



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

Απαιτήσεις Λογισμικού

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

2 Μαΐου 2018

Ιστορικό Αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Φρανκ	27 Απρ 2018	Αρχικό έγγραφο	0.1
Χριστίνα	29 Απρ 2018	Προσθήκες Περιγραφών	0.2
Κώστας	29 Απρ 2018	Προσθήκες Περιγραφών	0.3
Χριστίνα	30 Απρ 2018	Προσθήκες Περιγραφών	0.4
Φρανκ	1 Μαί 2018	Διορθώσεις	0.5
Κώστας	1 Μαί 2018	Λοιπά κεφάλαια	0.6
Χριστίνα	1 Μαί 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	Τυπογραφικές παραδοχές του εγγράφου	4
1.3	Αναγνωστικό κοινό και τρόπος ανάγνωσης	4
1.4	Dependencies	5
1.5	Σκοπός του έργου	6
2	Στατική Μοντελοποίηση	7
2.1	MakeOrder Package	7
2.2	Orders Package	11
2.3	BarmanMenus Package	15
2.4	EmptyBottle Package	18
2.5	EditRecipe Package	21
2.6	Login Package	25
3	Μη λειτουργικές απαιτήσεις	28
3.1	Απαιτήσεις επίδοσης	28
3.2	Απαιτήσεις Αξιοπιστίας	28
3.3	Απαιτήσεις Χρηστικότητας	28
4	Πρότυπα Σχεδιασμού που υιοθετήθηκαν	30
4.1	Πρότυπα συμπεριφοράς	30
A	Λίστα Ιχνηλασιμότητας	31
B	Ανοιχτά Θέματα	31

1 Εισαγωγή

1.1 Στόχος του εγγράφου

Στόχος του παρόντος εγγράφου είναι η επεξήγηση και ανάλυση των απαιτήσεων λογισμικού του συστήματος. Περιγράφονται τα χαρακτηριστικά και οι μέθοδοι που είναι απαραίτητες για την σωστή λειτουργία του συστήματος λαμβάνοντας υπ'όψιν της μη λειτουργικές και λειτουργικές απαιτήσεις που αναπτύχθηκαν στο προηγούμενο έγγραφο. Ορίζεται ο ακριβής τρόπος λειτουργίας του συστήματος σε σχέση με τους χρήστες, τη δυναμική των περιβαλλόντων διεπαφής καθώς και την συνεργασία με εξωτερικά συστήματα.

Αρχικά αναφέρεται η στατική μοντελοποίηση του συστήματος. Αυτή η δόμη καθορίζει τα είδη των αντικειμένων που είναι σημαντικά για το σύστημα καθώς και τις συσχετίσεις μεταξύ τους. Η ανάλυση των σεναρίων χρήσης σε κλάσεις υλοποιήθηκε με την χρήση της uml και των διαγραμμάτων κλάσεων. Τα διαγράμματα αυτά παρέχουν πολύ χρήσιμες πληροφορίες σχετικά με τον κώδικα όπως ο τύπος των δεδομένων που θα χρησιμοποιηθούν, οι σχέσεις μεταξύ των κλάσεων και οι ιδιότητές τους.

Οι κλάσεις ταξινομούνται σε πακέτα ανάλογα με την λειτουργία τους. Τα πακέτα ταξινομούνται και αυτά με την σειρά τους σε υψηλής, μέτριας και χαμηλής προτεραιότητας ανάλογα με την σπουδαιότητα της λειτουργίας που περιγράφουν. Κάθε πακέτο περιέχει αντικείμενα διαφορετικών τύπων. Αρχικά υπάρχουν τα αντικείμενα οντοτήτων (Entity Objects) τα οποία αναπαριστούν τη μόνιμη πληροφορία η οποία ανιχνεύεται από το σύστημα. Έπειτα υπάρχουν τα οριακά αντικείμενα (Boundary Objects) που αναπαριστούν την αλληλεπίδραση ανάμεσα στον χρήστη και το σύστημα και τα αντικείμενα ελέγχου (Control Objects) που αναπαριστούν τις εργασίες ελέγχου που εκτελούνται από το σύστημα και συντονίζουν την συμπεριφορά του.

Ο καθορισμός των μεθόδων των κλάσεων αυτών συνιστά τη μοντελοποίηση της δυναμικής συμπεριφοράς του συστήματος. Η δυναμική μοντελοποίηση χρησιμοποιείται στην αναπαράσταση της συμπεριφοράς και των αλληλεπιδράσεων των στατικών συστατικών του συστήματος όπως οι κλάσεις, τα αντικείμενα, οι συμπεριφορές τους και οι διεπαφές.

1.2 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.

1.3 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:

- Οι ειδικοί του πεδίου εφαρμογής προκειμένου να μελετήσουν τα διαγράμματα κλάσεων και να μοντελοποιήσουν το πεδίο εφαρμογής και
- Η ομάδα ανάπτυξης η οποία χρησιμοποιεί τα διαγράμματα κλάσεων κατά τις φάσεις ανάλυσης, σχεδίασης του συστήματος, σχεδίασης των αντικειμένων και προγραμματισμού. Τα μέλη της ομάδας ανάπτυξης ανάλογα με την δραστηριότητα τους μπορούν να έχουν διαφορετικούς ρόλους όπως:
 - Αναλυτής: Ο αναλυτής ενδιαφέρεται κυρίως για τις κλάσεις εφαρμογής, που σημαίνει ότι οι συσχετίσεις ανάμεσα στις κλάσεις είναι σχέσεις ανάμεσα στις αφαιρέσεις στο πεδίο εφαρμογής, τις λειτουργίες των κλάσεων του πεδίου εφαρμογής και στην εκμετάλλευση της κληρονομικότητας για την αναπαράσταση των ταξονομιών στο πεδίο εφαρμογής. Κρίνεται λοιπόν απαραίτητη η μελέτη των διαγραμμάτων κλάσεων από τον αναλυτή.
 - Σχεδιαστής Συστήματος: Ο σχεδιαστής συστήματος ενδιαφέρεται για την αρχιτεκτονική, τα δομικά στοιχεία, τις διεπαφές και τον τύπο δεδομένων του συστήματος έτσι ώστε να ικανοποιούνται οι λειτουργικές και οι μη λειτουργικές απαιτήσεις του. Κρίνεται λοιπόν απαραίτητο για αυτόν να

μελετήσει τον σχεδιασμό των πακέτων, την ταξινόμησή τους κατά προτεραιότητα και τα διαγράμματα κλάσεων, δίνοντας έμφαση στις οριακές κλάσεις (Boundary).

- Σχεδιαστής Αντικειμένων: Ο σχεδιαστής αντικειμένων ενδιαφέρεται για τον σχεδιασμό των αντικειμένων που αλληλεπιδρούν μεταξύ τους με σκοπό την επίλυση του προβλήματος. Πρέπει λοιπόν να δώσει μεγάλη έμφαση στα διαγράμματα κλάσεων και κυρίως στις κλάσεις οντοτήτων (Entities).
- Προγραμματιστής: Ο προγραμματιστής ενδιαφέρεται μόνο για την ανάπτυξη του προγράμματος συμπεριλαμβανομένου του σχεδιασμού, του προγραμματισμού και της δοκιμής του λογισμικού. Και για αυτόν είναι άκρως απαραίτητη η μελέτη των διαγραμμάτων κλάσεων.

Για την ευκολότερη ανάγνωση του εγγράφου είναι χωρισμένο σε 4 κεφάλαια και είναι δομημένο με τον παρακάτω τρόπο:

Κεφάλαιο 1ο: Αποτελεί την εισαγωγή του εγγράφου αναφέροντας τον γενικότερο σκοπό του, κάποιες παραδοχές, το αναγνωστικό κοινό στο οποίο απευθύνεται και τον στόχο της εφαρμογής.

Κεφάλαιο 2ο: Υλοποιείται η μοντελοποίηση των σεναρίων χρήσης του συστήματος σε κλάσεις. Αρχικά ταξινομούνται σε πακέτα ανάλογα με τις λειτουργίες που επιτελούν και ταξινομούνται ανάλογα με την προτεραιότητα που έχουν. Έπειτα περιγράφονται αναλυτικά οι μέθοδοι και τα χαρακτηριστικά τους και παρουσιάζονται τα διαγράμματα Uml. Εδώ πρέπει να αναφερθεί ότι για την μείωση της πολυπλοκότητας των διαγραμμάτων δεν έχουν προστεθεί οι συσχετίσεις μεταξύ των πακέτων αλλά αναφέρονται πιο κάτω σε πίνακα.

Κεφάλαιο 3ο: Στο κεφάλαιο αυτό αναφέρονται και ταξινομούνται οι μη λειτουργικές απαιτήσεις του συστήματος.

Κεφάλαιο 4ο: Παρουσιάζονται τα πρότυπα σχεδίασης που χρησιμοποιήθηκαν

Παράρτημα Α: Περιέχει τον πίνακα ιχνηλασιμότητας ο οποίος αποτελεί έναν μηχανισμό εξάρτησης μεταξύ των απαιτήσεων και των υπόλοιπων στοιχείων του λογισμικού. Μας καθοδηγεί, δηλαδή, από κάθε απαίτηση σε εκείνο το στοιχείο του σχεδίου ή της μονάδας λογισμικού που την καλύπτει.

Παράρτημα Β: Περιέχει μια λίστα με τα ανοιχτά θέματα που αφορούν το σύστημα που αναπτύσσεται.

1.4 Dependencies

Στα σχήματα που παρουσιάζονται στη συνέχεια του εγγράφου στη στατική μοντελοποίηση αφαιρέθηκαν ορισμένες συνδέσεις μεταξύ κλάσεων. Η αφαίρεση αυτή έγινε σε ορισμένες κλάσεις που έχουν σαν μεταβλητές αντικείμενα άλλων κλάσεων ή και συναρτήσεις οι οποίες δέχονται σαν όρισμα η επιστρέφουν αντικείμενα άλλων κλάσεων. Ο σκοπός της αφαίρεσης είναι η αποφυγή πολύπλοκων σχημάτων ώστε να διευκολυνθεί η ανάγνωση τους. Οι συσχετίσεις που παραλήφθηκαν αναφέρονται στη συνέχεια:

Κλάση Material του πακέτου MakeOrderPackage :

- Με την κλάση EditRecipeDBProxy στη συνάρτηση addDrink(in Recipe:Material[1..10], in Name:String) του πακέτου EditRecipePackage.
- Με την κλάση EditDrinkController στη συνάρτηση addRecipe(in Recipe:Material[1..10], in Name:String) του πακέτου EditRecipePackage.

Κλάση Order του πακέτου MakeOrderPackage :

- Με την κλάση ListOrdersUI στη συνάρτηση listOrdersUI (in Orders:Order[0..40], in Drinks Drink[1..10]) του πακέτου OrdersPackage.

