



Τεχνολογία Λογισμικού
Τομέας Ηλεκτρονικής και Υπολογιστών
Τμήμα ΗΜΜΥ
Α.Π.Θ.

8^ο Εξάμηνο
Άνοιξη 2019



AUTH CLOUD

Change the way you cloud

Απαιτήσεις Λογισμικού

Del.1.2

Version 0.9

Κ. Λέτρος 8851, konsletr@ece.auth.gr
Δ. Καρατζά 8828, dimitra.karatza11@gmail.com
Α. Μυρσινιάς 8873, antonis.myrsinias@gmail.com
Ι. Μανουσαρίδης 8855, giannismanu97@gmail.com

05/05/2019

**Μέλη της Ομάδας Ανάπτυξης**

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Καρατζά Δήμητρα	Auth Cloud	dimitra.karatza11@gmail.com
Λέτρος Κωνσταντίνος	Auth Cloud	konsletr@ece.auth.gr
Μανουσαρίδης Ιωάννης	Auth Cloud	giannismanu97@gmail.com
Μυρσινιάς Αντώνιος	Auth Cloud	antonis.myrsinias@gmail.com

Ιστορικό αλλαγών

Όνομα	Ημερομηνία	Αλλαγή	Έκδοση
Α. Συμεωνίδης	*	Δημιουργία προτύπου εγγράφου	0.0.1
Auth Cloud	02/04	Καθορισμός πακέτων	0.1
Auth Cloud	07/04	Αρχή συγγραφής του πακέτου LoginPackage	0.2
Auth Cloud	09/04	Αρχή συγγραφής του πακέτου PagePackage	0.3
Auth Cloud	13/04	Αρχή συγγραφής του πακέτου FilePackage	0.4
Auth Cloud	17/04	Αρχή συγγραφής Μη Λειτουργικών Απαιτήσεων	0.5
Auth Cloud	20/04	Συγγραφή Εισαγωγικών	0.6
Auth Cloud	23/04	Αρχή συγγραφής Προτύπων Σχεδίασης	0.7
Auth Cloud	02/05	Συγγραφή Παραρτημάτων Ι, ΙΙ	0.8



Πίνακας Περιεχομένων

1 Εισαγωγικά	5
1.1 Στόχος του Εγγράφου	5
1.2 Αναγνωστικό κοινό και τρόπος ανάγνωσης	6
1.3 Σκοπός του Έργου	6
2. Στατική Μοντελοποίηση	6
2.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας	6
2.1.1 LoginPackage	6
2.1.2 PagePackage	12
2.1.3 FilePackage	24
3. Μη λειτουργικές απαιτήσεις	36
3.1 Απαιτήσεις προϊόντος	36
3.1.1 Απαιτήσεις απόδοσης	36
3.1.2 Απαιτήσεις αξιοπιστίας	36
3.1.3 Απαιτήσεις μεταφερσιμότητας	37
3.2 Εξωτερικές απαιτήσεις	37
3.2.1 Απαιτήσεις ιδιωτικότητας	37
4. Πρότυπα Σχεδιασμού που υιοθετήθηκαν	38
4.1 Δομικά Πρότυπα	38
4.1.1 Object Adapter Design Pattern	38
4.2 Δημιουργικά Πρότυπα	39
4.2.1 Singleton Design Pattern	39
4.3 Πρότυπα Συμπεριφοράς	40
4.3.1 Memento Design Pattern	41
Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας	42
Παράρτημα ΙΙ – Ανοιχτά Θέματα	43



Λίστα σχημάτων

Σχήμα 1. Διάγραμμα κλάσεων πακέτου LoginPackage.....	11
Σχήμα 2. Διάγραμμα κλάσεων πακέτου PagePackage.....	23
Σχήμα 3. Διάγραμμα κλάσεων πακέτου FilePackage.....	35
Σχήμα 4: Adapter Design Pattern.....	38
Σχήμα 5: Adapter Pattern για την αρχική κλάση DevicePageHandlerProxy.....	39
Σχήμα 6: Adapter Pattern για την αρχική κλάση DownloadDeviceProxy.....	39
Σχήμα 7: Singleton Pattern Design Pattern.....	40
Σχήμα 8: Το πρότυπο Singleton όπως εφαρμόστηκε στις οριακές κλάσεις.....	40
Σχήμα 9: Memento Design Pattern.....	41
Σχήμα 10: Memento Pattern για την αρχική κλάση PageController.....	41



1 Εισαγωγικά

Τα πνευματικά δικαιώματα του παρόντος εγγράφου κατοχυρώνονται σύμφωνα με το πρότυπο του © CC BY 4.0 license (Creative Commons BY license). Περισσότερες πληροφορίες στο <https://creativecommons.org/licenses/by/4.0/>.

1.1 Στόχος του Εγγράφου

Το παρόν έγγραφο αποτελεί μια σύντομη και αναλυτική περιγραφή των απαιτήσεων λογισμικού που διαμορφώνει το σύστημα. Αποτελεί βοήθημα για την ανάλυση των χαρακτηριστικών αυτού, με κύριο γνώμονα τις απαιτήσεις χρηστών που περιεγράφηκαν στο προηγούμενο έγγραφο.

Στόχος του εγγράφου είναι ο προσδιορισμός των λειτουργιών του συστήματος και η αλληλεπίδρασή του με τους χρήστες και τα συνεργαζόμενα εξωτερικά συστήματα. Αναλύονται το περιβάλλον διεπαφής καθώς και τα διαχειριζόμενα από το σύστημα δεδομένα, κατόπιν περιγραφής των τύπων δεδομένων, των σελίδων, των φορμών και των μηνυμάτων που επιστρέφει το σύστημα, μέσω των οποίων ανταλλάσσονται τα δεδομένα αυτά.

Για την επίτευξη αυτών των στόχων απαιτείται στατική μοντελοποίηση του συστήματος, η οποία σχετίζεται με την μοντελοποίηση των απαιτήσεων χρηστών. Το σύστημα χωρίζεται σε πακέτα κλάσεων με βάση τις επιμέρους λειτουργικότητες του, οπότε εξασφαλίζεται η ευκολότερη κατανόηση και ανάπτυξη του συστήματος.

Σε κάθε πακέτο περιλαμβάνεται ένα σύνολο κλάσεων, οι οποίες διακρίνονται σε κλάσεις οντοτήτων (entities), σε οριακές κλάσεις (boundaries) και σε κλάσεις ελέγχου (controllers). Οι κλάσεις οντοτήτων καθορίζουν τη μόνιμη πληροφορία του συστήματος. Οι οριακές κλάσεις ή κλάσεις διεπαφής χρηστών αναπαριστούν την αλληλεπίδραση μεταξύ του χρήστη και του συστήματος ή του συστήματος και άλλων εξωτερικών συστημάτων. Οι κλάσεις ελέγχου ή ρυθμιστικές κλάσεις είναι υπεύθυνες για τις διαδικασίες ελέγχου του συστήματος και το συντονισμό των διεργασιών των επιμέρους κλάσεων. Ακόμη, σχεδιάζονται διαγράμματα για την αναπαράσταση των αλληλεπιδράσεων των κλάσεων και των πακέτων. Τέλος, περιγράφονται τα πρότυπα σχεδιασμού που υιοθετήθηκαν με στόχο την απόδοση επιπλέον λειτουργικότητας στο σύστημα.

Ως εκ τούτου, το παρόν έγγραφο καθίσταται ορόσημο για όλους όσους εμπλέκονται στη σχεδίαση του λογισμικού.

Το παρόν έγγραφο χρησιμοποιεί γραμματοσειρά Calibri μεγέθους 11pt για το κυρίως κείμενο. Για τις επικεφαλίδες το μέγεθος είναι 24,18,16, 14 ή 12, ανάλογα με το βάθος στο οποίο βρίσκεται η κάθε μία. Με πλάγια γράμματα γίνεται η αναφορά σε άλλες μεθόδους, με τη μορφή `className::methodName(inputs):returnType`. Με έντονη γραμματοσειρά γίνεται η αναφορά στα χαρακτηριστικά και στις μεθόδους του συστήματος προς επεξήγηση. Τέλος, τα διαγράμματα και οι κλάσεις που παρουσιάζονται και αναλύονται ακολουθούν τις παραδοχές της γλώσσας UML.



1.2 Αναγνωστικό κοινό και τρόπος ανάγνωσης

Το έγγραφο απευθύνεται στις ακόλουθες ομάδες ανθρώπων:

- **Ομάδα ανάπτυξης:** Η μελέτη των απαιτήσεων του συστήματος από κάθε μέλος της ομάδας ανάπτυξης θα συμβάλλει στην ορθή κατανόηση του αντικειμένου με στόχο τη διευκόλυνση και τον επιτυχημένο σχεδιασμό, ανάπτυξη, δοκιμή και αξιολόγηση του συστήματος. Ανάλογα με τη δραστηριότητα ανάπτυξης, τα μέλη μπορούν να έχουν διαφορετικούς ρόλους. Πιο συγκεκριμένα, οι ρόλοι που απαρτίζουν την ομάδα ανάπτυξης είναι οι μηχανικοί πελάτη, οι αρχιτέκτονες, οι αναλυτές, οι σχεδιαστές αντικειμένων και συστημάτων, καθώς και οι προγραμματιστές.
- **Ομάδα συντήρησης:** Η πλήρης κατανόηση της λειτουργικότητας και των βασικών οντοτήτων που δημιουργήθηκαν κατά την υλοποίηση και τον προγραμματισμό του συστήματος είναι απαραίτητη για τη βελτιστοποίησή του από την ομάδα συντήρησης και αναβάθμισης.

1.3 Σκοπός του Έργου

Η προς ανάπτυξη εφαρμογή συνίσταται στην παροχή μιας υπηρεσίας αποθήκευσης αρχείων για κινητά και tablet, η οποία επιτρέπει τη χρήση αποθηκευτικού νέφους, τον διαμοιρασμό αρχείων και την συνεργατική επεξεργασία από τους χρήστες της Πανεπιστημιακής κοινότητας.

Στόχος είναι να προσφέρει μια εύχρηστη λύση στο χρήστη, ο οποίος επιθυμεί να εξασφαλίσει την ασφάλεια των αρχείων του και την εύκολη πρόσβασή του σε αυτά, οπουδήποτε κι αν βρίσκεται, από το κινητό ή το tablet του, αποφεύγοντας έτσι τη δέσμευση μνήμης χρήσιμης της συσκευής του. Με τη χρήση αποθηκευτικού νέφους -ενός δικτυακού μοντέλου αποθήκευσης δεδομένων- τα δεδομένα αποθηκεύονται σε απομακρυσμένες δικτυακές τοποθεσίες, σε μεγάλα κέντρα αποθήκευσης δεδομένων. Ο χρήστης μπορεί να έχει πρόσβαση σε αυτά μέσω κάποιας διαδικτυακής διεπαφής, στην προκειμένη περίπτωση κατόπιν σύνδεσης στο λογαριασμό του στην εφαρμογή.

Μετά τη σύνδεση στο λογαριασμό του, λοιπόν, ο χρήστης μπορεί να ανεβάζει τα επιθυμητά αρχεία, να τα κατεβάζει στη συσκευή του, να κάνει προεπισκόπηση, διαγραφή, μετονομασία και κοινοποίηση αυτών με κάποιον άλλον χρήστη. Ακόμη, έχει τη δυνατότητα να βλέπει ποια αρχεία έχουν μοιραστεί άλλοι χρήστες μαζί του, να αναζητά ένα αρχείο με βάση το όνομα, να αποσυνδέεται από το λογαριασμό του.

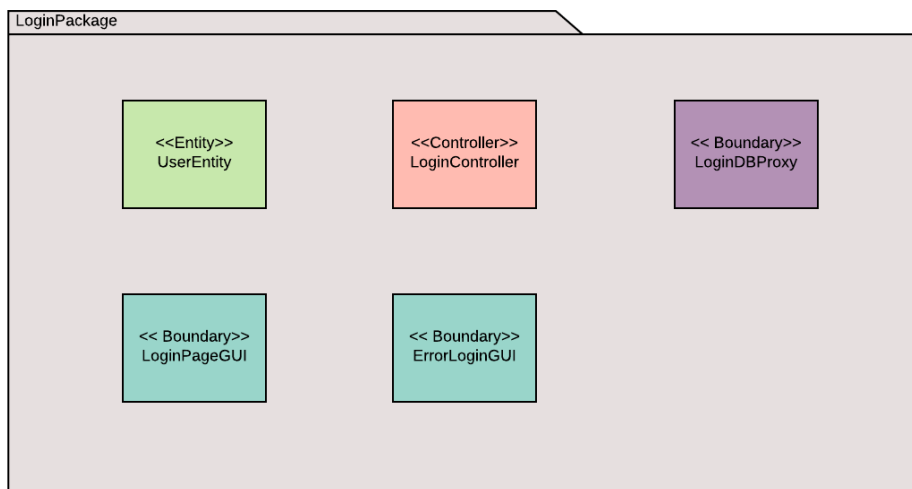
2. Στατική Μοντελοποίηση

2.1 Πακέτα λεξιλογίου σεναρίων υψηλής προτεραιότητας

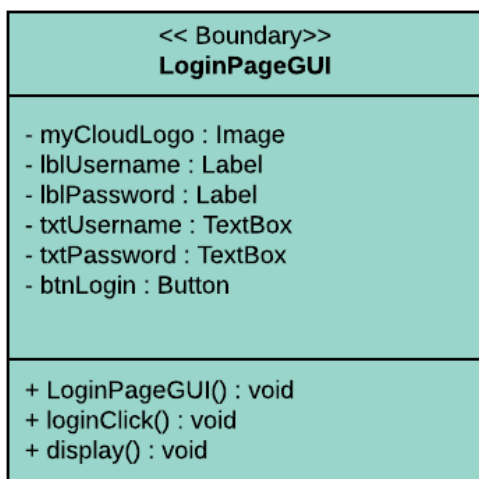
2.1.1 LoginPackage

2.1.1.1 Περιγραφή Πακέτου LoginPage:

Το πακέτο αυτό αναφέρεται στις επιλογές login χρήστη. Επιπρόσθετως, περιλαμβάνει την οθόνη που βλέπει ο χρήστης για την σύνδεση του στο σύστημα. Παρακάτω παρουσιάζονται οι κλάσεις που διαχειρίζονται τις παραπάνω επιλογές.



