεια

Στο σύστημά μας ο έλεγχος πρόσβασης γίνεται με την επιβεβαίωση μοναδικού κωδικού για κάθε χρήστη, την οποία ο χρήστης την εισάγει στην αρχική σελίδα σύνδεσης και εφ' όσον είναι σωστός ανακατευθύνεται στο αντίστοιχο μενού. Σε περίπτωση που δοθεί κωδικός που δεν αντιστοιχίζεται σε χρήστη τότε το σύστημα εμφανίζει μήνυμα λάθους και η σύνδεση απορρίπτεται. Οι κωδικοί με την αντιστοίχισή τους στον κατάλληλο χρήστη βρίσκονται αποθηκευμένα στην LoginDBProxy.

Το σύστημά μας διαθέτει μόνο 2 χρήστες: τον μπάρμαν και τον διαχειριστή. Έχουν και οι δύο δικαίωμα πρόσβασης στην εφαρμογή με διαφορετικά όμως δικαιώματα και διαφορετικά δεδομένα. Για να εξασφαλιστεί ότι κάθε χρήστης έχει πρόσβαση στα δικαιώματα που του αναλογούν, κατά την σύνδεση ανάλογα με τον κωδικό που έχει εισαχθεί, το σύστημα προβάλει διαφορετικό γραφικό περιβάλλον για τον κάθε ένα με τα αντίστοιχα δικαιώματα.

Η μοντελοποίηση των χρηστών του συστήματος που περιγράφουν η ασφάλεια και ο έλεγχος πρόσβασης παρουσιάζεται με έναν πίνακα πρόσβασης (access matrix), ο οποίος υλοποιείται με δυνατότητες και παρουσιάζεται παρακάτω.

1. Χρήστης Μπάρμαν

(α) StartUI

- StartUI()
- onLoginClick()
- display()

(β) LoginUI

- loginUI()
- onOKClick()
- onCancelClick()
- display()

(γ) EmptyBottleUI

- EmptyBottleUI(BottleName: String)
- onOKClick()
- display()

(δ) TakeOrderUI

- display()

(ε) BarmanMenuUI

- onTakeOrderButtonClick()
- onCancelOrderButtonClick()
- onShowOrderButtonClick()
- onReplaceBottleButtonClick()
- display()

(στ) ReplaceBottleUI

- ReplaceBottleUI()
- getCurrentEmptyBottles()
- selectedEmptyBottle()
- display()

(ζ) ConfirmationWindowUI

- ConfirmationWindowUI(Message: string)
- onYESClick()
- onNOClick()
- display()

(η) ListOfOrdersUI

- listOfOrdersUI()
- display()

- (θ) CancelOrderUI
 - onSearchClick()
 - display()
- (ι) ErrorUI
 - ErrorUI(Message: String)
 - onDoneClick()
 - display()
- (ια) SelectDrinkUI
 - selectDrinkUI(DrinksList: Drink[1..10])
 - onDrinkClick()
- (ιβ) ListOrdersUI
 - listOrdersUI(Orders: Order[0...40], Drinks Drink[1...10])
 - display()

2. Χρήστης Διαχειριστής

- ManagerMenuUI
 - onNewDrinkClick()
 - onRemoveDrinkClick()
 - onNewBottleClick()
 - onRemoveBottleClick()
 - display()
- NewDrinkUI
 - newDrinkUI(Bottle[0...10])
 - refreshBottles()
 - onDoneButtonClick()
 - onNewBottleClick()
 - onRemoveBottleClick()
 - display()
- NewBottleUI
 - newBottleUI(FreeBottleIDs: int[0...10])
 - onDoneClick()
 - onBackClick()
 - display()
- RemoveBottleUI
 - removeBottleUI(Bottles:Bottle[0...10])
 - onBottleClick(BottleID: int)
 - display()
- RemoveDrinkUI
 - display()

Στην παραπάνω υλοποίηση δυνατοτήτων, ο χρήστης manager κληρονομεί όλες τις λειτουργίες του barman και έχει επιπλέον κάποιες δυνατότητες προσθαφαίρεσης ποτών και συνταγών.