Κλάση Drink του πακέτου OrdersPackage :

- Με την κλάση DrinksDBProxy στη συνάρτηση getDrinks(): Drink[0..10] του πακέτου BarmanMenusPackage.
- Με την κλάση EditRecipeDBProxy στη συνάρτηση getDrinks(): Drink[0..10]] του πακέτου EditRecipePackage.

1.5 Σκοπός του έργου

Η εφαρμογή έχει ως στόχο την δημιουργία μιας εφαρμογής που δίνει κατάλληλες εντολές σε ένα ρομπότ ΝΑΟ έτσι ώστε αυτό να προσομοιώνει τις κινήσεις ενός μπάρμαν σε ένα μπαρ. Σκοπός είναι να γίνει δυνατό μετά την λήψη μιας παραγγελίας να παραδίνεται στον πελάτη το επιθυμητό ποτό από ένα ρομπότ μέσω μιας προστατευμένης και ασφαλούς διαδικασίας τόσο για τον ιδιοκτήτη όσο και για τον πελάτη.

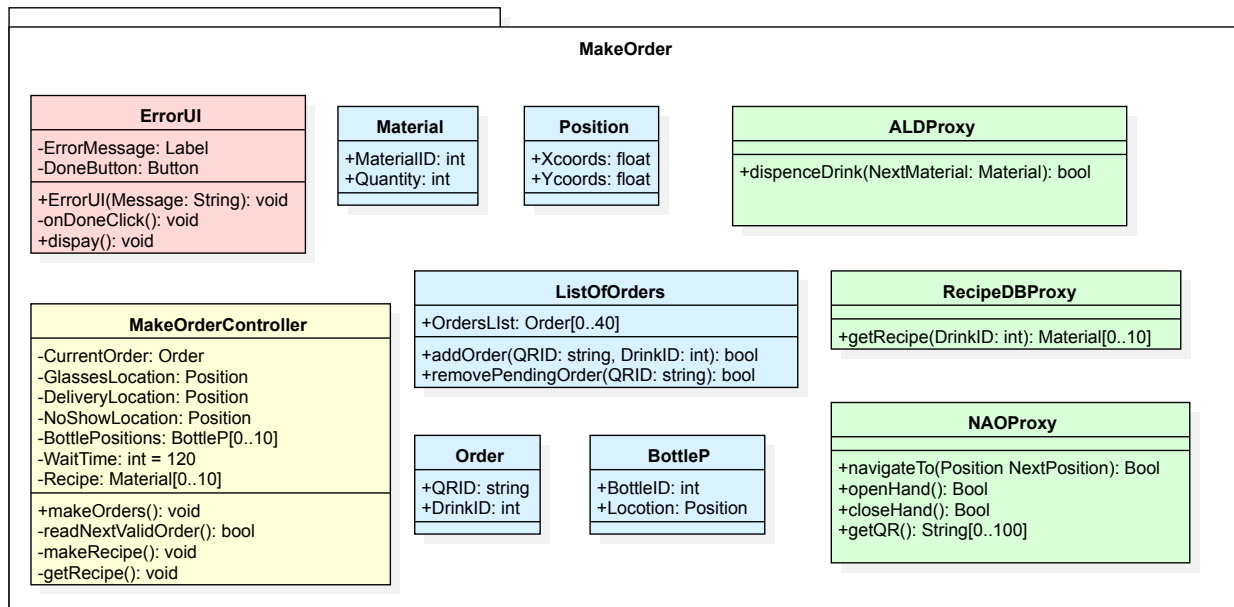
Θεωρούμε ότι η συγκριμένη εφαρμογή θα μπορούσε να φανεί πολύ χρήσιμη σε μαγαζιά τα οποία δεν έχουν την δυνατότητα να απασχολήσουν πολύ προσωπικό και θα χρειάζονταν βοήθεια. Επίσης θεωρούμε ότι θα ήταν για το μαγαζί μια ενδιαφέρουσα αναβάθμιση αρεστή στους πελάτες, ικανή να τους προσελκύσει καθώς και μια μεγάλη βοήθεια για τους εργαζομένους.

Σκοπός μας είναι να δημιουργήσουμε μια εφαρμογή εύχρηστη και συνεργάσιμη κυρίως για τους εργαζόμενους και τον ιδιοκτήτη του μπαρ , να διευκολύνει στο έργο τους και να μιμείται όσο το δυνατόν πιο πιστά τις ανθρώπινες κινήσεις καθώς και να ελαχιστοποιήσει τα ανθρώπινα λάθη της παραγωγής.

Γνωρίζουμε βεβαίως ότι αυτή θα μπορεί να λειτουργεί μόνο ως εξτρά βοήθεια στο ήδη υπάρχων προσωπικό και σε καμία περίπτωση δεν θα μπορούσε να αντικαταστήσει εξ' ολοκλήρου κάποιον υπάλληλο.

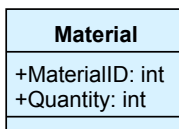
2 Στατική Μοντελοποίηση

2.1 MakeOrder Package



Το πακέτο MakeOrder περιέχει τις απαραίτητες κλάσεις για την εκτέλεση της παρασκευής ενός ποτού και την παράδοση του στον πελάτη.

Material

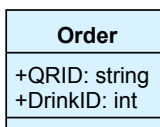


Ένα βήμα μιας συνταγής παρασκευής ενός ποτού.

Χαρακτηριστικά της κλάσης:

- **MaterialID: int:** Αναγνωριστικό ενός υλικού για την παρασκευή ενός ποτού
- **Quantity: int:** Η ποσότητα του υλικού όπου απαιτείται.

Order



Μία καταχώρηση στην λίστα παραγγελιών.

Χαρακτηριστικά της κλάσης:

- **QRID: string:** Το αναγνωριστικό της παραγγελίας.
- **DrinkID: int:** Το αναγνωριστικό του ποτού της παραγγελίας.

ListofOrders

ListOfOrders
+OrdersList: Order[0..40]
+addOrder(QRID: string, DrinkID: int): bool +removePendingOrder(QRID: string): bool

Η λίστα προτεραιότητας των παραγγελιών.

Χαρακτηριστικά της κλάσης:

- OrdersList: Order[0..40]: Με σειρά προτεραιότητας, οι εκκρεμές παραγγελίες.

Μέθοδοι της κλάσης:

- addOrder(QRID: string, DrinkID: int): bool: Προσθήκη νέας παραγγελίας στην λίστα με τα τα στοιχεία του ορίσματος. Επιστρέφει false όταν είναι γεμάτη η λίστα.
- removePendingOrder(QRID: string): bool: Αναζήτηση και αφαίρεση της παραγγελίας με το αναγνωριστικό παραγγελίας του ορίσματος. Επιστέφει false σε περίπτωση μη εύρεσης της παραγγελίας.

Position

Position
+Xcoords: float +Ycoords: float

Ζεύγος συντεταγμένων, συμβατό με τα API μετακίνησης του NAO.

Χαρακτηριστικά της κλάσης:

- Xcoords: float: Οι x συντεταγμένες.
- Ycoords: float: Οι y συντεταγμένες.

BottleP

BottleP
+BottleID: int +Locotion: Position

Ζεύγος συντεταγμένων, για κάθε θέση των μπουκαλιών.

Χαρακτηριστικά της κλάσης:

- BottleID: int: Αναγνωριστικό ενός υλικού - μπουκαλιού
- Location: Position: Η θέση του μπουκαλιού.

RecipeDBProxy

RecipeDBProxy
+getRecipe(DrinkID: int): Material[0..10]

Η RecipeDBProxy περιέχει μεθόδους για την ανάκληση των απαραίτητων δεδομένων από την βάση δεδομένων για την παρασκευή ενός ποτού.

Μέθοδοι της κλάσης:

- getRecipe(DrinkID: int): Material[0..10]: Δέχεται ως όρισμα ένα DrinkID και επιστρέφει την συνταγή του ως ένα μία λίστα Material[].

ALDProxy

ALDProxy
+dispenceDrink(NextMaterial: Material): bool

Η ALDProxy περιέχει μεθόδους για την διασύνδεση του συστήματος με το εξωτερικό σύστημα ALD για την εξαγωγή ορισμένης ποσότητας ποτού.

Μέθοδοι της κλάσης:

- dispenceDrink(NextMaterial: Material): bool: Δέχεται ως όρισμα ένα Material και επικοινωνεί με το ALD για την εξαγωγή ορισμένης ποσότητας ορισμένου ποτού. Επιστρέφει false σε περίπτωση αποτυχίας.

NAOProxy

NAOProxy
+navigateTo(Position NextPosition): Bool +openHand(): Bool +closeHand(): Bool +getQR(): String[0..100]

Η NAOProxy περιέχει μεθόδους για την διασύνδεση του συστήματος με το εξωτερικό σύστημα NAO.

Μέθοδοι της κλάσης:

- navigateTo(Position NextPosition): Bool: Μετακίνηση του NAO στις συντεταγμένες του ορίσματος. Επιστρέφει false σε περίπτωση αποτυχίας.
- openHand(): Bool: Άνοιγμα του βραχίονα του NAO. Επιστρέφει false σε περίπτωση αποτυχίας.
- closeHand(): Bool: Κλείσιμο του βραχίονα του NAO. Επιστρέφει false σε περίπτωση αποτυχίας.
- getQR(): String[0..100]: Ενεργοποίηση της λειτουργίας ανάγνωσης barcodes του NAO και επιστροφή των αποτελεσμάτων.

ErrorUI

ErrorUI
-ErrorMessage: Label -DoneButton: Button
+ErrorUI(Message: String): void -onDoneClick(): void +dispay(): void

Το `ErrorUI` είναι ένα popup window όπου εμφανίζεται σε περίπτωση σφάλματος και τερματίζει όταν πατηθεί το κουμπί "Done"

Χαρακτηριστικά της κλάσης:

- `ErrorMessage: Label`: Το μήνυμα λάθους
- `DoneButton: Button`: Κουμπί ενημέρωσης πως το πρόβλημα επιλύθηκε.

Μέθοδοι της κλάσης:

- `ErrorIUO(Message: string)`: void: Δέχεται ως όρισμα το μήνυμα όπου θα εμφανιστεί στο παράθυρο.
- `onDoneClick()`: void: Τερματίζει άμεσα.
- `display()`: void: Εμφανίζει το παράθυρο και τερματίζει μετά την εκτέλεση της `onDoneClick()`.