LoginPageGUI



Περιγραφή:

Η κλάση αυτή εκφράζει τη διεπαφή της σελίδας σύνδεσης του χρήστη στο σύστημα της εφαρμογής.

Χαρακτηριστικά της κλάσης:

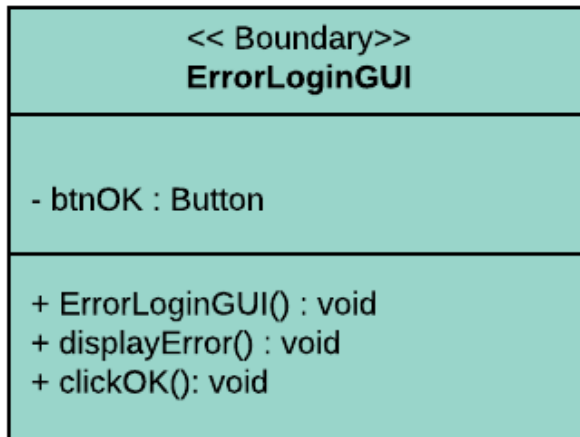
- **myCloudLogo** : Λογότυπο της εφαρμογής
- **lblUsername**: Επιγραφή αριστερά από το πεδίο συμπλήρωσης του ονόματος χρήστη.
- **lblPassword**: Επιγραφή αριστερά από το πεδίο συμπλήρωσης του κωδικού πρόσβασης.
- **txtUsername**: Πεδίο συμπλήρωσης για το όνομα του λογαριασμού του χρήστη.
- **txtPassword**: Πεδίο συμπλήρωσης για τον κωδικό πρόσβασης.
- **btnLogin**: Κουμπί για την είσοδο στο σύστημα της εφαρμογής.

Μέθοδοι της κλάσης:

- **LoginPageGUI()**: Η μέθοδος δόμησης της κλάσης LoginPageGUI.
- **loginClick()**: Η μέθοδος αυτή καλεί τη μέθοδο *LoginController::checkData(username: String, password: String):void* για τον έλεγχο των στοιχείων του χρήστη.
- **display()**: Η μέθοδος αυτή προβάλλει την διεπαφή LoginPage.



ErrorLoginGUI



Περιγραφή:

Πρόκειται για τη διεπαφή η οποία είναι υπεύθυνη για την επιστροφή των σφαλμάτων σύνδεσης του χρήστη.

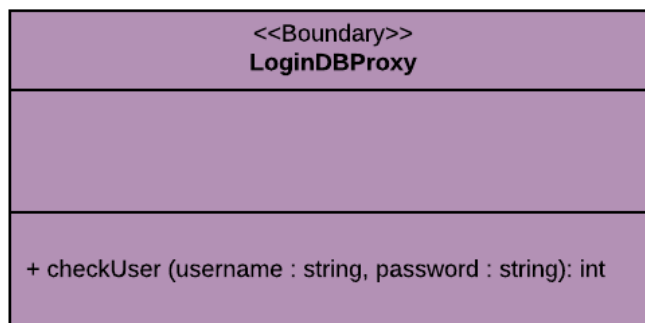
Χαρακτηριστικά της κλάσης:

- **btnOK** : Κουμπί για να επιβεβαιωθεί από το σύστημα, ότι ο χρήστης ότι είδε το σφάλμα και να κλείσει το παράθυρο.

Μέθοδοι της κλάσης:

- **ErrorLoginGUI()**: Η μέθοδος δόμησης της κλάσης ErrorLoginGUI.
- **displayError()**: Η μέθοδος αυτή είναι υπεύθυνη για την προβολή του μηνύματος σφάλματος “Λανθασμένη εισαγωγή ονόματος χρήστη ή κωδικού πρόσβασης” στον χρήστη.
- **clickOK()**: Η μέθοδος αυτή κλείνει το παράθυρο με το σφάλμα και καλεί τη μέθοδο *LoginController::displayLoginPage():void*.

LoginDBProxy



Περιγραφή:

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με τη βάση δεδομένων όπου είναι αποθηκευμένα τα στοιχεία του χρήστη.

**Μέθοδοι της κλάσης:**

- **checkUser(username: String, password: String):** Η μέθοδος αυτή ελέγχει αν τα στοιχεία που έχει δώσει ο χρήστης είναι ορθά. Σε περίπτωση ορθότητας επιστρέφεται το ID του χρήστη, ειδάλλως η τιμή 0.

UserEntity

<<Entity>> UserEntity	
- userID: int - username: String - password: String	
+ UserEntity(userID: int , username: String, password: String): void + setID(userID: int): void + setUsername(username: String): void + setPassword(password: String): void + getID(): int + getUsername(): String + getPassword(): String	

Περιγραφή:

Αυτή η κλάση υλοποιεί την οντότητα του χρήστη καθορίζοντας τα βασικά χαρακτηριστικά του, καθώς και τις μεθόδους που απαιτούνται.

Χαρακτηριστικά της κλάσης:

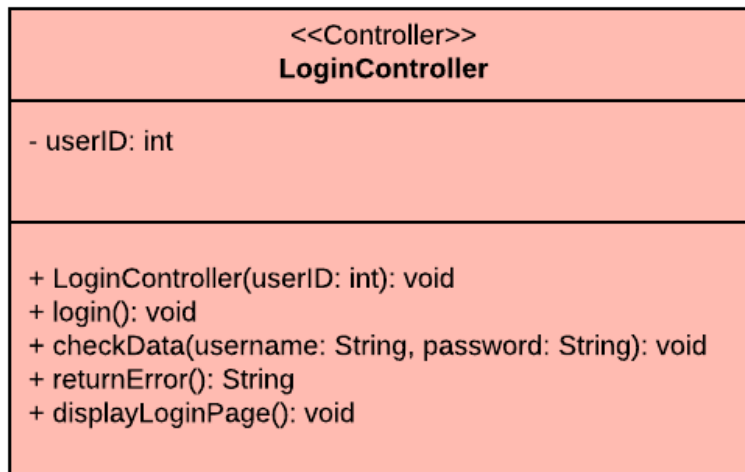
- **userID:** Ο μοναδικός χαρακτηριστικός αριθμός για κάθε χρήστη που τον ξεχωρίζει από τους υπόλοιπους.
- **username:** Το όνομα με το οποίο ο χρήστης συνδέεται στο σύστημα .
- **password:** Ο κωδικός πρόσβασης με τον οποίο ο χρήστης συνδέεται στο σύστημα.

Μέθοδοι της κλάσης:

- **UserEntity(userID: int , username: String, password: String):** Η μέθοδος δόμησης της κλάσης UserEntity.
- **setID(userID: int):** Η μέθοδος αυτή θέτει το userID του χρήστη σύμφωνα με το όρισμα που δέχεται.
- **setUsername(username: String):** Η μέθοδος αυτή θέτει το όνομα χρήστη σύμφωνα με το όρισμα που δέχεται.
- **setPassword(password: String):** Η μέθοδος αυτή θέτει τον κωδικό του χρήστη σύμφωνα με το όρισμα που δέχεται.
- **getID():** Η μέθοδος αυτή επιστρέφει το userID κάθε χρήστη.
- **getUsername():** Η μέθοδος αυτή επιστρέφει το username κάθε χρήστη.
- **getPassword():** Η μέθοδος αυτή επιστρέφει το password κάθε χρήστη.



LoginController



Περιγραφή:

Η κλάση αυτή αποτελεί έναν ελεγκτή, ο οποίος πραγματοποιεί την είσοδο του χρήστη στο σύστημα.

Χαρακτηριστικά της κλάσης:

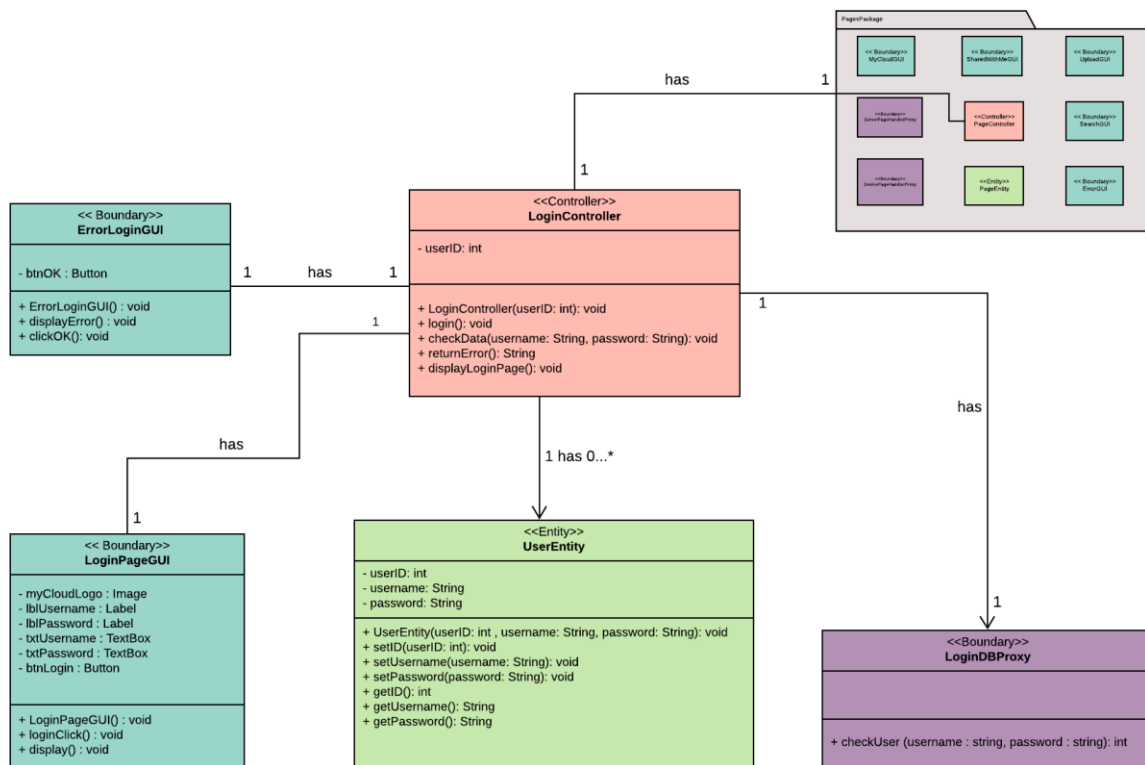
- **userID:** Ο μοναδικός χαρακτηριστικός αριθμός για κάθε χρήστη.

Μέθοδοι της κλάσης:

- **LoginController(userID: int):** Η μέθοδος δόμησης της κλάσης LoginController.
- **login():** Η μέθοδος αυτή συνδέει το χρήστη στο σύστημα, καλώντας τη μέθοδο *PageController::displayMyCloud(currentPage: String): void*, με όρισμα το String LoginPage.
- **checkData(username: String, password: String):** Η μέθοδος αυτή ελέγχει εάν τα στοιχεία που έδωσε ο χρήστης είναι ορθά. Σε περίπτωση ορθότητας καλείται η μέθοδος *LoginController::login():void*, διαφορετικά η *LoginController::returnError():String*.
- **returnError():** Η μέθοδος αυτή καλεί τη μέθοδο *errorLoginGUI::displayError(): void* λόγω αδυναμίας πιστοποίησης ονόματος χρήστη-κωδικού πρόσβασης.
- **displayLoginPage():** Η μέθοδος αυτή καλεί τη μέθοδο *LoginPageGUI::display(): void*.