4.5 Γενικός έλεγχος λογισμικού

Το σύστημα RoboBar χρησιμοποιείται από δυο διαφορετικές κατηγορίες ο απλός χρήστης και ο διαχειριστής. Δικαίωμα για αλλαγή των δεδομένων του συστήματος έχει μόνο ο διαχειριστής. Όλες οι αλλαγές πραγματοποιούνται από πολλούς controllers έτσι ώστε ο έλεγχος να είναι κατανεμημένος. Επιπλέον η μορφή των ακολουθιακών διαγραμμάτων έχει τη μορφή σκάλας, έτσι η καλύτερη επιλογή για τον γενικό έλεγχο του συστήματος είναι το αποκεντρωμένο μοντέλο σχεδίασης.

Η ακολουθία εκτέλεσης της κάθε λειτουργίας στο σύστημα έχει ως εξής:

Ο χρήστης μέσω του γραφικού περιβάλλοντος ενεργοποιεί μια λειτουργία. Στη συνέχεια για την εκτέλεση της λειτουργίας καλείται ένα αντικείμενο έλεγχου. Το αντικείμενο έλεγχου αναγνωρίζει το αίτημα και εκκινεί τη διαδικασία εκτέλεσης του αιτήματος η προωθεί το αίτημα σε άλλο αντικείμενο έλεγχου. Κατά τη διάρκεια της εκτέλεσης συχνά υπάρχει επικοινωνία με τις βάσεις δεδομένων για ανάκτηση των επιθυμητών δεδομένων, οι οποίες αποτελούν οριακά αντικείμενα.

Κατά τη λειτουργία του συστήματος περισσότεροι του ενός χρηστές μπορεί να είναι ταυτόχρονα συνδεδεμένοι σε αυτό αλλά μόνο ένας από αυτούς μπορεί να είναι διαχειριστής. Επομένως αποφεύγεται η περίπτωση δύο χρήστες να μεταβάλλουν ταυτόχρονα τα δεδομένα του συστήματος. Συνεπώς η εφαρμογή χρειάζεται δημιουργία πολλών νημάτων ώστε να μπορεί να χειριστεί πολλαπλά αιτήματα από τους χρηστές.

5 Οριακές Συνθήκες

5.1 Έναρξη εφαρμογής

Για την εκκίνηση της εφαρμογής ο χρήστης πρέπει να πατήσει το εικονίδιο της εφαρμογής RoboBar στη συσκευή που χρησιμοποιεί. Για την είσοδο στο σύστημα απαιτείται ο κωδικός που έχει δοθεί στο χρηστή. Επομένως απαραίτητη προϋπόθεση για την είσοδο του χρηστή στο σύστημα είναι η επικοινωνία της συσκευής του χρηστή με τον server μέσω WiFi.

Σύντομη περιγραφή:	Σενάριο χρήσης για την εκκίνηση της εφαρμογής.
Πυροδότηση Δραστηριότητας:	Επιλογή του εικονιδίου της εφαρμογής .
Προϋποθέσεις:	Να είναι εγκατεστημένη η εφαρμογή RoboBar στη συσκευή του χρήστη.

Βασική Ροή:		
Γραμμή	Ενέργεια χρήστη συστήματος	Απάντηση συστήματος
1.	Ο χρήστης πατάει το εικονίδιο της εφαρμογής.	Το σύστημα ελέγχει αν υπάρχει σύνδεση με τον server.
2.		Το σύστημα συνδέεται με την βάση δεδομένων.
Μετέπειτα κατάσταση:	Εμφανίζεται η σελίδα σύνδεσης.	