MakeOrderController

MakeOrderController
-CurrentOrder: Order -GlassesLocation: Position -DeliveryLocation: Position -NoShowLocation: Position -BottlePositions: BottleP[0..10] -WaitTime: int = 120 -Recipe: Material[0..10]
+makeOrders(): void -readNextValidOrder(): bool -makeRecipe(): void -getRecipe(): void

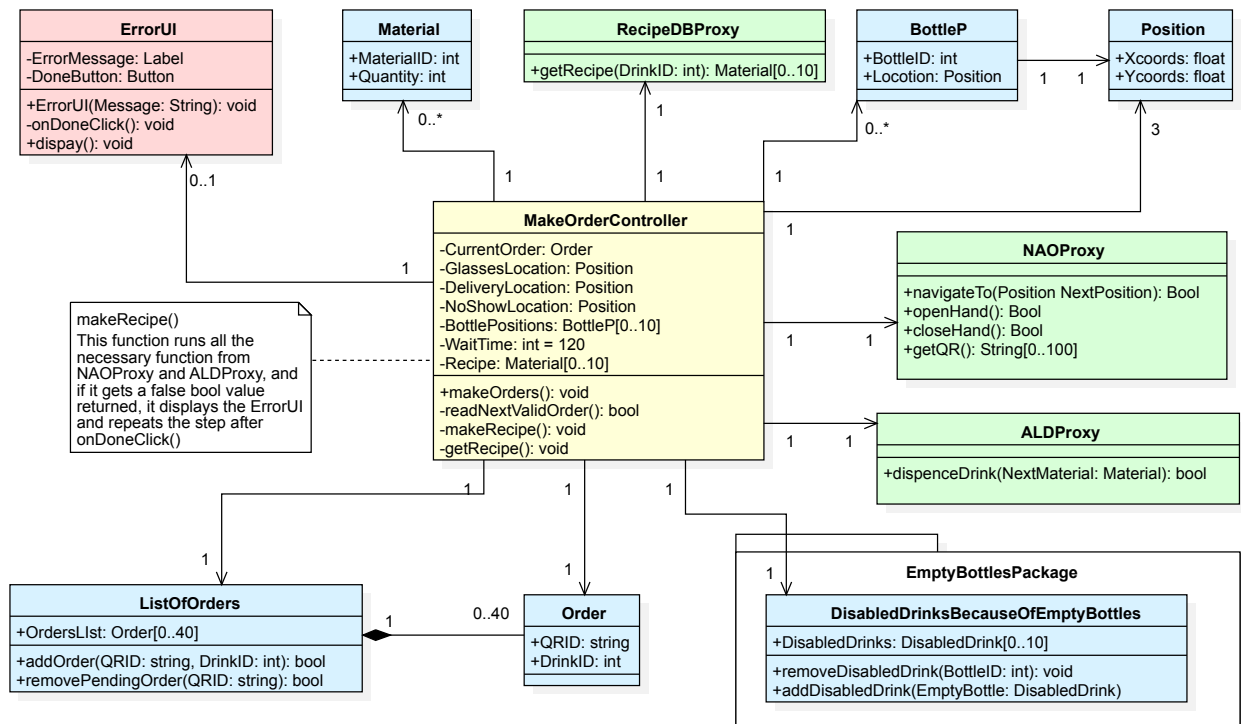
Ο `MakeOrderController`, διαβάζει την επόμενη παραγγελία από την `ListOfOrders` η οποία δεν έχει απεργασποιηθεί λόγω άδειου μπουκαλιού και δίνει τις απαραίτητες εντολές στο `NAO` και το `ALD` ώστε να παραχθεί το ποτό και να παραλειφθεί από τον πελάτη.

Χαρακτηριστικά της κλάσης:

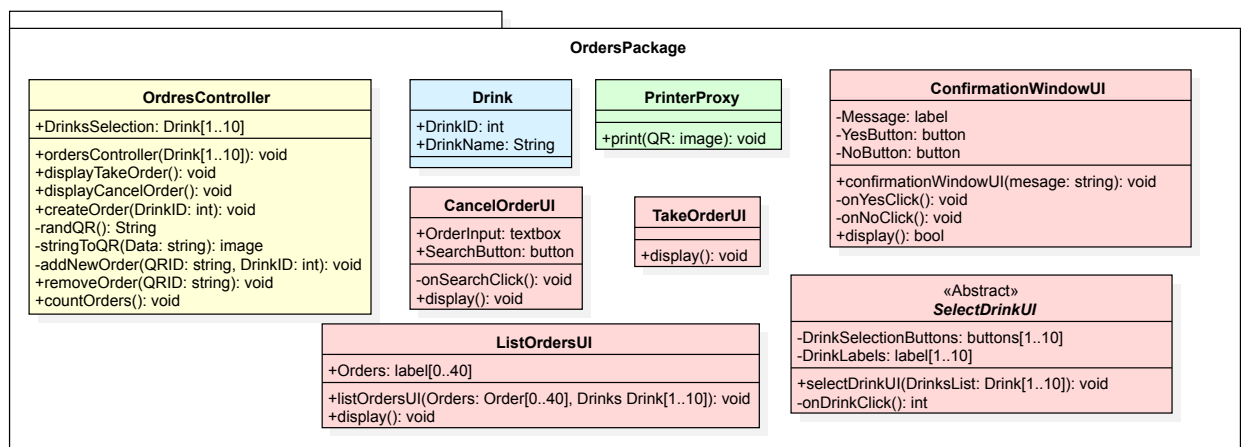
- `CurrentOrder: Order`: Αντίγραφο της καταχώρησης της παραγγελίας η οποία εκτελείται.
- `GlassesLocation: Position`: Η θέση των ποτηριών.
- `DeliveryLocation: Position`: Η θέση του σημείου παραλαβής ποτού από τον πελάτη.
- `NoShowLocation: Position`: Η θέση όπου αποθηκεύονται τα ποτά που δεν παραλήφθηκαν μέσα στο χρονικό περιθώριο `WaitTime` (s).
- `BottlePositions: BottleP[0..10]`: Οι θέσεις των στομιών όλων των μπουκαλιών.
- `WaitTime: int = 120`: Ο χρόνος αναμονής προσέλευσης του πελάτη για τη παραλαβή του ποτού του.
- `Recipe: Material[0..10]`: Τα βήματα της συνταγής της παραγγελίας όπου εκτελείται.

Μέθοδοι της κλάσης:

- `makeOrders()`: void: Ανά τακτά χρονικά διαστήματα εκτελεί την `readNextValidOrder()`. Αν επιστρέψει true, εκτελεί με την σειρά την `getRecipe()` και `makeRecipe()`.
- `readNextValidOrder()`: bool: Διαβάζει το πρώτο στοιχείο της λίστας `ListOfOrders` και αν το `DrinkID` της δεν περιλαμβάνεται στην λίστα `DisabledDrinksBecauseOfEmptyBottles`, την αποθηκεύει στην μεταβλητή `CurrentOrder`.
- `getRecipe()`: void: Εκτελεί την `getRecipe()` του `RecipeDBProxy` με όρισμα το `DrinkID` του `CurrentOrder` και αποθηκεύει τα επιστρεφόμενα δεδομένα στην μεταβλητή `Recipe`.
- `makeRecipe()`: void: Εκτελεί με την σωστή σειρά τις απαραίτητες εντολές του `NAOProxy` και `ALDProxy` χρησιμοποιώντας τα αποθηκευμένα δεδομένα της κλάσης ώστε να εκτελέσει την παρασκευή του ποτού όπως περιγράφεται στο ΣΧ6 του εγγράφου απαιτήσεων χρηστών. Κάθε φορά όπου μια συνάρτηση των `NAOProxy` και `ALDProxy` επιστρέφει false, εμφανίζει το `ErrorUI` και επαναλαμβάνει το βήμα.

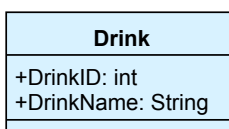


2.2 Orders Package



Το πακέτο Orders περιέχει τις κατάλληλες κλάσεις ώστε μέσω γραφικών διεπαφών να προσθέτονται και αφαιρούνται παραγγελίες, όπως και να γίνεται ενημέρωση των εκκρεμών παραγγελιών.

Drink



Συσχέτιση των DrinkIDs με το όνομα του μπουκαλιού όπως τα έχει εισάγει ο διαχειριστής.

Χαρακτηριστικά της κλάσης:

- DrinkID: int: Ο χαρακτηριστικός αριθμός κάθε ποτού
- DrinkName: String: Το όνομα κάθε ποτού

ConfirmationWindowUI

ConfirmationWindowUI
-Message: label -YesButton: button -NoButton: button
+confirmationWindowUI(message: string): void -onYesClick(): void -onNoClick(): void +display(): bool

Ανδύόμενο παράθυρο επιβεβαίωσης

Χαρακτηριστικά της κλάσης:

- Message: label: Η επιγραφή του παραθύρου
- YesButton: button: Το κουμπί "Ναι"
- NoButton: button: Το κουμπί "Όχι"

Μέθοδοι της κλάσης:

- confirmationWindowUI(message: string): void: Δέχεται ως όρισμα το μήνυμα όπου θα εμφανιστεί στο παράθυρο.
- onYesClick(): void: Προκαλεί τερματισμό της display() με επιστροφή true.
- onNoClick(): void: Προκαλεί τερματισμό της display() με επιστροφή false.
- display(): bool: Εμφανίζεται το παράθυρο επιβεβαίωσης και επιστρέφει bool ανάλογα με το ποιο κουμπί πατήθηκε.

SelectDrinkUI

«Abstract» <i>SelectDrinkUI</i>
-DrinkSelectionButtons: buttons[1..10] -DrinkLabels: label[1..10]
+selectDrinkUI(DrinksList: Drink[1..10]): void -onDrinkClick(): int

Μια abstract διεπαφή όπου εμφανίζει μια λίστα με τα ποτά και κουμπιά για την επιλογή τους.

Χαρακτηριστικά της κλάσης:

- DrinkSelectionButtons: buttons[1..10]: Κουμπιά επιλογής σε στήλη.
- DrinkLabels: label[1..10]: Ονομασίες επιλογών σε στήλη.

Μέθοδοι της κλάσης:

- selectDrinkUI(DrinksList: Drink[1..10]): void: Δέχεται ως όρισμα τις ονομασίες και τα αντίστοιχα IDs ποτών.
- onDrinkClick(): int: Επιστρέφει το DrinkID όπου αντιστοιχεί στο ποτό όπου επιλέχθηκε.

TakeOrderUI

TakeOrderUI
+display(): void

Υλοποίηση της SelectDrinkUI όπου επικοινωνεί με το OrdersController.

Μέθοδοι της κλάσης:

- display(): void: Εκτελεί την createOrder() του OrdersController με όρισμα την επιστροφή του onDrinkClick().

CancelOrderUI

CancelOrderUI
+OrderInput: textbox
+SearchButton: button
-onSearchClick(): void
+display(): void

Γραφική διεπαφή για την αφαίρεση μια παραγγελίας από την λίστα παραγγελιών.

Χαρακτηριστικά της κλάσης:

- OrderInput: textbox: Εισαγωγή του QRID της παραγγελίας όπου επιθυμείται η αφαίρεση.
- SearchButton: button: Κουμπί συνέχειας.

Μέθοδοι της κλάσης:

- onSearchClick(): void: Εκτελεί την removeOrder() του OrdersController με όρισμα το QRID όπου δόθηκε στο textbox.
- display(): void: Εμφανίζει την γραφική διεπαφή.

ListOrdersUI

ListOrdersUI
+Orders: label[0..40]
+listOrdersUI(Orders: Order[0..40], Drinks Drink[1..10]): void
+display(): void

Γραφική επαφή εμφάνισης μιας λίστας όλων των εκκρεμών παραγγελιών.