2.1.1.2 Διάγραμμα κλάσεων LoginPackage



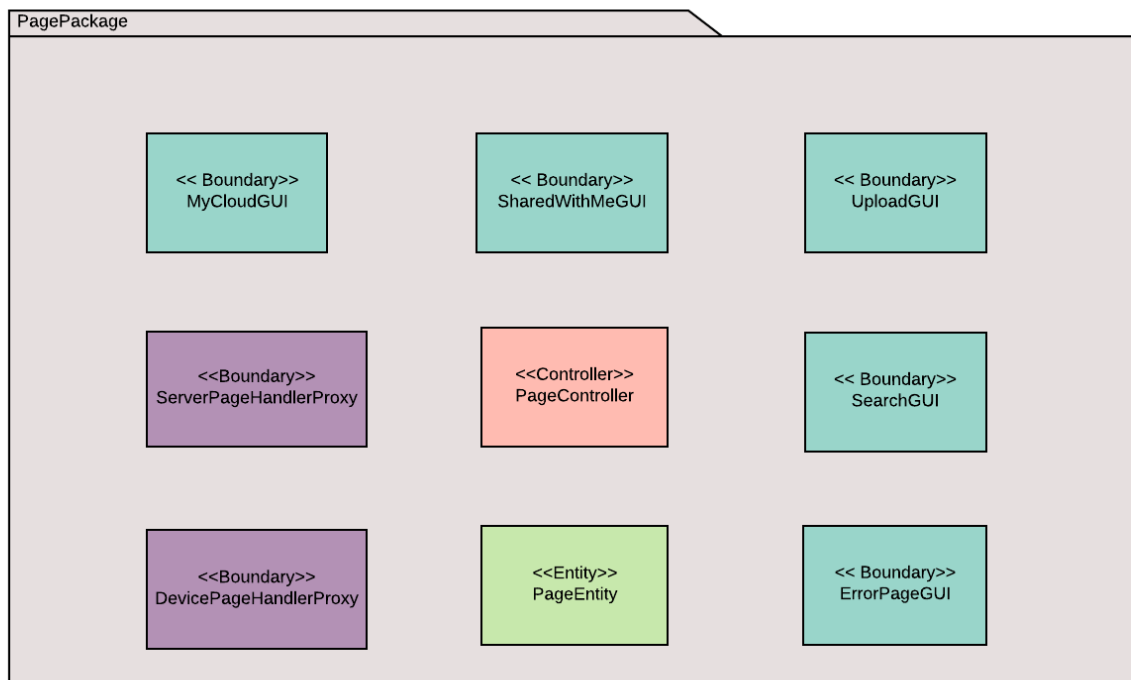
Σχήμα 1. Διάγραμμα κλάσεων πακέτου LoginPackage.



2.1.2 PagePackage

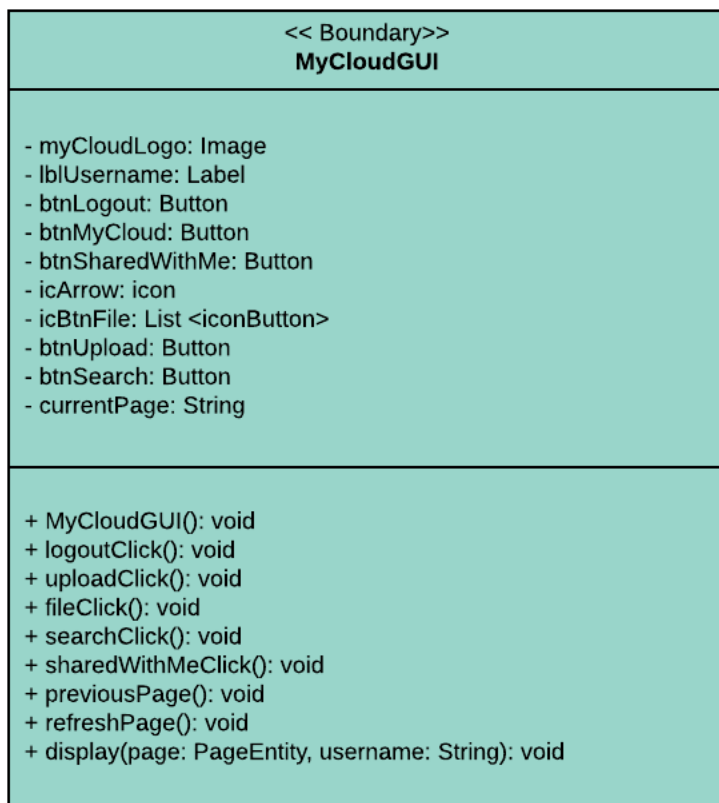
2.1.2.1 Περιγραφή Πακέτου PagePackage

Το πακέτο αυτό αναφέρεται στις γενικές ενέργειες του συστήματος στις οποίες μπορεί να προβεί ο χρήστης. Ουσιαστικά, περιλαμβάνει τις σελίδες “Το Cloud μου” και “Σε κοινή χρήση με εμένα” που προβάλλονται στο χρήστη καθώς και τις λειτουργίες Upload και Search για ένα αρχείο. Περαιτέρω, περιέχει δευτερεύουσας σημασίας σελίδες για τις λειτουργίες του Upload και του Search. Παρακάτω παρουσιάζονται οι κλάσεις που διαχειρίζονται τις παραπάνω επιλογές.





MyCloudGUI



Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας “To Cloud μου”.

Χαρακτηριστικά της κλάσης:

- **myCloudLogo** : Λογότυπο της εφαρμογής
- **lblUsername**: Ετικέτα με το όνομα του χρήστη κάτω αριστερά από το λογότυπο της εφαρμογής.
- **btnLogout**: Κουμπί για την αποσύνδεση από το σύστημα.
- **btnMyCloud**: Κουμπί για τη μετάβαση στη σελίδα “To Cloud μου”.
- **btnSharedWithMe**: Κουμπί για τη μετάβαση στη σελίδα “Σε κοινή χρήση με εμένα”.
- **icArrow**: Ένα εικονίδιο βελάκι που δείχνει σε ποια σελίδα βρίσκεται ο χρήστης.
- **icBtnFile**: Μία λίστα από εικονίδια τύπου button που απεικονίζουν τα αρχεία του χρήστη στην σελίδα “To Cloud μου”.
- **btnUpload**: Κουμπί για τη μεταφόρτωση ενός αρχείου.
- **btnSearch**: Κουμπί για την αναζήτηση ενός αρχείου.
- **currentPage**: Το String που δηλώνει σε ποια σελίδα βρίσκεται ο χρήστης.

Μέθοδοι της κλάσης:

- **MyCloudGUI()**: Η μέθοδος δόμησης της κλάσης MyCloudGUI.
- **logoutClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Αποσύνδεση” και αν έχει πατηθεί ενεργοποιεί τη μέθοδο *PageController::logout():void*.
- **uploadClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Μεταφόρτωση” και αν έχει πατηθεί ενεργοποιεί την μέθοδο *PageController::displayUpload(currentPage: String):void*.



- **fileClick():** Η μέθοδος αυτή ελέγχει αν πατήθηκε παρατεταμένα κάποιο αρχείο και αν έχει πατηθεί ενεργοποιεί την μέθοδο *PageController::openMenuForFile(selectedFile: FileEntity):void*.
- **searchClick():** Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Αναζήτηση” και αν έχει πατηθεί ενεργοποιεί την μέθοδο *PageController::displaySearch(currentPage: String, searchInput: String):void*.
- **sharedWithMeClick():** Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Σε κοινή χρήση με εμένα” και αν έχει πατηθεί ενεργοποιεί την μέθοδο *PageController::displaySharedWithMe(currentPage: String):void*.
- **previousPage():** Η μέθοδος αυτή ενεργοποιεί τη μέθοδο *PageController::previousPage():void*.
- **refreshPage():** Η μέθοδος αυτή είναι υπεύθυνη για την ανανέωση της σελίδας σύνδεσης.
- **display(page: PageEntity, username: String):** Η μέθοδος αυτή προβάλλει τη διεπαφή “To Cloud μου” με τη λίστα αρχείων του χρήστη, σύμφωνα με τα ορίσματα που δέχεται.

UploadGUI

<< Boundary>> UploadGUI
<ul style="list-style-type: none"> - myCloudLogo: Image - lblUpload: Label - lblSelectFile: Label - btnDirectory: List <Button> - icFile: List <icon> - lblFileName: List <Label> - cbxFile: List <CheckBox> - btnOpen: Button - btnCancel: Button - currentDevicePage: String
<ul style="list-style-type: none"> + UploadGUI(): void + openClick(): void + cancelClick(): void + checkBoxClick(): void + directoryClick(): void + previousPage(): void + refreshPage(): void + display(page: PageEntity): void

Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας όταν πραγματοποιείται η μεταφόρτωση ενός αρχείου.

Χαρακτηριστικά της κλάσης:

- **myCloudLogo** : Λογότυπο της εφαρμογής
- **lblUpload**: Ετικέτα με το όνομα “Μεταφόρτωση” κάτω από το λογότυπο της εφαρμογής.
- **lblSelectFile**: Ετικέτα με το όνομα “Επιλογή Αρχείου:” κάτω αριστερά από το λογότυπο “Μεταφόρτωση”.
- **btnDirectory**: Μία λίστα από κουμπιά για τη μετάβαση στους φακέλους της συσκευής του χρήστη.
- **icFile**: Μία λίστα από εικονίδια των αρχείων της σελίδας της συσκευής, στην οποία βρίσκεται ο χρήστης.



- **lblFileName**: Μία λίστα με Labels
- **cbxFile**: Μία λίστα από checkbox, όπου η επιλογή ενός εξ αυτών σηματοδοτεί την επιλογή ενός αρχείου.
- **btnOpen**: Κουμπί για την ενεργοποίηση της μεταφόρτωσης ενός επιλεγμένου αρχείου.
- **btnCancel**: Κουμπί για την ακύρωση της διαδικασίας της μεταφόρτωση.
- **currentDevicePage**: Το String που δηλώνει σε ποια σελίδα της συσκευής βρίσκεται ο χρήστης.

Μέθοδοι της κλάσης:

- **UploadGUI()**: Η μέθοδος δόμησης της κλάσης UploadGUI.
- **openClick()**: Η μέθοδος αυτή ενεργοποιεί τη μέθοδο *PageController::uploadFile(uploadFile: FileEntity:void)*.
- **cancelClick()**: Η μέθοδος αυτή ενεργοποιεί τη μέθοδο *PageController::previousPage():void*.
- **checkBoxClick()**: Η μέθοδος αυτή χρησιμοποιείται για την επιλογή του αρχείου της συσκευής.
- **directoryClick()**: Η μέθοδος ενεργοποιεί τη μέθοδο *PageController::browseFilesFromDevice(currentDevicePage: String):PageEntity*.
- **previousPage()**: Η μέθοδος αυτή ενεργοποιεί τη μέθοδο *PageController::previousPage():void*.
- **refreshPage()**: Η μέθοδος αυτή είναι υπεύθυνη για την ανανέωση της σελίδας σύνδεσης.
- **display(page: PageEntity)**: Η μέθοδος αυτή προβάλλει τη διεπαφή Upload σε συνδυασμό με τη διαθέσιμη λίστα αρχείων του χρήστη προς μεταφόρτωση, σύμφωνα με το PageEntity που δέχεται.

SearchGUI

<< Boundary>> SearchGUI
<ul style="list-style-type: none"> - myCloudLogo: Image - lblUsername: Label - btnLogout: Button - btnMyCloud: Button - btnSharedWithMe: Button - icArrow: icon - txtSearch: SearchView - btnSearch: Button - icBtnFile: List <iconButton> - searchInput: String - currentPage: String
<ul style="list-style-type: none"> + SearchGUI(): void + logoutClick(): void + searchClick(): void + myCloudClick(): void + sharedWithMeClick(): void + fileClick(): void + previousPage(): void + refreshPage(): void + display(page: PageEntity, username: String): void

Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας όταν πραγματοποιείται η αναζήτηση ενός αρχείου.

**Χαρακτηριστικά της κλάσης:**

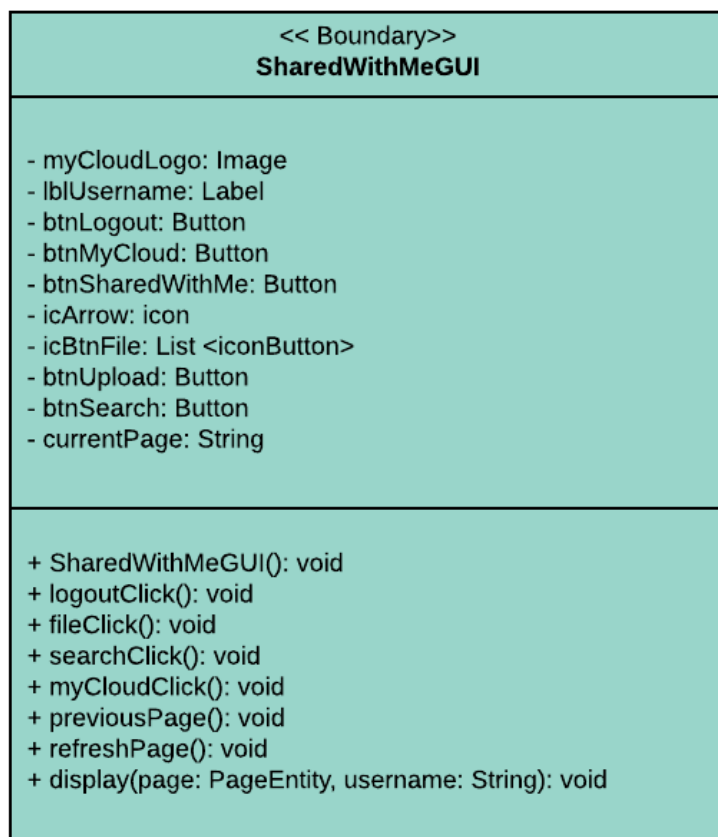
- **myCloudLogo** : Λογότυπο της εφαρμογής
- **lblUsername**: Ετικέτα με το όνομα του χρήστη κάτω αριστερά από το λογότυπο της εφαρμογής.
- **btnLogout**: Κουμπί για την αποσύνδεση από το σύστημα.
- **btnMyCloud**: Κουμπί για τη μετάβαση στη σελίδα “Το Cloud μου”.
- **btnSharedWithMe**: Κουμπί για τη μετάβαση στη σελίδα “Σε κοινή χρήση με εμένα”.
- **icArrow**: Ένα εικονίδιο βελάκι που δείχνει σε ποια σελίδα βρίσκεται ο χρήστης.
- **txtSearch**: Πεδίο συμπλήρωσης για την αναζήτηση αρχείου που ανανεώνει αυτόματα τη μεταβλητή `searchInput` με την προσθήκη ή αφαίρεση κάθε χαρακτήρα από το χρήστη.
- **btnSearch**: Κουμπί για την αναζήτηση αρχείου.
- **icBtnFile**: Μία λίστα από εικονίδια τύπου `button` που απεικονίζουν τα αρχεία του χρήστη που πληρούν τα εισαχθέντα στοιχεία της αναζήτησης.
- **searchInput**: Το `String` όπου αποθηκεύεται το όνομα του αρχείου προς αναζήτηση που εισήγαγε ο χρήστης στο `SearchView txtSearch`.
- **currentPage**: Το `String` που δηλώνει σε ποια σελίδα βρίσκεται ο χρήστης.