Εναλλακτική Ροή (EP-1): Το σύστημα δεν μπορεί να συνδεθεί με τον στο τοπικό δίκτυο μέσω WiFi.		
Εφόσον το σύστημα στη γραμμή 1 στη Βασική Ροή δεν συνδεθεί, πρέπει να επιστρέφει σχετικό μήνυμα σφάλματος.		
Γραμμή	Ενέργεια χρήστη συστήματος	Απάντηση συστήματος
1.		Το σύστημα εμφανίζει μήνυμα σφάλματος.
Μετέπειτα κατάσταση:	Το σενάριο χρήσης επιστρέφει στη γραμμή 1 της βασικής ροής.	

5.2 Τερματισμός λογισμικού

Για τον τερματισμό της εφαρμογής ο χρήστης πρέπει να επιλέξει το πλήκτρο «Τερματισμός». Μετά την επιλογή για τερματισμό από το χρήστη το σύστημα ελέγχει αν υπάρχει κάποια διαδικασία σε εξέλιξη και εμφανίζει αντίστοιχο μήνυμα μαζί με κουμπί επιβεβαίωσης της επιλογής. Σε διαφορετική περίπτωση το σύστημα τερματίζει άμεσα. Κατά τον τερματισμό ο χρήστης αποσυνδέεται αυτόματα από το σύστημα και αποδεσμεύονται όλοι οι πόροι που δέσμευσε η εφαρμογή.

Σύντομη περιγραφή:	Σενάριο χρήσης για τον τερματισμό της εφαρμογής.
Πυροδότηση Δραστηριότητας:	Ο χρήστης επιλέγει το κουμπί «Τερματισμός».
Προϋποθέσεις:	Ο χρήστης θα πρέπει να έχει εκκινήσει την εφαρμογή.

Βασική Ροή:		
Γραμμή	Ενέργεια χρήστη συστήματος	Απάντηση συστήματος
1.	Ο χρήστης επιλέγει το κουμπί «Τερματισμός».	Το σύστημα ελέγχει αν υπάρχει κάποια λειτουργία σε εξέλιξη.
2.		Το σύστημα αποσυνδέει τον χρήστη.
3.		Το σύστημα αποσυνδέετε από τα εξωτερικά συστήματα.
4.		Το σύστημα αποδεσμεύει αυτούς που δέσμευσε.
Μετάπεται κατά-σταση:	Η εφαρμογή τερματίζει.	

Εναλλακτική Ροή (EP-2): Υπάρχει λειτουργία σε εξέλιξη.		
Εφόσον το σύστημα στη γραμμή 1 δει ότι υπάρχει κάποια λειτουργία σε εξέλιξη εμφανίζει μήνυμα επιβεβαίωσης του τερματισμού.		
Γραμμή	Ενέργεια χρήστη συστήματος	Απάντηση συστήματος
1.		Το σύστημα βλέπει ότι υπάρχει κάποια λειτουργία σε εξέλιξη.
2.		Το σύστημα εμφανίζει μήνυμα μαζί με δυο κουμπιά επιβεβαίωσης η ακύρωσης του τερματισμού.
3.	Ο χρήστης επιλέγει το κουμπί επιβεβαίωσης	Το σύστημα τερματίζει τις εκκρεμές λειτουργίες.
Μετάπεται κατά-σταση:	Το σενάριο χρήσης επιστρέφει στη γραμμή 2 της βασικής ροής.	

Εναλλακτική Ροή (EP-3): Ο χρήστης επιλέγει το κουμπί ακύρωση.		
Εφόσον ο χρήστης στη γραμμή 3 της EP-1 επιλέξει το κουμπί ακύρωση ο τερματισμός του συστήματος ακυρώνεται.		
Γραμμή	Ενέργεια χρήστη συστήματος	Απάντηση συστήματος
1.	Ο χρήστης επιλέγει το κουμπί ακύρωση	Το σύστημα ακυρώνει την διαδικασία τερματισμού.
Μετάπεται κατά-σταση:	.	