Χαρακτηριστικά της κλάσης:

- Orders: label[0..40]: Λίστα με όλες τις εκκρεμείς παραγγελίες QRIDs και DrinkNames.

Μέθοδοι της κλάσης:

- listOrdersUI(in Orders:Order[0..40], in Drinks Drink[1..10]): void: Δημιουργεί τα Orders labels με τα στοιχεία του ορίσματος.
- display(): void: Εμφανίζει την γραφική διεπαφή

PrinterProxy

PrinterProxy
+print(QR: image): void

Επικοινωνεί με τον εκτυπωτή για την εκτύπωση του QR image σε εισιτήριο.

Μέθοδοι της κλάσης:

- print(QR: image): void: Στέλνει την εικόνα του ορίσματος προς εκτύπωση.

OrdersController

OrdresController
+DrinksSelection: Drink[1..10]
+ordersController(Drink[1..10]): void +displayTakeOrder(): void +displayCancelOrder(): void +createOrder(DrinkID: int): void -randQR(): String -stringToQR(Data: string): image -addNewOrder(QRID: string, DrinkID: int): void +removeOrder(QRID: string): void +countOrders(): void

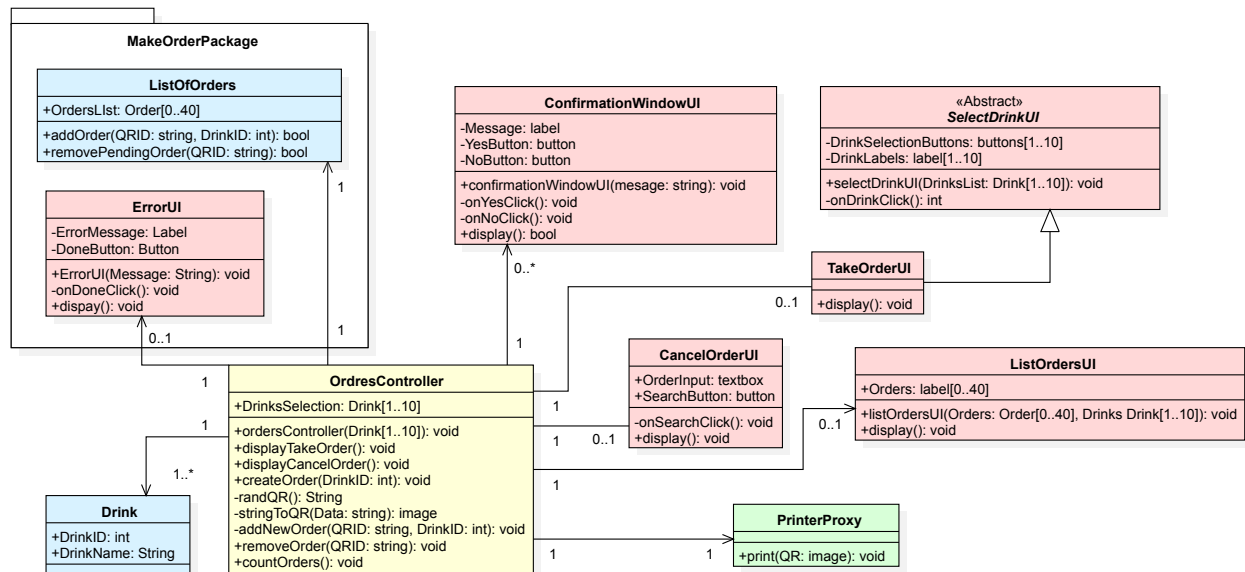
Η κλάση αυτή ελέγχει τη διαχείριση των παραγγελιών

Χαρακτηριστικά της κλάσης:

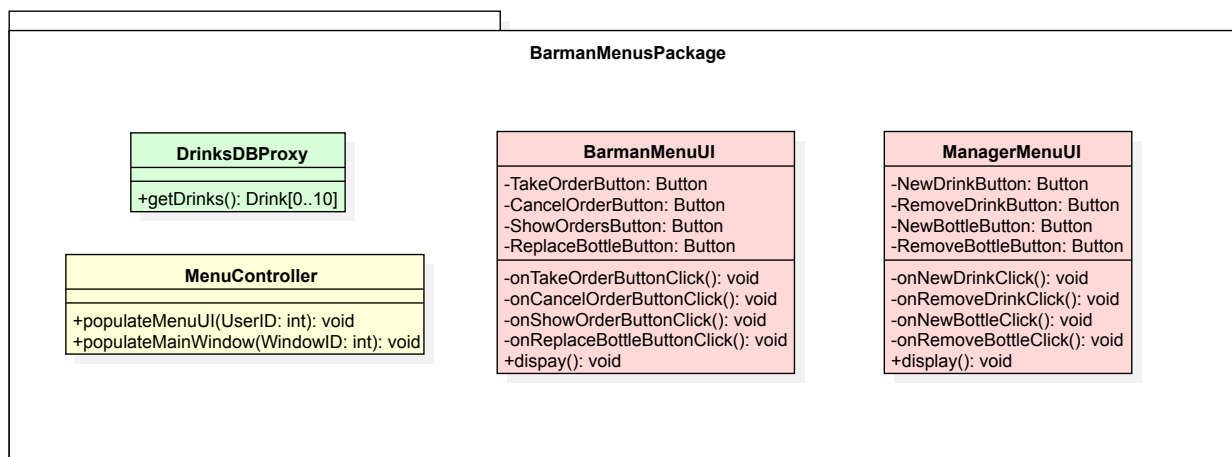
- DrinksSelection: Drink[1..10]: Λίστα με τα ποτά διαθέσιμα προς παρασκευή.

Μέθοδοι της κλάσης:

- ordersController(Drink[1..10]): void: Αρχικοποίηση της λίστας των διαθέσιμων ποτών.
- displayTakeOrder(): void: Εμφάνιση του TakeOrderUI.
- displayCancelOrder(): void: Εμφάνιση του CandelOrderUI.
- createOrder(DrinkID: int): void: Η συνάρτηση αυτή παίρνει ως όρισμα ένα DrinkID και εμφανίζει το ConfirmationWindowUI. Σε περίπτωση true επιστροφής, εκτελεί τις συναρτήσεις randQR(), stringToQR(), την print() του PrinterProxy και την addNewOrder().
- randQR(): bool: Δημιουργεί ένα τυχαίο string με αλφαριθμητικούς χαρακτήρες σταθερού μήκους.
- stringToQR(data: string): Παίρνει σαν όρισμα ένα string και το μετατρέπει σε QRcode σε μορφή εικόνας
- addNewOrder(QRID: string, DrinkID: int): void: Εκτελεί την addOrder() του ListOfOrders με τα ίδια ορίσματα. Εμφανίζει μήνυμα λάθους στην περίπτωση false επιστροφής.
- removeOrder(QRID: string): void: Η συνάρτηση αυτή παίρνει ως όρισμα το QRID και εμφανίζει το ConfirmationWindowUI. Σε περίπτωση true επιστροφής, εκτελεί την removePendingOrder() του ListOfOrders. Εμφανίζει μήνυμα λάθους στην περίπτωση false επιστροφής του removePendingOrder().
- countOrders(): void: Η συνάρτηση αυτή επιστρέφει το πλήθος των Orders στο ListOfOrders



2.3 BarmanMenus Package



Το πακέτο BarmanMenus περιέχει τις γραφικές διεπαφές του μενού και συνδέει τις επιλογές του μενού με τις γραφικές διεπαφές του κεντρικού παραθύρου.

DrinksDBProxy



Κλάση για την ανάκληση της λίστας των ποτών από την βάση δεδομένων.

Μέθοδοι της κλάσης:

- `getDrinks(): Drink[0..10]`: Επιστρέφει έναν πίνακα με όλα τα Drinks από την βάση δεδομένων.

BarmanMenuUI

BarmanMenuUI
-TakeOrderButton: Button -CancelOrderButton: Button -ShowOrdersButton: Button -ReplaceBottleButton: Button
-onTakeOrderButtonClick(): void -onCancelOrderButtonClick(): void -onShowOrderButtonClick(): void -onReplaceBottleButtonClick(): void +dispay(): void

Η κλάση αυτή εμφανίζει το μενού με τις επιλογές όπου είναι διαθέσιμες στον χρήστη Barman.

Χαρακτηριστικά της κλάσης:

- TakeOrderButton: Button: Το button για νέα παραγγελία.
- CancelOrderButton: Button: Το button ακύρωσης της παραγγελίας.
- ShowOrderButton: Button: Το button για την εμφάνιση της παραγγελίας.
- ReplaceBottleButton: Button: Το button για την αντικατάσταση μπουκαλιού.

Μέθοδοι της κλάσης:

- onTakeOrderButtonClick(): void: Εκτελεί την populateMainWindow(1) του MenuController.
- onCancelOrderButtonClick(): void: Εκτελεί την populateMainWindow(2) του MenuController.
- onShowOrderButtonClick(): void: Εκτελεί την populateMainWindow(3) του MenuController.
- onReplaceBottleButtonClick(): void: Εκτελεί την populateMainWindow(4) του MenuController.
- dispay(): void: Εμφανίζει το μενού.

ManagerMenuUI

ManagerMenuUI
-NewDrinkButton: Button -RemoveDrinkButton: Button -NewBottleButton: Button -RemoveBottleButton: Button
-onNewDrinkClick(): void -onRemoveDrinkClick(): void -onNewBottleClick(): void -onRemoveBottleClick(): void +display(): void

Η κλάση αυτή κληρονομεί όλα τα στοιχεία της κλάσης BarmanMenuUI και προσθέτει τις επιλογές όπου είναι διαθέσιμες στον Διαχειριστή.

Χαρακτηριστικά της κλάσης:

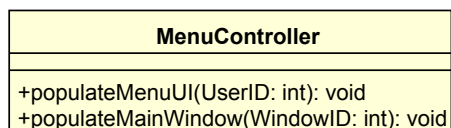
- NewDrinkButton: Το button για καταχώρηση νέας συνταγής ποτού.
- RemoveDrinkButton: Το button για διαγραφή συνταγής.
- NewBottleButton: Το button για καταχώρηση νέου .
- RemoveBottleButton: Το button για διαγραφή υλικού.

Μέθοδοι της κλάσης:

- onNewDrinkClick(): Η συνάρτηση αυτή εκτελείται όταν επιλεγεί το NewDrinkButton χρησιμοποιώντας την συνάρτηση displayNewDrink() του EditDrinkController.

- `onRemoveDrinkClick()`: Η συνάρτηση αυτή εκτελείται όταν επιλεγεί το `RemoveDrinkButton` χρησιμοποιώντας την συνάρτηση `displayRemoveDrink()` του `EditDrinkController`.
- `onNewBottleClick()`: Η συνάρτηση αυτή εκτελείται όταν επιλεγεί το `NewBottleButton` χρησιμοποιώντας την συνάρτηση `displayNewDottle()` του `EditDrinkController`.
- `onRemoveBottleClick()`: Η συνάρτηση αυτή εκτελείται όταν επιλεγεί το `RemoveDrinkButton` χρησιμοποιώντας την συνάρτηση `displayRemoveBottle()` του `EditDrinkController`.
- `display()`: εμφανίζει το μενού.