Μέθοδοι της κλάσης:

- **SearchGUI()**: Η μέθοδος δόμησης της κλάσης `SearchGUI`.
- **logoutClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Αποσύνδεση” και αν έχει πατηθεί ενεργοποιεί την μέθοδο `PageController::logout():void`.
- **searchClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Αναζήτηση” και αν έχει πατηθεί ενεργοποιεί την μέθοδο `PageController::displaySearch(currentPage: String, searchInput: String):void`.
- **myCloudClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Το Cloud μου” και αν έχει πατηθεί ενεργοποιεί την μέθοδο `PageController::displayMyCloud(currentPage: String):void`.
- **sharedWithMeClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Σε κοινή χρήση με εμένα” και αν έχει πατηθεί ενεργοποιεί την μέθοδο `PageController::displaySharedWithMe(currentPage : String):void`.
- **fileClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε παρατεταμένα κάποιο αρχείο και αν έχει πατηθεί ενεργοποιεί την μέθοδο `PageController::openMenuForFile(selectedFile: FileEntity):void`.
- **previousPage()**: Η μέθοδος αυτή ενεργοποιεί τη μέθοδο `PageController::previousPage():void`.
- **refreshPage()**: Η μέθοδος αυτή είναι υπεύθυνη για την ανανέωση της σελίδας σύνδεσης.
- **display(page: PageEntity, username: String)**: Η μέθοδος αυτή προβάλλει τη διεπαφή `Search` με τη ζητούμενη λίστα αρχείων.



SharedWithMeGUI



Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση της σελίδας “Σε κοινή χρήση με εμένα”.

Χαρακτηριστικά της κλάσης:

- **myCloudLogo** : Λογότυπο της εφαρμογής
- **lblUsername**: Ετικέτα με το όνομα του χρήστη κάτω αριστερά από το λογότυπο της εφαρμογής.
- **btnLogout**: Κουμπί για την αποσύνδεση από το σύστημα.
- **btnMyCloud**: Κουμπί για τη μετάβαση στη σελίδα “Το Cloud μου”.
- **btnSharedWithMe**: Κουμπί για τη μετάβαση στη σελίδα “Σε κοινή χρήση με εμένα”.
- **icArrow**: Ένα εικονίδιο βελάκι που δείχνει σε ποια σελίδα βρίσκεται ο χρήστης.
- **icBtnFile**: Μία λίστα από εικονίδια τύπου button που απεικονίζουν τα αρχεία του χρήστη στην σελίδα “Σε κοινή χρήση με εμένα”.
- **btnUpload**: Κουμπί για τη μεταφόρτωση ενός αρχείου.
- **btnSearch**: Κουμπί για την αναζήτηση ενός αρχείου.
- **currentPage**: Το String που δηλώνει σε ποια σελίδα βρίσκεται ο χρήστης.

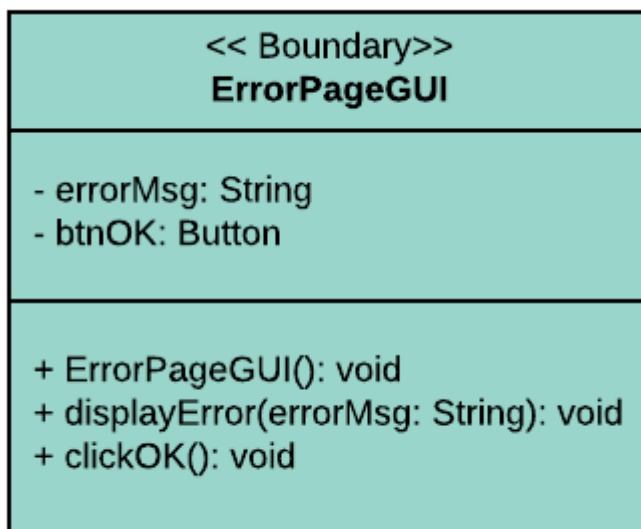
Μέθοδοι της κλάσης:

- **SharedWithMeGUI()**: Η μέθοδος δόμησης της κλάσης SharedWithMeGUI.
- **logoutClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Αποσύνδεση” και αν έχει πατηθεί ενεργοποιεί την μέθοδο logout() του ελεγκτή PageController.
- **fileClick()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε παρατεταμένα κάποιο αρχείο και αν έχει πατηθεί ενεργοποιεί την μέθοδο PageController::openMenuForFile(selectedFile : FileEntity):void.



- **searchClick():** Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Αναζήτηση” και αν έχει πατηθεί ενεργοποιεί την μέθοδο *PageController::displaySearch(currentPage : String, searchInput: String):void*.
- **myCloudClick():** Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “Το Cloud μου” και αν έχει πατηθεί ενεργοποιεί την μέθοδο *PageController::displayMyCloud(currentPage : String):void*.
- **previousPage():** Η μέθοδος αυτή ενεργοποιεί τη μέθοδο *PageController::previousPage():void*.
- **refreshPage():** Η μέθοδος αυτή είναι υπεύθυνη για την ανανέωση της σελίδας σύνδεσης.
- **display(page: PageEntity, username: String):** Η μέθοδος αυτή προβάλλει τη διεπαφή “Σε κοινή χρήση με εμένα” με τη λίστα αρχείων του χρήστη, σύμφωνα με τα ορίσματα που δέχεται.

ErrorPageGUI



Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση αναδυόμενου παραθύρου σε περίπτωση σφάλματος κατά την ενέργεια Μεταφόρτωσης ή της Αναζήτησης.

Χαρακτηριστικά της κλάσης:

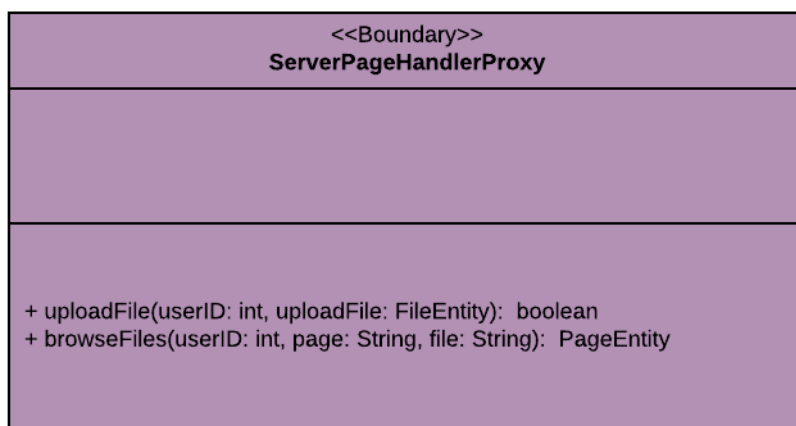
- **errorMsg** : Το String που περιέχει το μήνυμα σφάλματος που προκλήθηκε.
- **btnOK** : Κουμπί για την επιβεβαίωση ότι ο χρήστης ενημερώθηκε για το σφάλμα.

Μέθοδοι της κλάσης:

- **ErrorPageGUI():** Η μέθοδος δόμησης της κλάσης ErrorPageGUI.
- **displayError(errorMsg: String):** Η μέθοδος αυτή είναι υπεύθυνη για την προβολή του μηνύματος σφάλματος “Ανεπαρκής χώρος” ή “Δεν βρέθηκε αρχείο με αυτό το όνομα” στο χρήστη, ανάλογα με την τιμή της μεταβλητής errorMsg.
- **clickOK():** Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί “OK” και αν έχει πατηθεί ενεργοποιεί τη μέθοδο *PageController::previousPage():void*.



ServerPageHandlerProxy



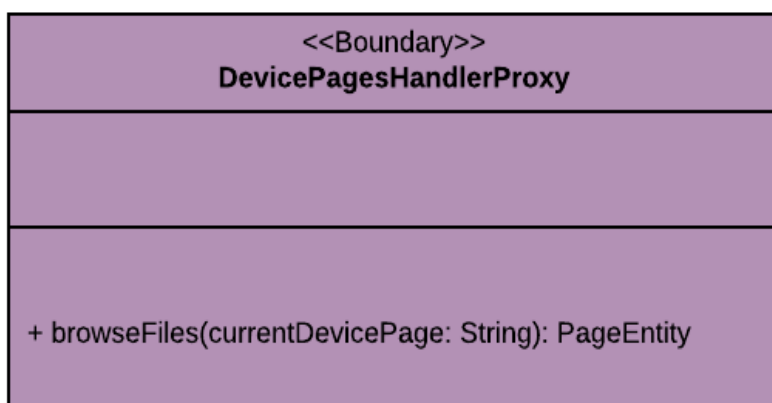
Περιγραφή:

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με τον εξυπηρετητή του συστήματος όπου είναι αποθηκευμένα τα αρχεία του χρήστη.

Μέθοδοι της κλάσης:

- **uploadFile(userID: int, uploadFile: FileEntity):** Η μέθοδος αυτή μεταφορτώνει το αρχείο που δόθηκε ως όρισμα στον αποθηκευτικό χώρο του χρήστη με το αντίστοιχο userID. Επιστρέφει True ή False, αν η μεταφόρτωση ήταν επιτυχής ή ανεπιτυχής αντίστοιχα.
- **browseFiles(userID: int, page: String, file: String):** Η μέθοδος αυτή αναζητά σύμφωνα με το userID, τα μεταφορτωμένα αρχεία του χρήστη στη σελίδα που καθορίζεται από το String page. Αν το String file δεν είναι κενό, γίνεται και αναζήτηση με βάση το όνομα. Η μέθοδος επιστρέφει ένα PageEntity.

DevicePageHandlerProxy



Περιγραφή:

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με τη συσκευή του χρήστη.

Μέθοδοι της κλάσης:

- **browseFiles(currentDevicePage: String):** Η μέθοδος αυτή αναζητά τα αρχεία του χρήστη στη σελίδα που καθορίζεται από το String currentDevicePage και επιστρέφει ένα PageEntity.



PageEntity

<<Entity>> PageEntity	
- pageName: String - pageTotalSpace: double - lstFiles: List <FileEntity>	
+ PageEntity(userID: int , name: String, pageTotalSpace: double, lstFiles: List <FileEntity>): void + setPageName(pageName: String): void + setPageTotalSpace(pageTotalSpace: double): void + setLstFiles(lstFiles: List <FileEntity>): void + getLstFiles(): void + getPageName(): String + getPageTotalSpace(): double	

Περιγραφή:

Αυτή η κλάση υλοποιεί την οντότητα της σελίδας καθορίζοντας τα βασικά χαρακτηριστικά της, καθώς και τις μεθόδους που απαιτούνται.

Χαρακτηριστικά της κλάσης:

- **pageName:** Το όνομα της κάθε σελίδας.
- **pageTotalSpace:** Ο συνολικός χώρος που δεσμεύουν τα αρχεία της σελίδας στον εξυπηρετητή (server).
- **lstFiles:** Μία λίστα τύπου FileEntity, τα οποία είναι το σύνολο των αρχείων της σελίδας.

Μέθοδοι της κλάσης:

- **PageEntity(userID: int , name: String, pageTotalSpace: double, lstFiles: List<FileEntity>):** Η μέθοδος δόμησης της κλάσης PageEntity.
- **setPageName(pageName: String):** Η μέθοδος αυτή θέτει το όνομα της σελίδας σύμφωνα με το όρισμα που δέχεται.
- **setPageTotalSpace(pageTotalSpace: double):** Η μέθοδος αυτή θέτει το συνολικό αποθηκευτικό χώρο που δεσμεύουν τα αρχεία της σελίδας, σύμφωνα με το όρισμα που δέχεται.
- **setLstFiles(lstFiles: List<FileEntity>):** Η μέθοδος αυτή θέτει τη λίστα των FileEntity που περιέχει η σελίδα.
- **getLstFiles():** Η μέθοδος αυτή επιστρέφει τη λίστα των FileEntity που περιέχει η σελίδα.
- **getPageName():** Η μέθοδος αυτή επιστρέφει το όνομα της σελίδας.
- **getPageTotalSpace():** Η μέθοδος αυτή επιστρέφει το συνολικό χώρο που δεσμεύουν τα αρχεία της σελίδας.