5.3 Σφάλματα Λογισμικού

Σφάλματα μπορούν να συμβούν κατά τη διάρκεια προσπέλασης μιας βάσης δεδομένων για την ενημέρωση των στοιχείων της, για ανάκτηση δεδομένων από αυτή ή κατά τη διάρκεια της εκτέλεσης κάποιας ενέργειας η οποία απαιτεί επικοινωνία του συστήματος με ένα εξωτερικό σύστημα. Σε περίπτωση λοιπόν που το σύστημα δεν μπορεί να επικοινωνήσει με την βάση δεδομένων η με κάποιο εξωτερικό σύστημα εμφανίζει αντίστοιχο μήνυμα σφάλματος μαζί με ένα κουμπί επανάληψης της ενεργείας και ένα κουμπί ακύρωσης.

Σύντομη περιγραφή:	Σενάριο χρήσης για διαχείριση σφαλμάτων της εφαρμογής.
Πυροδότηση Δραστηριότητας:	Το σύστημα δεν μπορεί να συνδεθεί με τα εξωτερικά συστήματα.
Προϋποθέσεις:	Ο χρήστης θα πρέπει να έχει εκκινήσει την εφαρμογή.

Βασική Ροή:		
Γραμμή	Ενέργεια χρήστη συστήματος	Απάντηση συστήματος
1.	Ο χρήστης κάνει κάποια ενέργεια που προϋποθέτει επικοινωνία με την βάση δεδομένων ή το NAO ή το ALD.	Το σύστημα δεν μπορεί να επικοινωνήσει με το εξωτερικό σύστημα.
2.		Το σύστημα εμφανίζει μήνυμα σφάλματος και ένα κουμπί για επανάληψη της διαδικασίας και ένα για ακύρωση.
3.	Ο χρήστης επιλέγει το κουμπί επανάληψη	Το σύστημα προσπαθεί ξανά να εκτελέσει την εντολή του χρήστη.
Μετέπειτα κατά-σταση:	Το σύστημα εκτελεί την εντολή του χρήστη αν κατάφερε να συνδεθεί με το εξωτερικό σύστημα που αδυνατούσε αλλιώς επιστρέφει στη γραμμή 2 της Βασικής ροής του σεναρίου.	

Εναλλακτική Ροή (EP-4): Ο χρήστης επιλέγει το κουμπί ακύρωση.		
Εφόσον ο χρήστης στη γραμμή 3 της βασικής ροής του σεναρίου επιλέξει το κουμπί ακύρωση, το σύστημα επιστρέφει στην αρχική οθόνη.		
Γραμμή	Ενέργεια χρήστη συστήματος	Απάντηση συστήματος
1.	Ο χρήστης επιλέγει το κουμπί ακύρωση	Το σύστημα επιστρέφει στην αρχική οθόνη.
Μετέπειτα κατά-σταση:	.	

5.4 Διακοπή τροφοδοσίας

Σε περίπτωση διακοπής τροφοδοσίας ο server του συστήματος έχει την δυνατότητα να επαναλειτουργεί αυτόματα και να συνεχίζει κανονικά τις λειτουργίες του.

A Λίστα Ικνηλασιμότητας

Στο έγγραφο αυτό έχουν γίνει κάποιες αλλαγές σε σχέση με το έγγραφο απαιτήσεων χρηστών και το έγγραφο απαιτήσεων λογισμικού όσο αφορά της λειτουργίες του συστήματος που περιγράφηκαν. Οι αλλαγές που πραγματοποιηθήκαν είναι οι παρακάτω:

- Στην παράγραφο με τις οριακές συνθήκες περιγράφεται ο τρόπος τερματισμού του λογισμικού μέσω του κουμπιού «Τερματισμός». Η λειτουργεί αυτή και το κουμπί δεν είχαν περιγράψει στα προηγούμενα έγγραφα.