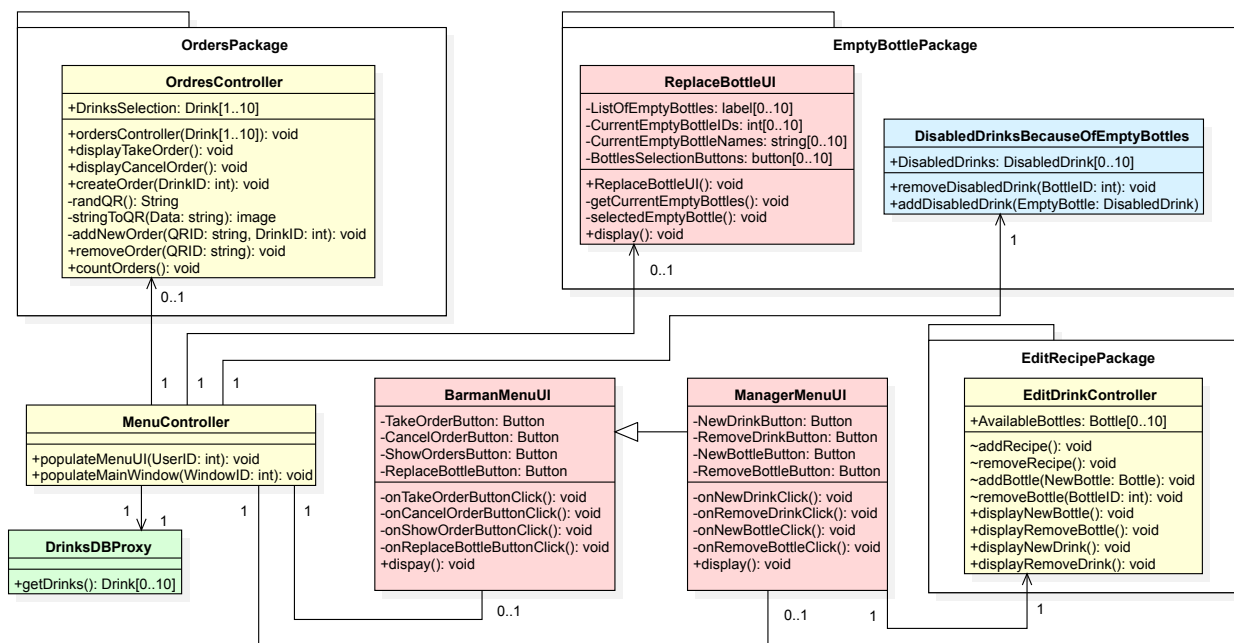
MenuController



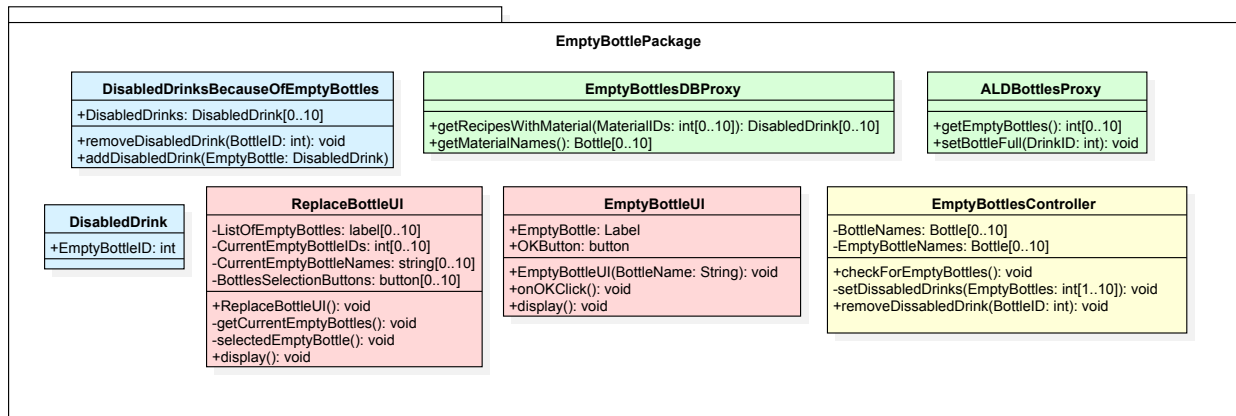
Η `MenuController` εμφανίζει το κατάλληλο μενού ανάλογα με τον χρήστη όπου έχει συνδεθεί, και εμφανίζει τις γραφικές διεπαφές του κεντρικού παραθύρου.

Μέθοδοι της κλάσης:

- `populateMenuUI(UserID: int)`: Δέχεται σαν όρισμα το ID του Manager ή του Μπάρμαν και δημιουργεί είτε ένα `BarmanMenuUI` είτε ένα `ManagerMenuUI`.
- `populateMainWindow(WindowID: int)`: Η συνάρτηση αυτή δέχεται σαν όρισμα ένα `WindowID` από το `BarmanMenuUI` και ζητάει την αντίστοιχη εκτέλεση UI. Επίσης μέσω των συσχετίσεων της με το `DrinksDBProxy` και `DisabledDrinksBecauseOfEmptyBottles` δίνει τα κατάλληλα ορίσματα στους constructors.

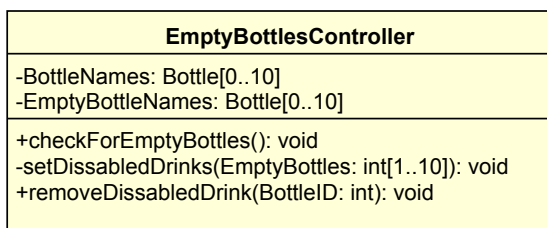


2.4 EmptyBottle Package



Το πακέτο EmptyBottle διαχειρίζεται τα άδεια μπουκάλια.

EmptyBottlesController



Η κλάση αυτή ελέγχει ποια ποτά απενεργοποιούνται λόγω άδειων μπουκαλιών.

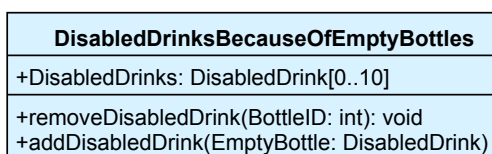
Χαρακτηριστικά της κλάσης:

- BottleNames: Bottle[0..10]: Η λίστα με τα ονόματα των Bottle.
- EmptyBottleNames: Bottle[0..10]: Η λίστα με τα άδεια Bottles.

Μέθοδοι της κλάσης:

- checkForEmptyBottle(): void: Ανανεώνει την EmptyBottleNames λίστα με την επιστροφή της getEmptyBottles() του ALDBottleProxy .
- setDisabledRecipe(EmptyBottles[1..10]): Εκτελεί την συνάρτηση addDisabledDrink() της κλάσης DisablesDrinksBecauseOfEmptyBottles με τα κατάλληλα ορίσματα.
- removeDisabledDrink(): Εκτελεί την συνάρτηση removeDisabledDrink() της κλάσης DisablesDrinksBecauseOfEmptyBottles με τα κατάλληλα ορίσματα.

DisabledDrinksBecauseOfEmptyBottles



Το αντικείμενο των απενεργοποιημένων ποτών λόγω άδειων μπουκαλιών.

Χαρακτηριστικά της κλάσης:

- DisabledDrinks: Η λίστα με τα DisabledDrink .

Μέθοδοι της κλάσης:

- removeDisabledDrink(BottleID:int): Η συνάρτηση αυτή παίρνει ως όρισμα το BottleID και το διαγράφει από το DisabledDrinks .
- addDisabledDrink(EmptyBottle:DisabledDrink): Η συνάρτηση αυτή παίρνει ως όρισμα ένα EmptyBottle και το προσθέτει στο DisabledDrinks.

DisabledDrink

DisabledDrink
+EmptyBottleID: int

Το αντικείμενο των ποτών που απενεργοποιούνται, κληρονομεί όλα τα χαρακτηριστικά του Drink με προσθήκη ενός στοιχείου BottleID.

Χαρακτηριστικά της κλάσης:

- EmptyBottleID: Ο χαρακτηριστικός αριθμός κάθε Bottle που τελειώνει και δεν είναι δυνατή η παρασκευή του ποτού με DrinkID.

EmptyBottlesDBProxy

EmptyBottlesDBProxy
+getRecipesWithMaterial(MaterialIDs: int[0..10]): DisabledDrink[0..10] +getMaterialNames(): Bottle[0..10]

Η βάση δεδομένων που αποθηκεύονται τα άδεια μπουκάλια.

Μέθοδοι της κλάσης:

- getRecipesWithMaterial(MaterialIDs: int[0..10]): DisabledDrink[0..10]: Η συνάρτηση αυτή παίρνει ως όρισμα τα MaterialIDs και επιστρέφει τα τα αντίστοιχα αντικείμενα DisabledDrink.
- getMaterialNames(): Bottle[0..10]: Η συνάρτηση αυτή επιστρέφει λίστα με όλα τα καταχωρημένα Bottle.

ReplaceBottleUI

ReplaceBottleUI
-ListOfEmptyBottles: label[0..10] -CurrentEmptyBottleIDs: int[0..10] -CurrentEmptyBottleNames: string[0..10] -BottlesSelectionButtons: button[0..10]
+ReplaceBottleUI(): void -getCurrentEmptyBottles(): void -selectedEmptyBottle(): void +display(): void

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον που εμφανίζεται όταν επιλεγεί η αντικατάσταση ενός μπουκαλιού.

Χαρακτηριστικά της κλάσης:

- `ListOfEmptyBottles`: Η λίστα με τα ονόματα των `EmptyBottles`.
- `CurrentEmptyBottleIDs`: Η λίστα με τα `BottleID` των `EmptyBottle`.
- `CurrentEmptyBottleNames`: Η λίστα με τα `BottleName` των `EmptyBottle`.
- `BottlesSelectionButtons`: Τα buttons επιλογής των `EmptyBottles`.

Μέθοδοι της κλάσης:

- `ReplaceBottleUI()`: Ο δομητής της κλάσης
- `getCurrentEmptyBottles()`: Η συνάρτηση αυτή επιστρέφει τα `EmptyBottles` διαβάζοντας το `EmptyBottleNames` του `EmptyBottlesController`
- `selectedEmptyBottle()`: Η συνάρτηση αυτή εκτελείται όταν πατηθεί ένα button από την `BottleSelectionButton`
- `display()`: Η συνάρτηση αυτή εμφανίζει την διεπαφή.