PageController

<<Controller>> PageController
- lstErrorMessages : List <String> - username : String - userID : int
+ PageController (userID: int, username: String): void + displayMyCloud (currentPage: String): void + displaySharedWithMe (currentPage: String): void + displaySearch (currentPage: String, searchInput: String): void + displayUpload (currentPage : String): void + previousPage (): void + logout (): void + browseFiles (currentPage: String, file: String): PageEntity + browseFilesFromDevice (currentDevicePage: String): PageEntity + uploadFile (uploadFile: FileEntity): void + openMenuForFile (selectedFile : FileEntity): void

Περιγραφή:

Η κλάση αυτή αποτελεί έναν ελεγκτή, ο οποίος πραγματοποιεί όλες τις ενέργειες που απαιτούν οι σελίδες “Το Cloud μου” και “Σε κοινή χρήση με εμένα”, καθώς και οι διαθέσιμες ενέργειες του χρήστη για Upload και Search.

Χαρακτηριστικά της κλάσης:

- **lstErrorMessages**: Μία λίστα με τα διάφορα μηνύματα σφάλματος που εμφανίζει το σύστημα.
- **username**: Ένα String με το όνομα του χρήστη.
- **userID**: Ο μοναδικός χαρακτηριστικός αριθμός για κάθε χρήστη.

Μέθοδοι της κλάσης:

- **PageController(userID: int, username: String)**: Η μέθοδος δόμησης της κλάσης PageController.
- **displayMyCloud(currentPage: String)**: Η μέθοδος αυτή καλεί τη μέθοδο *PageController::browseFiles(currentPage: String, file: String): PageEntity*, με κενό String file. Μετά εμφανίζει τη σελίδα “Το Cloud μου”, ενεργοποιώντας τη μέθοδο *MyCloudGUI::display(page: PageEntity, username: String):void*, με όρισμα το PageEntity που επιστράφηκε προηγουμένως.
- **displaySharedWithMe(currentPage: String)**: Η μέθοδος αυτή καλεί τη μέθοδο *PageController::browseFiles(currentPage: String, file: String): PageEntity*, με κενό String file. Μετά εμφανίζει τη σελίδα “Σε κοινή χρήση με εμένα”, ενεργοποιώντας τη μέθοδο *SharedWithMeGUI::display(page: PageEntity, username: String):void*, με όρισμα το PageEntity που επιστράφηκε προηγουμένως.
- **displaySearch(currentPage: String, searchInput: String)**: Η μέθοδος αυτή καλεί τη μέθοδο *PageController::browseFiles(currentPage: String, file: String): PageEntity*, δίνοντας ως όρισμα την τιμή της προερχόμενης σελίδας και την τιμή της μεταβλητής searchInput. Αν η μέθοδος *PageController::browseFiles(currentPage: String, file: String): PageEntity* επιστρέψει NULL, τότε

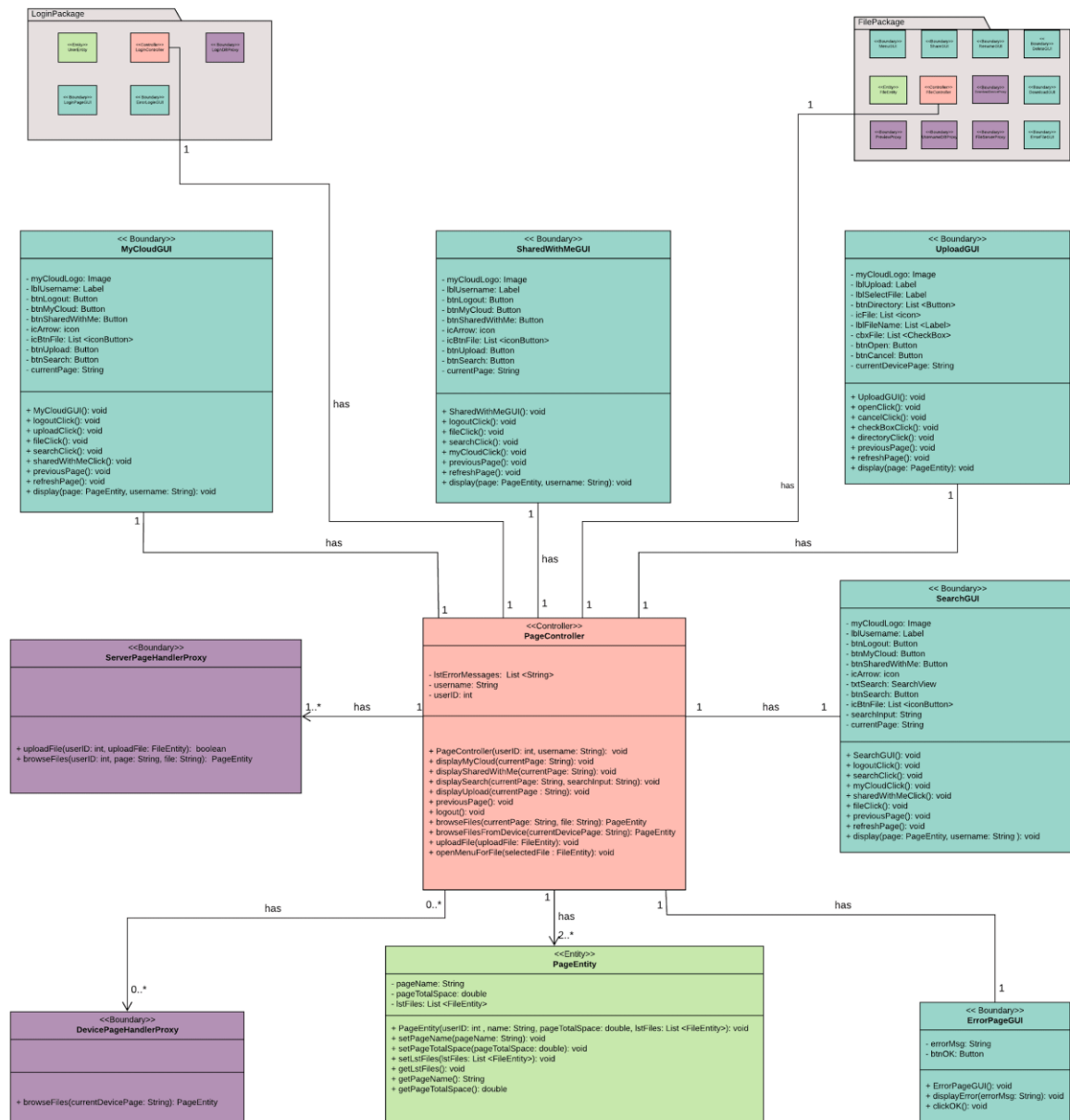


ενεργοποιεί την *ErrorPageGUI::displayError(errorMsg: String): void*, με τιμή errorMsg = "search". Διαφορετικά εμφανίζει τη διεπαφή αναζήτησης, ενεργοποιώντας τη μέθοδο *SearchGUI::display(page: PageEntity, username: String): void*, με όρισμα το PageEntity που επιστράφηκε προηγουμένως.

- **displayUpload(currentPage : String):** Η μέθοδος αυτή καλεί τη μέθοδο *PageController::browseFilesFromDevice(currentPage: String): PageEntity* και εμφανίζει τη διεπαφή για μεταφόρτωση, ενεργοποιώντας τη μέθοδο *uploadGUI::display(page: PageEntity, username: String):void*, δίνοντας ως όρισμα το PageEntity που επιστράφηκε προηγουμένως.
- **previousPage():** Η μέθοδος αυτή είναι υπεύθυνη για την επιστροφή στην προηγούμενη σελίδα.
- **logout():** Η μέθοδος που αποσυνδέει τον χρήστη από το σύστημα, καλώντας την μέθοδο *LoginController::displayLoginPage(): void*.
- **browseFiles(currentPage: String, file: String):** Η μέθοδος αυτή ενεργοποιεί την μέθοδο *ServerPagesHandlerProxy::browseFiles(userID: int, page: String, file: String): PageEntity* για την εύρεση των αρχείων που έχει ο χρήστης μεταφορτωμένα στο νέφος, στη σελίδα που δίνεται από το String page και επιστρέφει μία λίστα τύπου FileEntity. Το String file, όταν δεν είναι το κενό String, χρησιμοποιείται για αναζήτηση αρχείων με βάση το όνομα.
- **browseFilesFromDevice(currentDevicePage: String):** Η μέθοδος αυτή ενεργοποιεί τη μέθοδο *DevicePageHandlerProxy:: browseFiles(currentDevicePage: String): PageEntity* για την εύρεση των αρχείων που έχει ο χρήστης στην συσκευή του, στη σελίδα που βρίσκεται και επιστρέφει ένα PageEntity. Η σελίδα που βρίσκεται δίνεται ως όρισμα με το String currentDevicePage.
- **uploadFile(uploadFile: FileEntity):** Η μέθοδος αυτή ενεργοποιεί τη μέθοδο *ServerProxy::uploadFile(userID: int, uploadFile: FileEntity): boolean* για την μεταφόρτωση του επιλεγμένου αρχείου. Σε περίπτωση που επιστραφεί τιμή False, ενεργοποιείται η *ErrorPageGUI::displayError(errorMsg: String): void*, με τιμή errorMsg = "upload".
- **openMenuForFile(selectedFile : FileEntity):** Η μέθοδος αυτή είναι υπεύθυνη για το άνοιγμα του MenuGUI για το αρχείο που επιλέχθηκε, ενεργοποιώντας τη μέθοδο *FileController::displayMenu(file: FileEntity): void*.



2.1.2.2 Διάγραμμα κλάσεων PagePackage



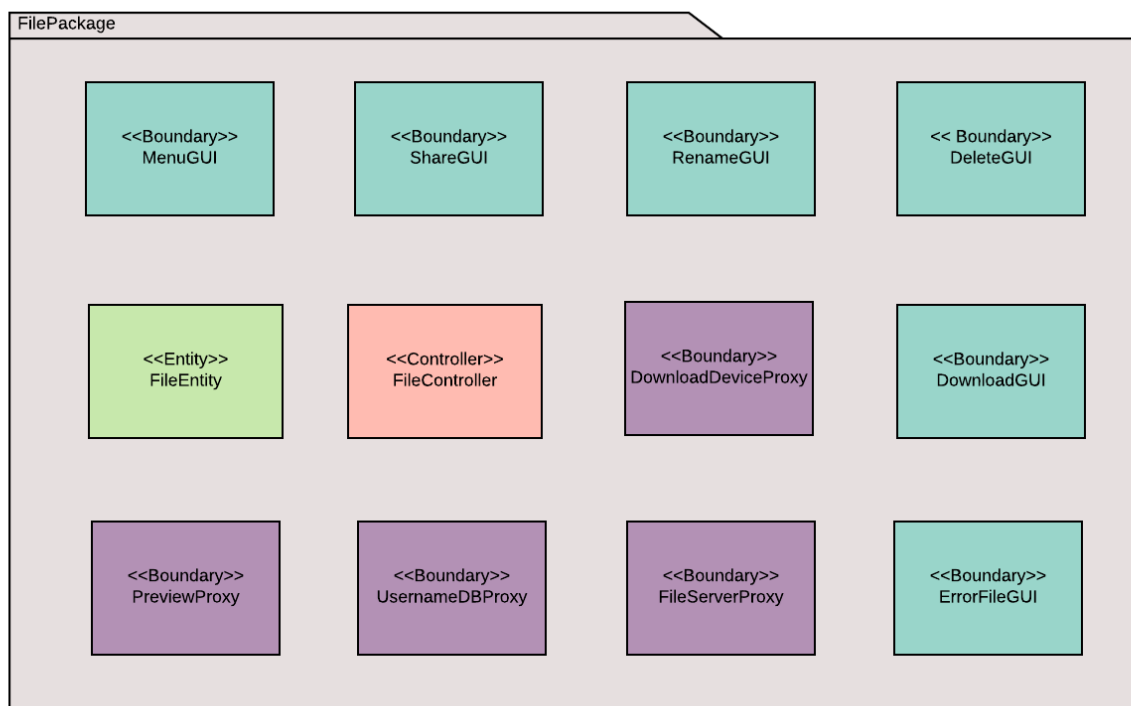
Σχήμα 2. Διάγραμμα κλάσεων πακέτου PagePackage.