EmptyBottleUI

EmptyBottleUI
+EmptyBottle: Label +OKButton: button
+EmptyBottleUI(BottleName: String): void +onOKClick(): void +display(): void

Το γραφικό περιβάλλον που εμφανίζεται ως προς ενημέρωση πως ένα μπουκάλι είναι άδειο.

Χαρακτηριστικά της κλάσης:

- `EmptyBottle`: Το όνομα του άδειου Bottle.
- `OKButton`: Το button "OK".

Μέθοδοι της κλάσης:

- `EmptyBottleUI(BottleName: String)`: Δημιουργεί το κατάλληλο `EmptyBottle` label από το string του ορίσματος.
- `onOkClick()`: Τερματίζει άμεσα.
- `display()`: Εμφανίζει την γραφική διεπαφή και τερματίζει όταν πατηθεί το κουμπί OK.

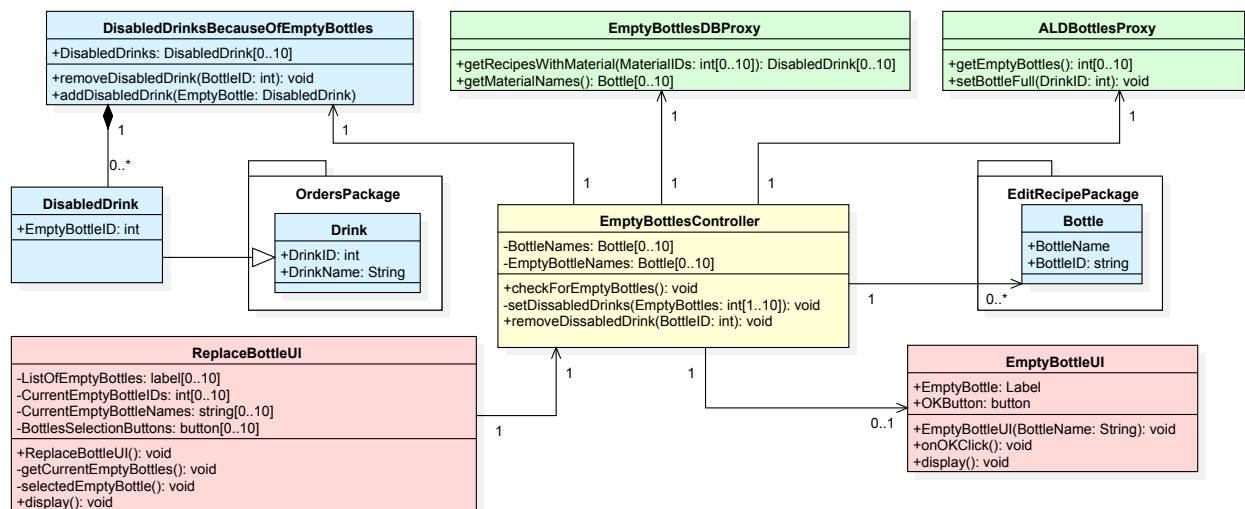
ALDBottlesProxy

ALDBottlesProxy
+getEmptyBottles(): int[0..10] +setBottleFull(DrinkID: int): void

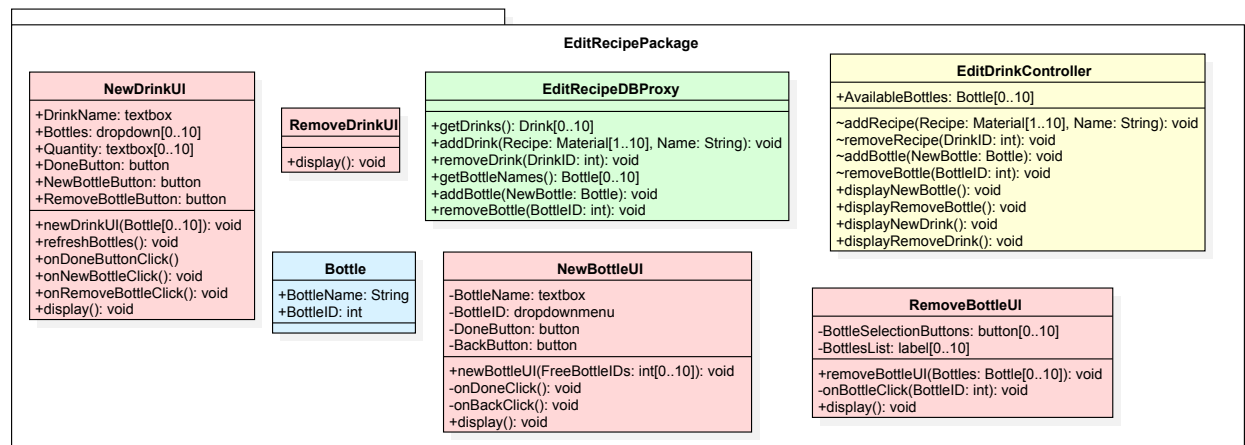
Διασύνδεση του συστήματος ανίχνευσης άδειων μπουκαλιών του ALD.

Μέθοδοι της κλάσης:

- `getEmptyBottles()`: Η συνάρτηση αυτή επιστρέφει μια λίστα με τα `EmptyBottles`.
- `setBottleFull(BottleID: int)`: Η συνάρτηση αυτή δηλώνει ένα `Bottle` ως γεμάτο.

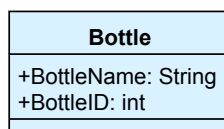


2.5 EditRecipe Package



Το EditRecipe package περιέχει τις κατάλληλες κλάσεις για την προσθαφαίρεση ποτών και μπουκαλιών από την βάση δεδομένων.

Bottle



Συσχέτιση του ID ενός μπουκαλιού με το όνομα του.

Χαρακτηριστικά της κλάσης:

- BottleName: string: Το όνομα ενός μπουκαλιού
- BottleID: int: Ο χαρακτηριστικός αριθμός ενός μπουκαλιού.

NewDrinkUI

NewDrinkUI
+DrinkName: textbox +Bottles: dropdown[0..10] +Quantity: textbox[0..10] +DoneButton: button +NewBottleButton: button +RemoveBottleButton: button
+newDrinkUI(Bottle[0..10]): void +refreshBottles(): void +onDoneButtonClick() +onNewBottleClick(): void +onRemoveBottleClick(): void +display(): void

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον για την προσθήκη ενός νέου ποτού.

Χαρακτηριστικά της κλάσης:

- DrinkName: Το όνομα του νέου ποτού
- Bottles: dropdown[0..10]: Μενού πολλαπλών επιλογών όπου περιέχουν τα ονόματα όλων των διαθέσιμων μπουκαλιών,
- Quantity: textbox[0..10]: Η ποσότητα του επιλεγμένου μπουκαλιού όπου απαιτείται για αυτό το ποτό.
- DoneButton: Κουμπί συνέχειας.
- NewBottleButton: Κουμπί για την παραπομπή προσθήκης νέου μπουκαλιού
- RemoveBottleButton: Κουμπί για την παραπομπή αφαίρεσης μπουκαλιού

Μέθοδοι της κλάσης:

- newDrinkUI(Bottle[0..10]): void: Συμπληρώνει τα Bottles dropdownmenus με τα ονόματα από τα ορίσματα.
- refreshBottles(): void: Η συνάρτηση αυτή ανανεώνει το AvailableBottles όταν εκτελεστούν οι RemoveBottleUI και NewBottleUI
- onDoneButtonClick(): void: Εκτελεί την addRecipe() του EditDrinkController.
- onNewBottleClick(): void: Εκτελεί την displayNewBottle() του EditDrinkController
- onRemoveBottleClick(): void: Εκτελεί την displayRemoveBottle() του EditDrinkController
- display(): void: Εμφανίζει την διεπαφή.

RemoveDrinkUI

RemoveDrinkUI
+display(): void

Υλοποίηση της SelectDrinkUI όπου επικοινωνεί με τον EditDrinkController.

Μέθοδοι της κλάσης:

- display(): void: Εκτελεί την removeRecipe() του OrdersController με όρισμα την επιστροφή του onDrinkClick().

NewBottleUI

NewBottleUI
-BottleName: textbox -BottleID: dropdownmenu -DoneButton: button -BackButton: button
+newBottleUI(FreeBottleIDs: int[0..10]): void -onDoneClick(): void -onBackClick(): void +display(): void

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον που εμφανίζεται όταν επιλεγεί Προσθήκη Νέου Μπουκαλιού

Χαρακτηριστικά της κλάσης:

- BottleName: textbox: Το όνομα του νέου Bottle
- BottleID: dropdownmenu: Ο χαρακτηριστικός αριθμός του νέου Bottle
- DoneButton: button: Το button επιβεβαίωσης
- BackButton: button: Το button "πίσω"

Μέθοδοι της κλάσης:

- newBottleUI(FreeBottleIDs: int[0..10]): void: Συμπληρώνει τα διαθέσιμα BottleIDs στο dropdownmenu.
- onDoneClick(): Εκτελεί την addBottle() του EditDrinkController με ορίσματα την εισαγωγή από την διεπαφή.
- onBackClick(): Τερματίζει άμεσα.
- display(): Εμφανίζει την διεπαφή.

RemoveBottleUI

RemoveBottleUI
-BottleSelectionButtons: button[0..10] -BottlesList: label[0..10]
+removeBottleUI(Bottles: Bottle[0..10]): void -onBottleClick(BottleID: int): void +display(): void

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον που εμφανίζεται όταν επιλεγεί Διαγραφή Ποτού

Χαρακτηριστικά της κλάσης:

- BottleSelectionButtons: button[0..10]: Κουμπιά επιλογής μπουκαλιού.
- BottlesList: label[0..10]: Τα ονόματα των μπουκαλιών.

Μέθοδοι της κλάσης:

- removeBottleUI(Bottles: Bottle[0..10]): void: Συμπληρώνει τα ονόματα των bottles στα labels.
- onBottleClick(): Εκτελεί την removeBottle του EditDrinkController με όρισμα το id του μπουκαλιού όπου επιλέχθηκε.
- display(): void: Εμφανίζει την διεπαφή.

EditRecipeDBProxy

EditRecipeDBProxy
+getDrinks(): Drink[0..10] +addDrink(Recipe: Material[1..10], Name: String): void +removeDrink(DrinkID: int): void +getBottleNames(): Bottle[0..10] +addBottle(NewBottle: Bottle): void +removeBottle(BottleID: int): void

Συναρτήσεις για την προσθήκη και αφαίρεση ποτών και μπουκαλιών από την βάση δεδομένων.

Μέθοδοι της κλάσης:

- getDrinks(): Drink[0..10]: Επιστρέφει τα ονόματα και IDs όλων των ποτών στην βάση δεδομένων.
- addDrink(Recipe: Material[1..10], Name: String): void: Προσθέτει ένα ποτό στην βάση δεδομένων με στοιχεία του ορίσματος.
- removeDrink(DrinkID: int): void: Διαγράφει ένα ποτό από την βάση δεδομένων βάσει του ID του.
- getBottleNames(): Bottle[0..10]: Επιστρέφει τα ονόματα και IDs όλων των μπουκαλιών στην βάση δεδομένων.
- addBottle(NewBottle: Bottle): void: Προσθέτει ένα μπουκάλι στην βάση δεδομένων με στοιχεία του ορίσματος.
- removeBottle(BottleID: int): void : Διαγράφει ένα μπουκάλι από την βάση δεδομένων βάσει του ID του.

EditDrinkController

EditDrinkController
+AvailableBottles: Bottle[0..10] ~addRecipe(Recipe: Material[1..10], Name: String): void ~removeRecipe(DrinkID: int): void ~addBottle(NewBottle: Bottle): void ~removeBottle(BottleID: int): void +displayNewBottle(): void +displayRemoveBottle(): void +displayNewDrink(): void +displayRemoveDrink(): void

Περιέχει τις κατάλληλες διασυνδέσεις και συναρτήσεις για την προσθαφαίρεση ποτών και μπουκαλιών από την βάση δεδομένων.

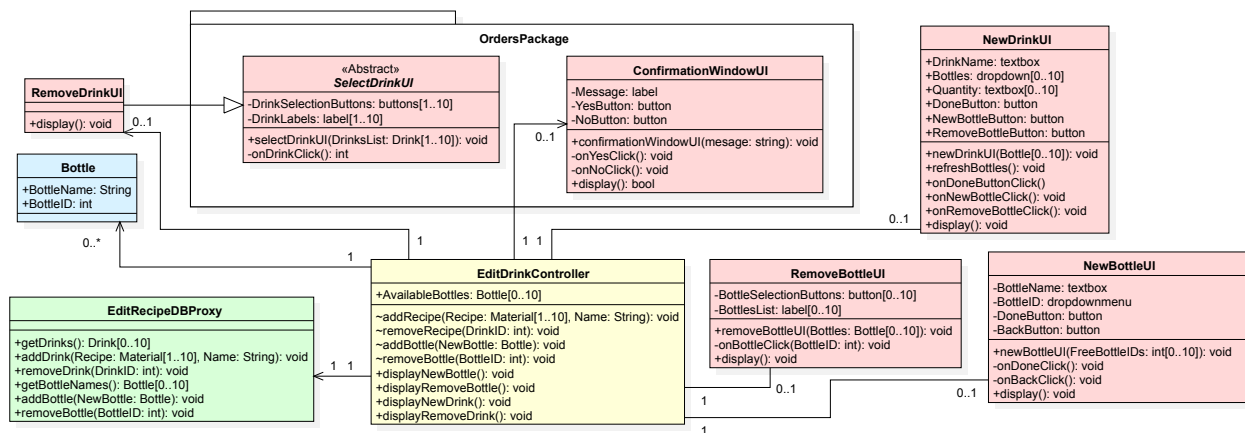
Χαρακτηριστικά της κλάσης:

- AvailableBottles: Η λίστα με όλα τα διαθέσιμα bottles.

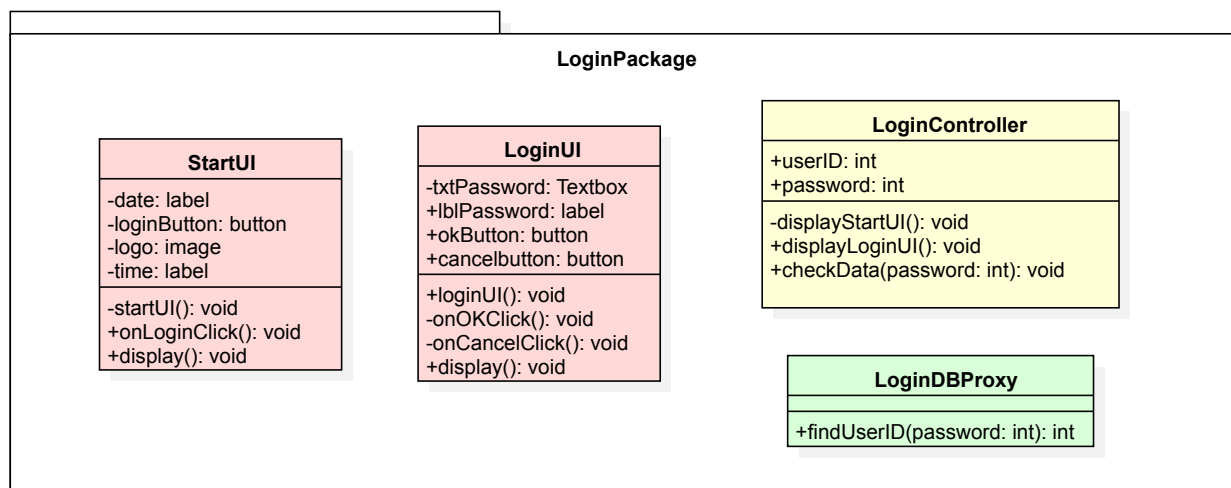
Μέθοδοι της κλάσης:

- addRecipe(Recipe: Material[1..10], Name: String): void: Εκτελεί την addDrink του EditRecipeProxy μεταφέροντας τα ορίσματα του.
- removeRecipe(DrinkID: int): void: Εκτελεί την removeDrink του EditRecipeProxy μεταφέροντας τα ορίσματα του.
- addBottle(NewBottle: Bottle): void: Εκτελεί την addBottle του EditRecipeProxy μεταφέροντας τα ορίσματα του.
- removeBottle(BottleID: int): void: Εκτελεί την removeBottle του EditRecipeProxy μεταφέροντας τα ορίσματα του.
- displayNewBottle(): void: Εμφανίζει την γραφική διεπαφή NewBottleUI.

- `displayRemoveBottle(): void`: Εμφανίζει την γραφική διεπαφή `RemoveBottleUI`.
- `displayNewDrink(): void`: Εμφανίζει την γραφική διεπαφή `NewDrinkUI`.
- `displayRemoveDrink(): void`: Εμφανίζει την γραφική διεπαφή `RemoveDrinkUI`.

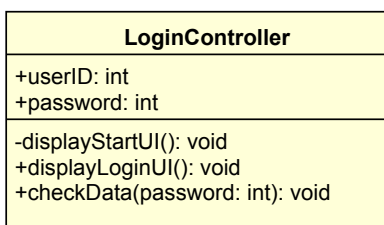


2.6 Login Package



Το πακέτο Login περιέχει τις απαραίτητες κλάσεις ώστε να γίνει η σύνδεση χρήστη.

LoginController



Η κλάση αυτή ελέγχει τη διαδικασία της σύνδεσης χρήστη ζητώντας κωδικό, ελέγχοντας την ορθότητά του και ανακατευθύνοντας στο κατάλληλο μενού.

Χαρακτηριστικά της κλάσης:

- userID: Ο χαρακτηριστικός αριθμός κάθε χρήστη.
- password: Ο κωδικός κάθε χρήστη.

Μέθοδοι της κλάσης:

- displayStartUI(): Η συνάρτηση αυτή εμφανίζει την κλάση StartUI.
- displayLoginUI(): Η συνάρτηση αυτή εμφανίζει την κλάση LoginUI.
- checkData(password: int): Η συνάρτηση αυτή παίρνει ως όρισμα το password και εκτελεί την findUserID() της LoginDBProxy αν είναι true ή την ErrorUI αν είναι false.

LoginUI

LoginUI
-txtPassword: Textbox +lblPassword: label +okButton: button +cancelbutton: button
+loginUI(): void -onOKClick(): void -onCancelClick(): void +display(): void

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον της οθόνης σύνδεσης.

Χαρακτηριστικά της κλάσης:

- txtPassword: Το πεδίο συμπλήρωσης password.
- lblPassword: Η επικεφαλίδα του txtPassword.
- okButton: Το button OK.
- cancelButton: Το button "Ακύρωση" .

Μέθοδοι της κλάσης:

- loginUI(): Ο δομητής της κλάσης.
- okClick(): Η συνάρτηση αυτή εκτελείται όταν πατηθεί το okButton.
- onCancelClick(): Η συνάρτηση αυτή εκτελείται όταν πατηθεί το cancelButton.
- display(): Εμφανίζει το παράθυρο.

StartUI

StartUI
-date: label -loginButton: button -logo: image -time: label
-startUI(): void +onLoginClick(): void +display(): void

Η κλάση αυτή περιγράφει το γραφικό περιβάλλον της αρχικής οθόνης.

Χαρακτηριστικά της κλάσης:

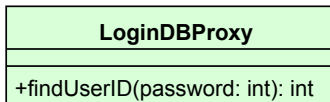
- data: Η επικεφαλίδα της κλάσης.
- loginButton: Το button "Login".
- logo: Το logo της εφαρμογής.

- time: Η ημερομηνία και ώρα.

Μέθοδοι της κλάσης:

- StartUI(): Ο δομητής της κλάσης.
- onLoginClick(): Η συνάρτηση αυτή εκτελείται όταν πατηθεί το button loginButton.
- display(): Εμφανίζει το παράθυρο.

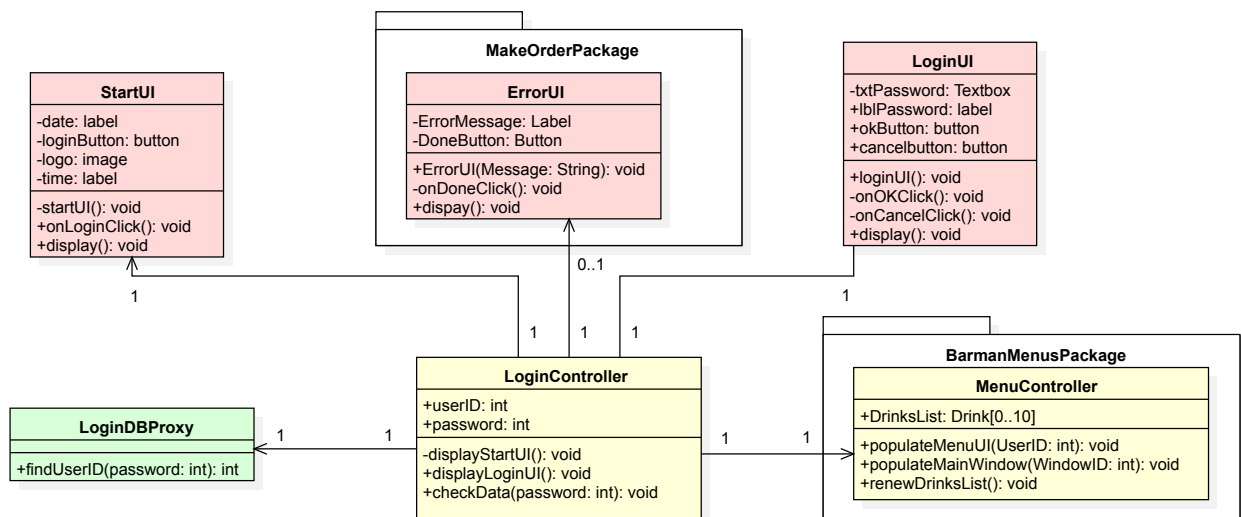
LoginDBProxy



Η κλάση αυτή περιγράφει την βάση δεδομένων που αποθηκεύονται οι χρήστες με τους αντίστοιχους κωδικούς.