2.1.3 FilePackage

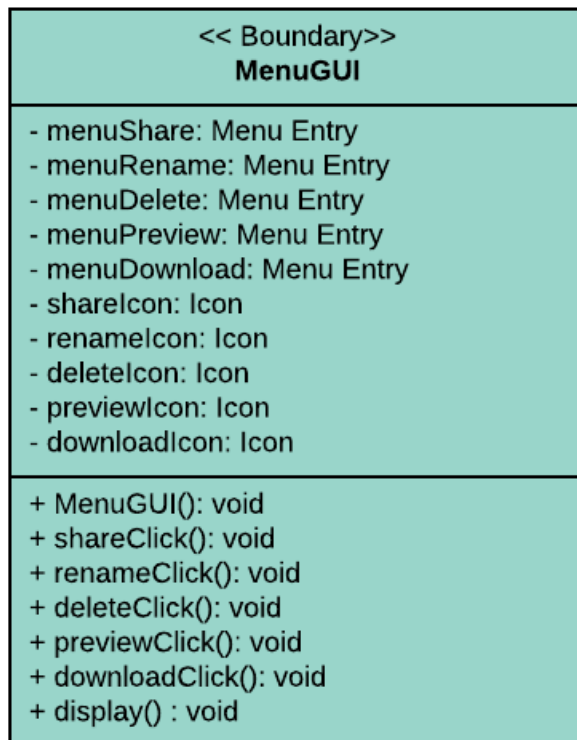
2.1.3.1 Περιγραφή Πακέτου FilePackage

Το πακέτο αυτό αναφέρεται στις ενέργειες του συστήματος στις οποίες μπορεί να προβεί ο χρήστης για διαχειριστεί ένα αρχείο. Οι ενέργειες αυτές είναι οι εξής: διαμοιρασμός, διαγραφή, μετονομασία, λήψη και προεπισκόπηση ενός αρχείου.. Παρακάτω παρουσιάζονται οι κλάσεις που διαχειρίζονται τις παραπάνω επιλογές.





MenuGUI



Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση του αναδυόμενου μενού ενεργειών.

Χαρακτηριστικά της κλάσης:

- **menuShare:** Επιλογή Μενού για το διαμοιρασμό του επιλεγμένου αρχείου με άλλον χρήστη του συστήματος.
- **menuRename:** Επιλογή Μενού για τη μετονομασία του επιλεγμένου αρχείου.
- **menuDelete:** Επιλογή Μενού για τη διαγραφή του επιλεγμένου αρχείου.
- **menuPreview:** Επιλογή Μενού για την προεπισκόπηση του επιλεγμένου αρχείου.
- **menuDownload:** Επιλογή Μενού για τη λήψη του επιλεγμένου αρχείου.
- **shareIcon:** Ένα εικονίδιο ανθρωπάκι που συμβολίζει το διαμοιρασμό με άλλο χρήστη.
- **renameIcon:** Ένα εικονίδιο μολύβι που συμβολίζει τη μετονομασία ενός αρχείου.
- **deleteIcon:** Ένα εικονίδιο με έναν κάδο που συμβολίζει τη διαγραφή ενός αρχείου.
- **previewIcon:** Ένα εικονίδιο με ένα μάτι που συμβολίζει την προεπισκόπηση ενός αρχείου.
- **downloadIcon:** Ένα εικονίδιο βέλος με κατεύθυνση προς τα κάτω που συμβολίζει τη λήψη ενός αρχείου.

Μέθοδοι της κλάσης:

- **MenuGUI():** Η μέθοδος δόμησης της κλάσης MenuGUI.
- **shareClick():** Η μέθοδος αυτή ελέγχει αν επιλέχθηκε η ενέργεια Διαμοιρασμός από το χρήστη και καλεί την μέθοδο *FileController::shareFile(receiverName: String):void*.
- **renameClick():** Η μέθοδος αυτή ελέγχει αν επιλέχθηκε η ενέργεια Μετονομασία από το χρήστη και καλεί τη μέθοδο *FileController::renameFile(newName: String):void*.



- **deleteClick():** Η μέθοδος αυτή ελέγχει αν επιλέχθηκε η ενέργεια Διαγραφή από το χρήστη και καλεί τη μέθοδο *FileController::deleteFile():void*.
- **previewClick():** Η μέθοδος αυτή ελέγχει αν επιλέχθηκε η ενέργεια Προεπισκόπηση από το χρήστη και καλεί τη μέθοδο *FileController::previewFile():void*.
- **downloadClick():** Η μέθοδος αυτή ελέγχει αν επιλέχθηκε η ενέργεια Διαμοιρασμός από το χρήστη και καλεί τη μέθοδο *FileController::downloadFile():void*.
- **display():** Η μέθοδος αυτή προβάλλει στην οθόνη το γραφικό περιβάλλον του μενού.

ShareGUI

<< Boundary >> ShareGUI
- shareMsg: String - receiverName: String - txtReceiverName: TextBox - btnOK: Button - btnCancel: Button
+ ShareGUI(): void + clickOK(): void + clickCancel(): void + display(): void

Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση του αναδυόμενου παραθύρου εισαγωγής του χρήστη παραλήπτη για το επιλεγμένο αρχείο κατά την ενέργεια του Διαμοιρασμού.

Χαρακτηριστικά της κλάσης:

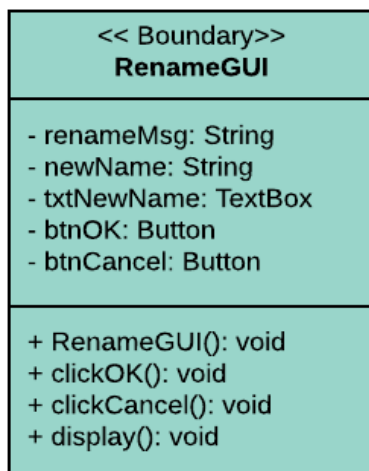
- **shareMsg:** Το μήνυμα που εμφανίζει το σύστημα σχετικά με την εισαγωγή ονόματος χρήστη παραλήπτη από τον χρήστη.
- **receiverName:** Το String όπου αποθηκεύεται το όνομα του χρήστη παραλήπτη που εισήγαγε ο χρήστης στο αντίστοιχο TextBox, για να διαμοιραστεί το επιθυμητό αρχείο.
- **txtReceiverName:** Πεδίο συμπλήρωσης για το όνομα του χρήστη παραλήπτη που θα λάβει το επιλεγμένο αρχείο.
- **btnOK :** Κουμπί για την εκκίνηση της διαδικασίας Διαμοιρασμού από το σύστημα.
- **btnCancel:** Κουμπί για την ακύρωση της ενέργειας Διαμοιρασμού.

Μέθοδοι της κλάσης:

- **ShareGUI():** Η μέθοδος δόμησης της κλάσης ShareGUI.
- **clickOK():** Η μέθοδος αυτή αν έχει πατηθεί το κουμπί "OK" ενεργοποιεί τη μέθοδο *FileController::shareFile(receiverName: String):void*.
- **clickCancel():** Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί "Άκυρο" και καλεί τη μέθοδο *FileController::previousPage():void*.
- **display():** Η μέθοδος αυτή προβάλλει στην οθόνη το γραφικό περιβάλλον του διαμοιρασμού.



RenameGUI



Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση του αναδυόμενου παραθύρου για το αίτημα οριστικής διαγραφής του επιλεγμένου αρχείου, από το χρήστη στο σύστημα.

Χαρακτηριστικά της κλάσης:

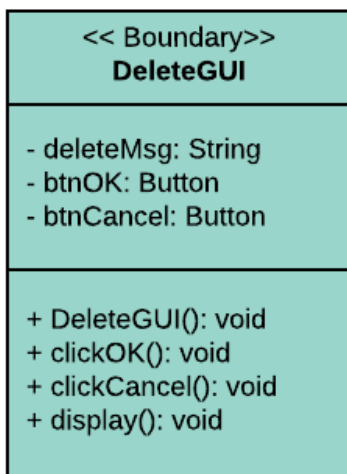
- **renameMsg**: Το μήνυμα που εμφανίζει το σύστημα για την επιβεβαίωση εισαγωγής νέου ονόματος του επιλεγμένου αρχείου.
- **newName**: Το String όπου αποθηκεύεται το νέο όνομα του αρχείου μετά την εισαγωγή του από τον χρήστη.
- **txtNewName**: Πεδίο συμπλήρωσης για το νέο όνομα του επιλεγμένου προς μετονομασία αρχείου.
- **btnOK** : Κουμπί για την εκκίνηση της διαδικασίας μετονομασίας από το σύστημα.
- **btnCancel**: Κουμπί για την ακύρωση της ενέργειας Μετονομασίας.

Μέθοδοι της κλάσης:

- **RenameGUI()**: Η μέθοδος δόμησης της κλάσης RenameGUI.
- **clickOK()**: Η μέθοδος αυτή αν έχει πατηθεί το κουμπί "OK" ενεργοποιεί τη μέθοδο *FileController::renameFile(newName: String):void*.
- **clickCancel()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί "Ακυρο" και καλεί τη μέθοδο *FileController::previousPage():void*.
- **display()**: Η μέθοδος αυτή προβάλλει στην οθόνη το γραφικό περιβάλλον της μετονομασίας.



DeleteGUI



Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση του αναδυόμενου παραθύρου για το αίτημα οριστικής διαγραφής του επιλεγμένου αρχείου, από το χρήστη στο σύστημα.

Χαρακτηριστικά της κλάσης:

- **deleteMsg**: Το μήνυμα που εμφανίζει το σύστημα για την επιβεβαίωση οριστικής διαγραφής του επιλεγμένου αρχείου.
- **btnOK** : Κουμπί για την οριστική διαγραφή από το σύστημα.
- **btnCancel**: Κουμπί για την ακύρωση της ενέργειας Διαγραφής.

Μέθοδοι της κλάσης:

- **DeleteGUI()**: Η μέθοδος δόμησης της κλάσης DeleteGUI.
- **clickOK()**: Η μέθοδος αυτή αν έχει πατηθεί το κουμπί "OK" ενεργοποιεί τη μέθοδο *FileController::deleteFile():void*.
- **clickCancel()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί "Ακυρο" και καλεί τη μέθοδο *FileController::previousPage():void*.
- **display()**: Η μέθοδος αυτή προβάλλει στην οθόνη το γραφικό περιβάλλον της διαγραφής.



DownloadGUI

<< Boundary>> DownloadGUI
- downloadMsg: String - btnOK: Button - btnCancel: Button
+ DownloadGUI(): void + clickOK(): void + clickCancel(): void + display(): void

Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση του αναδυόμενου παραθύρου για καθορισμό από το χρήστη της τοποθεσίας λήψης του επιλεγμένου αρχείου στη συσκευή.

Χαρακτηριστικά της κλάσης:

- **downloadMsg**: Το μήνυμα που εμφανίζει το σύστημα για την επιβεβαίωση λήψης του επιλεγμένου αρχείου.
- **btnOK**: Κουμπί για την εκκίνηση της διαδικασίας λήψης.
- **btnCancel**: Κουμπί για την ακύρωση της ενέργειας λήψης.

Μέθοδοι της κλάσης:

- **DownloadGUI()**: Η μέθοδος δόμησης της κλάσης DownloadGUI.
- **clickOK()**: Η μέθοδος αυτή αν έχει πατηθεί το κουμπί "OK" ενεργοποιεί τη μέθοδο *FileController::downloadFile():void*.
- **clickCancel()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί "Ακυρο" και καλεί τη μέθοδο *FileController::previousPage():void*.
- **display()**: Η μέθοδος αυτή προβάλλει στην οθόνη το γραφικό περιβάλλον της λήψης.



ErrorFileGUI

<< Boundary>> ErrorFileGUI
- errorMsg: String - btnOK: Button
+ ErrorFileGUI(): void + displayError(errorMsg: String): void + clickOK(): void

Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση αναδυόμενου παραθύρου σε περίπτωση σφάλματος σε οποιαδήποτε ενέργεια.

Χαρακτηριστικά της κλάσης:

- **errorMsg**: Το String που περιέχει το μήνυμα σφάλματος που προκλήθηκε.
- **btnOK** : Κουμπί για την επιβεβαίωση ότι ο χρήστης ενημερώθηκε για το σφάλμα.

Μέθοδοι της κλάσης:

- **ErrorFileGUI()**: Η μέθοδος δόμησης της κλάσης ErrorShareGUI.
- **displayError(errorMsg: String)**: Η μέθοδος αυτή προβάλλει το μήνυμα σφάλματος
- **clickOK()**: Η μέθοδος αυτή ελέγχει αν πατήθηκε το κουμπί "OK" και καλεί τη μέθοδο *FileController::previousPage(): void*.

FileServerProxy

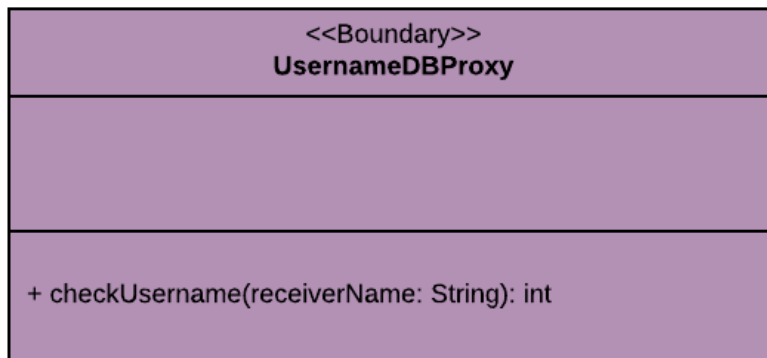
<<Boundary>> FileServerProxy
+ deleteFile(file: FileEntity): void + renameFile(file: FileEntity, newName: String): boolean + shareFile(file: FileEntity, receiverID: int): boolean

Περιγραφή:

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με τη βάση δεδομένων των αρχείων.