Μέθοδοι της κλάσης:

- findUserID(password: string): Η συνάρτηση αυτή αντιστοιχίζει το password με το κατάλληλο UserID.



3 Μη λειτουργικές απαιτήσεις

3.1 Απαιτήσεις επίδοσης

<ΜΛΑ-1> Το Σύστημα πρέπει να εκτελεί μια συνταγή το πολύ μέχρι 2 λεπτά.

Περιγραφή: Ο χρόνος όπου απαιτείται από την στιγμή εκκίνησης της παρασκευής μιας παραγγελίας, μέχρι την στιγμή όπου ο Πελάτης έχει την δυνατότητα να παραλάβει την παραγγελία του, θα πρέπει να είναι το πολύ μέχρι 2 λεπτά.

User Priority (5/5): Ο χρόνος αναμονής είναι πολύ σημαντικός παράγοντας για όλους τους χρήστες καθώς ένας μεγάλος χρόνος αναμονής θα αποτρέψει τους Πελάτες από το να χρησιμοποιούν το σύστημα.

Technical Priority (1/5): Ο χρόνος αναμονής δεν επηρεάζει την εύλογη τεχνική λειτουργία του Συστήματος.

Stability (4/5): Ο χρόνος εκτέλεσης μιας συνταγής είναι σημαντικό να παραμείνει σύντομος και η πιθανότητα να αλλάξει είναι μικρή.

<ΜΛΑ-4> Το Σύστημα πρέπει να μπορεί να παρασκευάζει το ποτό με πιθανότητα αποτυχίας κάτω από 1%.

Περιγραφή: Οι αστοχίες από τα εξωτερικά συστήματα ΝΑΟ και ALD θα πρέπει συνδυαστικά να μην προκαλούν οποιαδήποτε είδους αποτυχία σε ποσοστό πάνω από 1%.

User Priority (5/5): Όλοι οι χρήστες του συστήματος επιθυμούν μικρά ποσοστά αποτυχίας. Οι μεν Πελάτες ωφελούνται με μικρότερο χρόνο αναμονής και ο Μπάρμαν καλείται να εμπλακεί στην επίλυση της αποτυχίας λιγότερο.

Technical Priority (3/5): Με την ελαχιστοποίηση των πιθανών σεναρίων αποτυχίας παραλείπεται η ανάγκη για την ανάπτυξη μεθόδων επίλυσης τους.

Stability (4/5): Η πιθανότητα αποτυχίας στην παρασκευή ενός ποτού είναι σημαντικό να είναι χαμηλή.

3.2 Απαιτήσεις Αξιοπιστίας

<ΜΛΑ-2> Το Σύστημα πρέπει να εκτελεί τις παραγγελίες με σειρά.

Περιγραφή: Ο Μπάρμαν πρέπει να έχει την δυνατότητα να αιτηθεί κι άλλες παραγγελίες όσο το Σύστημα βρίσκεται σε στάδιο παρασκευής.

User Priority (4/5): Είναι ευκολότερο για τον Μπάρμαν να προσθέτει παραγγελίες σε μια ουρά της στιγμή όπου έρχονται, παρά να πρέπει να περιμένει την ολοκλήρωση της προηγούμενης παραγγελίας για να αιτηθεί την επόμενη.

Technical Priority (3/5): N/A

Stability (5/5): Η απαίτηση αυτή δεν θα μεταβληθεί στο μέλλον καθώς αποτελεί βάση στην οποία λειτουργεί το σύστημα παραγγελιών.

<ΜΛΑ-3> Το Σύστημα πρέπει να μπορεί να δεχθεί μέχρι και 40 παραγγελίες.

Περιγραφή: Πρέπει να μπορούν να προστεθούν στην λίστα εκκρεμών παραγγελιών μέχρι 40 παραγγελίες.

User Priority (3/5): Μία μεγάλη ουρά παραγγελιών είναι επιθυμητή από τον Μπάρμαν.

Technical Priority (3/5): Χωρίς μια μεγάλη λίστα εκκρεμών παραγγελιών, θα υπάρξει υπερχειλίση και δεν θα πραγματοποιηθούν όλες οι παραγγελίες.

Stability (3/5): Υπάρχει η δυνατότητα μεταβολής του μέγιστου αριθμού παραγγελιών στο μέλλον.

3.3 Απαιτήσεις Χρηστικότητας

<ΜΛΑ-5> Το Σύστημα πρέπει να έχει γραφικό περιβάλλον εύχρηστο σε οθόνη αφής .

Περιγραφή: Για την ευχρηστία του συστήματος στο στενό χώρο ενός μπαρ, η ανάγκη ύπαρξης ποντικιού και εξωτερικού πληκτρολογίου δεν είναι επιθυμητή. Γι' αυτό τον λόγο, το γραφικό περιβάλλον, θα πρέπει να

είναι δομημένο με τρόπο εύχρηστο για την λειτουργία του σε οθόνη αφής.

User Priority (3/5): Μόνο ένας χρήστης, ο Μπάρμαν έχει να ωφεληθεί από αυτό.

Technical Priority (0/5): N/A

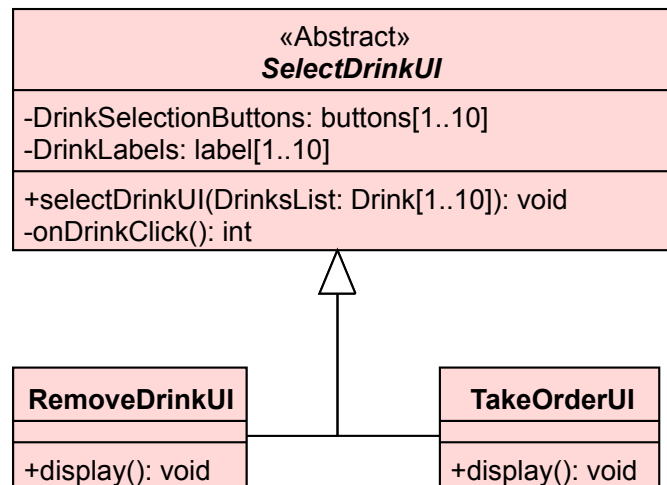
Stability (5/5): Λαμβάνοντας υπόψιν το γεγονός πως η εφαρμογή είναι σχεδιασμένη για λειτουργία σε οθόνη αφής, αυτή η μη-λειτουργική απαίτηση δεν ενδέχεται να αλλάξει.

4 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

4.1 Πρότυπα συμπεριφοράς

Template Method Pattern

Σε δύο σημεία του συστήματος απαιτούνταν η επιλογή ενός ποτού από μια λίστα επιλογών και η εκτέλεση διαφορετικών ενεργειών ανάλογα με την περίπτωση. Από την στιγμή όπου η λειτουργικότητα είναι η ίδια με διαφορά σε μία συνάρτηση, χρησιμοποιήθηκε το template pattern για την δημιουργία μιας abstract κλάσης όπου λειτουργεί ως το template και οι υλοποιήσεις των διαφορών τους σε concrete implementations.



Χρησιμοποιώντας το template method Pattern στην προκειμένη περίπτωση επιτεύχθηκε μείωση του μεγέθους του κώδικα αποφεύγοντας τις επαναλήψεις. Επιπρόσθετα, μειώνεται η πιθανότητα εμφάνισης λαθών και διευκολύνεται η αποσφαλμάτωση και η εφαρμογή αλλαγών σε αυτό το τμήμα του κώδικα.

Πρότυπα όπου δεν χρησιμοποιήθηκαν

Η μελέτη των διαγραμμάτων κλάσεων αυτού του συστήματος δεν παρουσίασε ευκαιρίες για την εφαρμογή άλλων προτύπων. Οι ΜΛΑ όπου τέθηκαν στο έγγραφο απαιτήσεων χρηστών είναι τέτοιες όπου δεν συνδυάζονται με κάποιον ξεκάθαρο τρόπο με κάποιο πρότυπο. Θα μπορούσε να είχε εφαρμοστεί το bridge pattern στην δόμηση των NAOProxy και ALDProxy, όμως από την στιγμή όπου δεν είχε τεθεί απαίτηση επεκτασιμότητας, η εφαρμογή του προτύπου θα ήταν πλεονάζουσα.

Επιπλέον αναζήτηση σε online βιβλιογραφία δεν έφερε νέες ιδέες για άλλα πρότυπα όπου θα μπορούσαν να χρησιμοποιηθούν. Πιθανότατα να υπάρχει κάποιο pattern για τα UI, αλλά ο χρόνος όπου αφιερώθηκε στην αναζήτηση δεν επαρκούσε για την κατανόηση τους.

A Λίστα Ικνηλασιμότητας

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

- Το σύστημα όταν ο χρήστης επιλεγεί την ενημερώσει εκκρεμών παραγγελιών εμφανίζει μαζί με τις εκκρεμείς παραγγέλσεις και το QRID τις κάθε παραγγελίας(ΣΧ5 στο έγγραφο απαιτήσεων χρηστών).
- Το σύστημα δίνει τη δυνατότητα στο χρηστή να προσθέσει ή να αφαιρέσει κάποιο υλικό . Η δυνατότητα αυτή είχε αναφερθεί στα ανοιχτά θέματα του έγγραφου απαιτήσεων χρηστών .
- Όταν αδειάσει ένα μπουκάλι το σύστημα αφαιρεί όλα τις συνταγές που περιέχουν το υλικό αυτό από την λίστα επιλογής πότου . Στο έγγραφο απαιτήσεων χρηστών το είχε περιγραφεί ότι το σύστημα απλά ενημέρωνε το χρηστή για τη μη διαθεσιμότητα του πότου(ΣΧ3) .

B Ανοιχτά Θέματα

- Σε ορισμένες περιπτώσεις υπάρχει πιθανότητα ανάγκης για προσθήκη επιπλέον μεαβλητών ορισμάτων ή και κλάσεων. Αυτό θα γίνει πιο εμφανές κατά τη διαδικασία της συγγραφής κώδικα, όπου αναμένεται να προκύψουν νέες ανάγκες.
- Το σύνολο των αλγορίθμων που θα υλοποιήσουν το σύστημα θα μελετώνται και θα βελτιώνονται συνεχώς ώστε να γίνεται συνεχώς αποδοτικότερο.
- Η παρατήρηση και η διεξοδική μελέτη του τρόπου λειτουργίας του συστήματος ενδέχεται να οδηγήσει στην εμφάνιση καινούργιων αναγκών, απαιτήσεω λογισμικού που είναι πιθανό να οδηγήσουν σε ανάγκες ανασχεδιασμού τμημάτων του έργου. Συνεπώς, κρίνεται αναγκαίο να υπάρχει συνεχής αλληλεπίδραση ανάμεσα σε χρήστες και δημιουργούς για την διόρθωση λαθών και την βελτίωση του συστήματος.