**Μέθοδοι της κλάσης:**

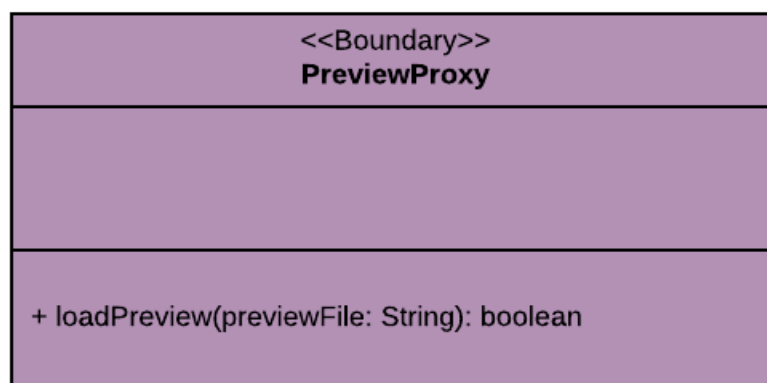
- **deleteFile(file: FileEntity):** Η μέθοδος αυτή διαγράφει μόνιμα από τη βάση δεδομένων το αρχείο που δέχεται ως όρισμα.
- **renameFile(file: FileEntity, newName: String):** Η μέθοδος αυτή μετονομάζει το επιλεγμένο αρχείο στο νέο όνομα που δέχεται ως όρισμα.
- **shareFile(file: FileEntity, receiverID: int):** Η μέθοδος αυτή δημιουργεί ένα αντίγραφο του επιλεγμένου αρχείου στο λογαριασμό του χρήστη με το αναγνωριστικό που δέχεται ως όρισμα και επιστρέφει 1, διαφορετικά αν αποτύχει επιστρέφει 0.

UsernameDBProxy**Περιγραφή:**

Η κλάση αυτή χρησιμοποιείται για επικοινωνία με τη βάση δεδομένων των χρηστών.

Μέθοδοι της κλάσης:

- **checkUsername(receiverName: String):** Η μέθοδος αυτή αναζητά στη βάση δεδομένων το όνομα χρήστη που δέχεται ως όρισμα και επιστρέφει το αναγνωριστικό του, userID. Σε περίπτωση μη έγκυρου ονόματος χρήστη, επιστρέφει την τιμή 0.

PreviewProxy**Περιγραφή:**

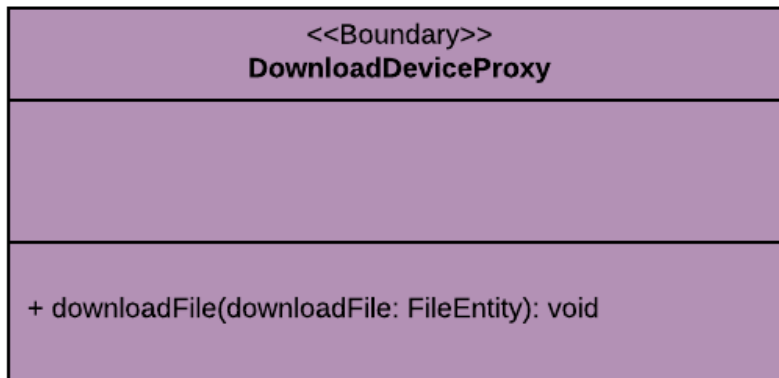
Η κλάση αυτή χρησιμοποιείται για επικοινωνία με το σύστημα απεικόνισης αρχείων.

Μέθοδοι της κλάσης:

- **loadPreview(previewFile: String):** Η μέθοδος αυτή παίρνει σαν όρισμα το όνομα του επιλεγμένου αρχείου και πραγματοποιεί την προεπισκόπηση. Σε περίπτωση μη έγκυρου τύπου αρχείου, επιστρέφει την τιμή 0.



DownloadDeviceProxy



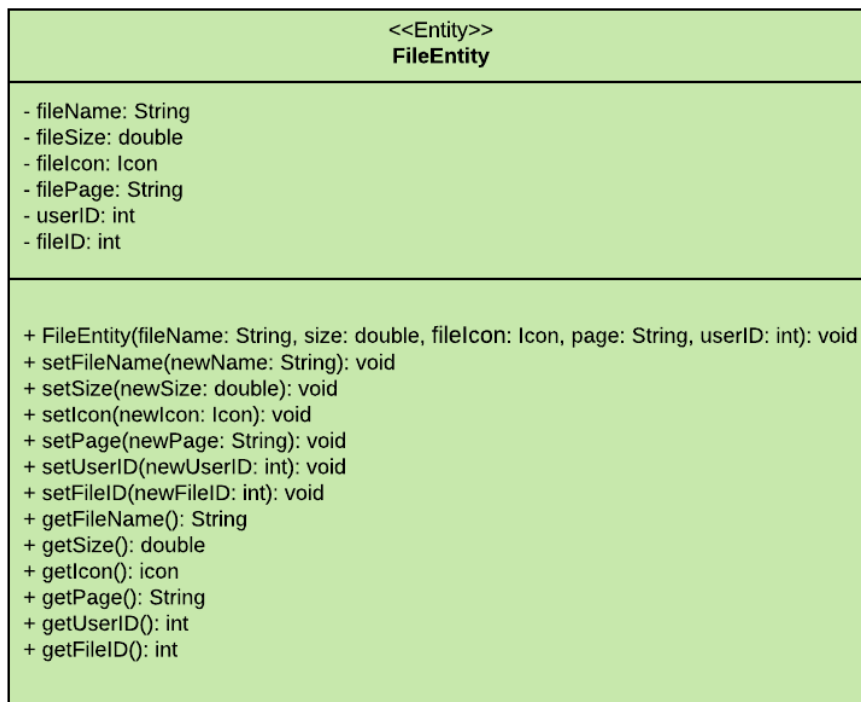
Περιγραφή:

Η κλάση αυτή αναπαριστά τη διεπαφή για την εμφάνιση αναδυόμενου παραθύρου σε περίπτωση σφάλματος κατά την ενέργεια Προεπισκόπησης.

Μέθοδοι της κλάσης:

- **downloadFile(downloadFile: FileEntity):** Η μέθοδος αυτή παίρνει σαν όρισμα το επιλεγμένο αρχείο, ώστε να εκκινήσει τη διαδικασία λήψης στη συσκευή.

FileEntity



Περιγραφή:

Η κλάση αυτή υλοποιεί την οντότητα του αρχείου προς διαχείριση καθορίζοντας τα βασικά χαρακτηριστικά του, καθώς και μεθόδους για τη χρησιμοποίησή αυτών από άλλες κλάσεις.

**Χαρακτηριστικά της κλάσης:**

- **fileName:** Το όνομα του αρχείου που θέλει να διαχειριστεί ο χρήστης .
- **fileSize:** Το μέγεθος του αρχείου σε Bytes που θέλει να διαχειριστεί ο χρήστης.
- **fileIcon:** Το εικονίδιο του αρχείου που θέλει να διαχειριστεί ο χρήστης.
- **filePage:** Η σελίδα, “To Cloud” μου ή “Σε κοινή χρήση με εμένα”, στην οποία βρίσκεται το αρχείο που θέλει να διαχειριστεί ο χρήστης.
- **userID:** Το αναγνωριστικό του χρήστη στην κατοχή του οποίου βρίσκεται το επιλεγμένο αρχείο.
- **fileID:** Το αναγνωριστικό κάθε αρχείου.

Μέθοδοι της κλάσης:

- **FileEntity(fileName: String, size: double, fileIcon: Icon, page: String, userID: int):** Η μέθοδος δόμησης της κλάσης FileEntity.
- **setFileName(newName: String):** Η μέθοδος αυτή θέτει το όνομα του αρχείου προς διαχείριση στο νέο όνομα που έχει ως όρισμα.
- **setSize(newSize: double):** Η μέθοδος αυτή θέτει το μέγεθος του αρχείου προς διαχείριση στο νέο μέγεθος που έχει ως όρισμα.
- **setIcon(newIcon: Icon):** Η μέθοδος αυτή θέτει το εικονίδιο του αρχείου προς διαχείριση στο νέο εικονίδιο που έχει ως όρισμα.
- **setPage(newPage: String):** Η μέθοδος αυτή θέτει το όνομα της σελίδας που βρίσκεται το αρχείο προς διαχείριση στο όνομα της νέας σελίδας που έχει ως όρισμα.
- **setUserID(newUserID: int):** Η μέθοδος αυτή θέτει τον αριθμό του αναγνωριστικού του χρήστη που είναι κάτοχος του επιλεγμένου αρχείου στο νέο αριθμό που έχει ως όρισμα.
- **setFileID(newFileID: int):** Η μέθοδος αυτή θέτει τον αριθμό του αναγνωριστικού αρχείου προς διαχείριση στο νέο αριθμό που έχει ως όρισμα.
- **getFileName():** Η μέθοδος αυτή επιστρέφει το όνομα του αρχείου προς διαχείριση.
- **getSize():** Η μέθοδος αυτή επιστρέφει το μέγεθος του αρχείου προς διαχείριση.
- **getIcon():** Η μέθοδος αυτή επιστρέφει το εικονίδιο του αρχείου προς διαχείριση.
- **getPage():** Η μέθοδος αυτή επιστρέφει το όνομα της σελίδας που βρίσκεται το αρχείο προς διαχείριση .
- **getUserID():** Η μέθοδος αυτή επιστρέφει τον αριθμό του αναγνωριστικού του χρήστη που είναι κάτοχος του επιλεγμένου αρχείου.
- **getFileID():** Η μέθοδος αυτή επιστρέφει τον αριθμό του αναγνωριστικού αρχείου προς διαχείριση.



FileController

<<Controller>> FileController
- file: FileEntity
+ FileController(file: FileEntity): void + displayMenu(file: FileEntity): void + displayShareFile(): String + displayRenameFile(): void + displayDeleteFile(): void + displayDownloadFile(): void + shareFile(receiverName: String): void + renameFile(newName: String): void + deleteFile(): void + previewFile(): void + downloadFile(): void + previousPage(): void

Περιγραφή: Η κλάση αυτή αποτελεί έναν ελεγκτή, ο οποίος πραγματοποιεί τη διαχείριση των αρχείων.

Χαρακτηριστικά της κλάσης:

- **file:** Ένα αντικείμενο με τα χαρακτηριστικά του επιλεγμένου αρχείου προς διαχείριση.

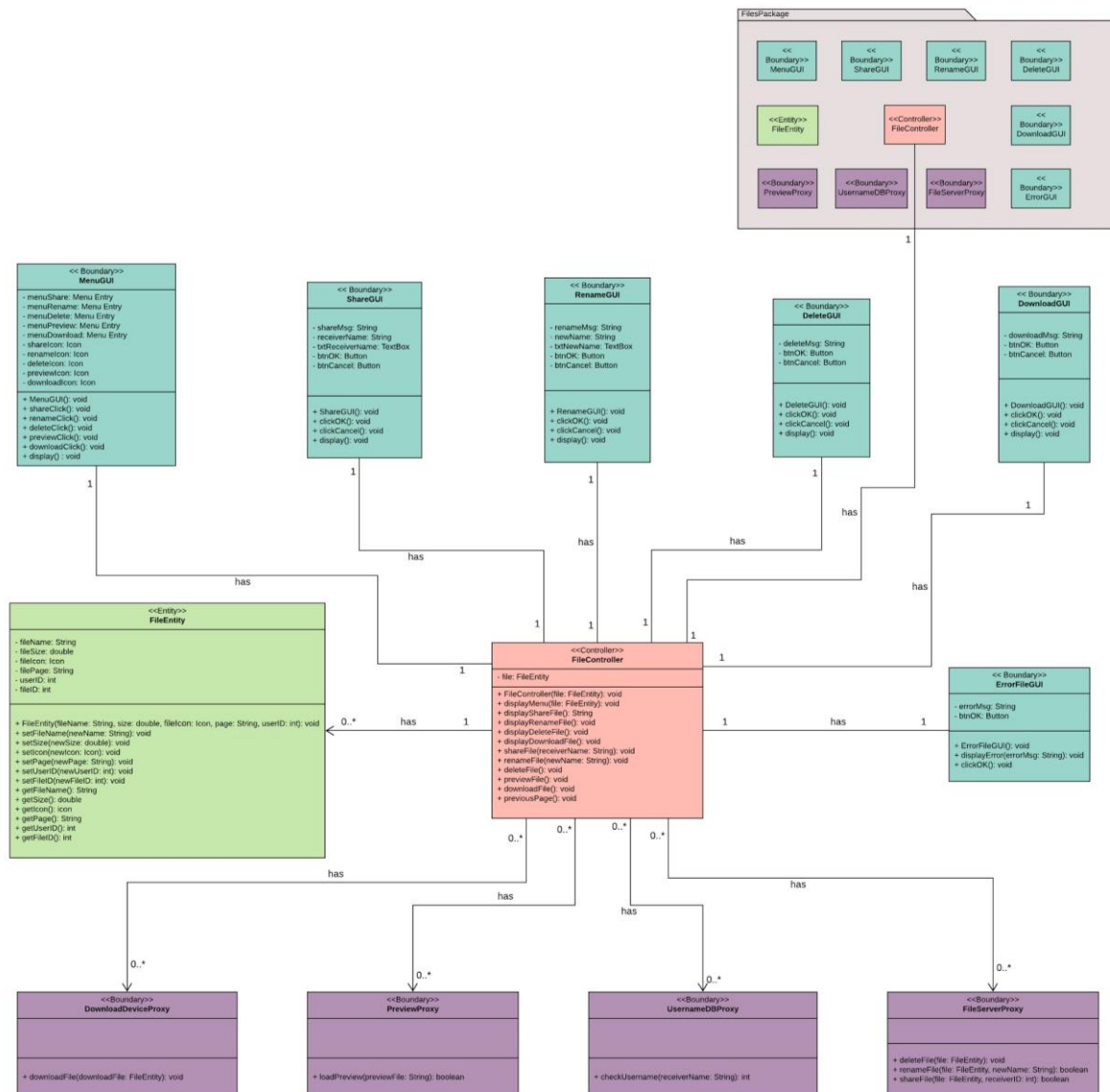
Μέθοδοι της κλάσης:

- **FileController(file: FileEntity):** Η μέθοδος δόμησης της κλάσης FileController.
- **displayMenu(file: FileEntity):** Η μέθοδος αυτή καλεί τη μέθοδο *MenuGUI::display():void*.
- **displayShareFile():** Η μέθοδος αυτή καλεί τη μέθοδο *ShareGUI::display():void* και επιστρέφει το όνομα του χρήστη (receiverName) με τον οποίο πρόκειται να γίνει ο διαμοιρασμός.
- **displayRenameFile():** Η μέθοδος αυτή καλεί τη μέθοδο *RenameGUI::display():void*.
- **displayDeleteFile():** Η μέθοδος αυτή καλεί τη μέθοδο *DeleteGUI::display():void*.
- **displayDownloadFile():** Η μέθοδος αυτή καλεί τη μέθοδο *DownloadGUI::display():void*.
- **shareFile(receiverName: String):** Η μέθοδος αυτή καλεί τη μέθοδο *UsernameDBProxy::checkUsername(receiverName: String):int*, ώστε να εξακριβωθεί αν υπάρχει το username που έδωσε ο χρήστης ως όνομα παραλήπτη. Στη συνέχεια, αφού ελέγξει ότι υπάρχει διαθέσιμος χώρος στο λογαριασμό του παραλήπτη, καλεί τη μέθοδο *FileServerProxy::shareFile(file: FileEntity, receiverID: int): boolean*, ώστε να γίνει ο διαμοιρασμός. Σε περίπτωση σφάλματος καλείται η μέθοδος *ErrorFileGUI::displayError(errorMsg: String):void*.
- **renameFile(newName: String):** Η μέθοδος αυτή καλεί τη μέθοδο *FileServerProxy::renameFile(file: FileEntity, newName: String): boolean* ώστε να αναζητηθεί και να μετονομαστεί το επιλεγμένο αρχείο στη βάση δεδομένων του συστήματος στο νέο όνομα. Αν η τελευταία επιστρέψει μηδέν, όταν δηλαδή το αρχείο έχει όνομα με μη επιτρεπούς χαρακτήρες, καλεί τη μέθοδο *ErrorFileGUI::displayError(errorMsg: String):void*.
- **deleteFile():** Η μέθοδος αυτή καλεί τη μέθοδο *FileServerProxy::deleteFile(file: FileEntity): void*, ώστε να αναζητηθεί και να διαγραφεί το επιλεγμένο αρχείο από τη βάση δεδομένων του συστήματος.



- **previewFile():** Η μέθοδος αυτή καλεί τη μέθοδο *PreviewProxy::loadPreview(file: FileEntity): boolean* ώστε να πραγματοποιηθεί η προεπισκόπηση. Αν η τελευταία επιστρέψει μηδέν, όταν δηλαδή το αρχείο έχει μη συμβατό τύπο, καλεί τη μέθοδο *ErrorFileGUI::displayError(errorMessage: String):void*.
- **downloadFile():** Η μέθοδος αυτή καλεί τη μέθοδο *DownloadDeviceProxy::downloadFile(downloadFile: FileEntity): void* ώστε να ξεκινήσει η διαδικασία λήψης του αρχείου.
- **previousPage():** Η μέθοδος αυτή καλεί τη μέθοδο *FilePageController::previousPage():void* ώστε ο χρήστης να μεταβεί στην προηγούμενη σελίδα του.

2.1.3.2 Διάγραμμα κλάσεων FilePackage



Σχήμα 3. Διάγραμμα κλάσεων πακέτου FilePackage.



3. Μη λειτουργικές απαιτήσεις

3.1 Απαιτήσεις προϊόντος

3.1.1 Απαιτήσεις απόδοσης

3.1.1.1 Απαιτήσεις ταχύτητας

<ΜΛΑ- 1>

Το σύστημα θα πρέπει να έχει χρόνο απόκρισης 250ms.

Περιγραφή: Όταν ο χρήστης εκτελεί μια ενέργεια το σύστημα θα πρέπει να αποκρίνεται στο αίτημά του σε χρόνο μικρότερο των 250ms.

User Priority (5/5): Ο χρόνος απόκρισης της εφαρμογής είναι μείζονος σημασίας για το χρήστη, καθώς είναι επιθυμητή μία γρήγορη και άμεσα αποκρίσιμη εφαρμογή.

Technical Priority (5/5): Για το σύστημα ο χρόνος απόκρισης είναι πολύ σημαντικό χαρακτηριστικό, καθώς εξαρτάται αφενός από την ταχύτητα αποστολής και παραλαβής δεδομένων από τον εξυπηρετητή και αφετέρου από την ταχύτητα επεξεργασίας της κεντρικής μονάδας.

Stability (3/5): Ο αναμενόμενος χρόνος απόκρισης του συστήματος ενδέχεται να αλλάξει σύμφωνα με τις νέες απαιτήσεις των χρηστών καθώς και την εξέλιξη της υπολογιστικής ισχύς των σύγχρονων συστημάτων.

3.1.1.2 Απαιτήσεις μεγέθους

<ΜΛΑ- 2>

Το σύστημα θα πρέπει να παρέχει τουλάχιστον 5GB χώρο αποθήκευσης στον χρήστη.

Περιγραφή: Το σύστημα θα πρέπει να έχει αρκετό αποθηκευτικό χώρο, ώστε να παρέχει στους χρήστες επαρκή χωρητικότητα για τη χρήση των υπηρεσιών της εφαρμογής.

User Priority (5/5): Η ύπαρξη επαρκούς αποθηκευτικού χώρου αποτελεί βασική απαίτηση κάθε χρήστη που χρησιμοποιεί την εφαρμογή, ώστε να μπορεί να αποθηκεύσει μεγάλο πλήθος δεδομένων.

Technical Priority(3/5): Ο όγκος που καταλαμβάνουν τα δεδομένα του χρήστη δεν επηρεάζουν σημαντικά τη λειτουργία του συστήματος.

Stability (5/5): Ο αποθηκευτικός χώρος που θα προσφέρει το σύστημα στον κάθε χρήστη δεν ενδέχεται να μεταβληθεί στο άμεσο μέλλον.

3.1.2 Απαιτήσεις αξιοπιστίας

<ΜΛΑ-4>

Το σύστημα πρέπει να διατηρεί τα δεδομένα ασφαλή.

Περιγραφή: Το σύστημα πρέπει να αποτρέπει τυχόν απώλειες δεδομένων των χρηστών.

User Priority (5/5): Για τον χρήστη η προστασία των δεδομένων του από τυχόν βλάβες του συστήματος αποτελεί απαίτηση απαραίτητης σημασίας.



Technical Priority (5/5): Η αποτροπή απώλειας δεδομένων του χρήστη καθορίζει σε εξαιρετικά μεγάλο βαθμό τον τρόπο σχεδιασμού του συστήματος. Έτσι, στην περίπτωση που το σύστημα υποστεί κάποια βλάβη είναι αναγκαία η ύπαρξη μέτρων πρόληψης, που εξασφαλίζουν την ασφάλεια των δεδομένων του χρήστη και την αξιοπιστία του συστήματος.

Stability (4/5): Καθώς οι τρόποι δημιουργίας αντιγράφων ασφαλείας θα αναπτύσσονται και θα βελτιώνονται, το σύστημα θα πρέπει να συμβαδίζει με αυτούς και να παρέχει τον ύψιστο βαθμό αξιοπιστίας στον χρήστη.

3.1.3 Απαιτήσεις μεταφερσιμότητας

<ΜΛΑ- 3>

Το σύστημα πρέπει να είναι μεταφέρσιμο.

Περιγραφή: Το σύστημα πρέπει να είναι σχεδιασμένο με τέτοιο τρόπο, έτσι ώστε να καθίσταται δυνατή η καλή λειτουργία του στα ακόλουθα λογισμικά: Android, iOS και Windows Phone.

User Priority (4/5): Για τον χρήστη είναι πολύ σημαντικό να έχει τη δυνατότητα να χρησιμοποιεί την εφαρμογή ανεξαρτήτως λογισμικού.

Technical Priority (4/5): Η απαίτηση αυτή προσδιορίζει σε μεγάλο βαθμό τον τρόπο σχεδιασμού του συστήματος. Συνεπώς, το σύστημα θα πρέπει να έχει σημαντική ευελιξία στις διάφορες αρχιτεκτονικές των λογισμικών.

Stability (5/5): Δεδομένου ότι η δημιουργία καινούργιων συστημάτων είναι αδιάλειπτη, κρίνεται αναγκαίο η εφαρμογή να είναι μεταφέρσιμη.

3.2 Εξωτερικές απαιτήσεις

3.2.1 Απαιτήσεις ιδιωτικότητας

<ΜΛΑ-5>

Το σύστημα πρέπει να προστατεύει τα δεδομένα του χρήστη από τρίτους.

Περιγραφή: Το σύστημα πρέπει να είναι σχεδιασμένο ώστε να εμποδίζει την πρόσβαση τρίτων στα δεδομένα του χρήστη.

User Priority (5/5): Ο χρήστης για να χρησιμοποιήσει το σύστημα θα πρέπει να γνωρίζει ότι τα δεδομένα του διατηρούνται ασφαλή από άλλους χρήστες.

Technical Priority (5/5): Η ασφάλεια των δεδομένων απαιτεί το σύστημα να είναι σχεδιασμένο με τέτοιο τρόπο, ώστε να μην εκμεταλλεύεται τις αδυναμίες του συστήματος κακόβουλο λογισμικό.

Stability (3/5): Εξαιτίας της συνεχούς δημιουργίας νέων κακόβουλων λογισμικών, είναι απαραίτητο οι λειτουργίες προστασίας των δεδομένων του χρήστη να αναβαθμίζονται αντίστοιχα.

4. Πρότυπα Σχεδιασμού που υιοθετήθηκαν

Τα πρότυπα σχεδιασμού που χρησιμοποιεί το σύστημα προτιμήθηκαν διότι προσφέρουν την πληρέστερη και πιο κατανοητή λύση στα προβλήματα του συστήματος και του χαρίζουν απλότητα, ευελιξία και ευχέρεια στη δημιουργία του αλλά και τη συντήρησή του. Δεν αφορούν αποκλειστικά τις μη λειτουργικές απαιτήσεις, καθώς αυτές δεν έθεταν περιορισμούς, οι οποίοι θα επέβαλαν τη χρήση συγκεκριμένων προτύπων.

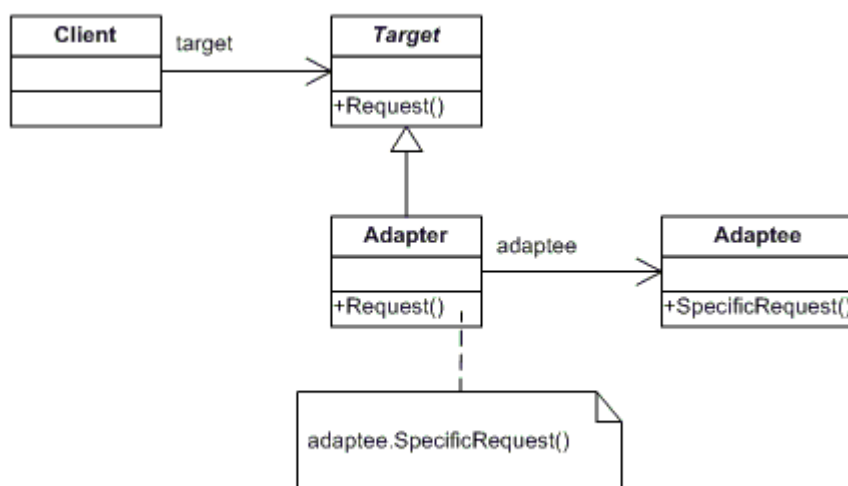
4.1 Δομικά Πρότυπα

Τα δομικά (structural) πρότυπα προσδιορίζουν τον τρόπο με τον οποίο οι κλάσεις και τα αντικείμενα συναρμολογούνται σε μεγαλύτερες δομές. Τα δομικά πρότυπα που εστιάζονται σε κλάσεις, περιγράφουν το πώς χρησιμοποιείται η σχέση κληρονομικότητας για να παράγουμε χρήσιμες διεπαφές.

4.1.1 Object Adapter Design Pattern

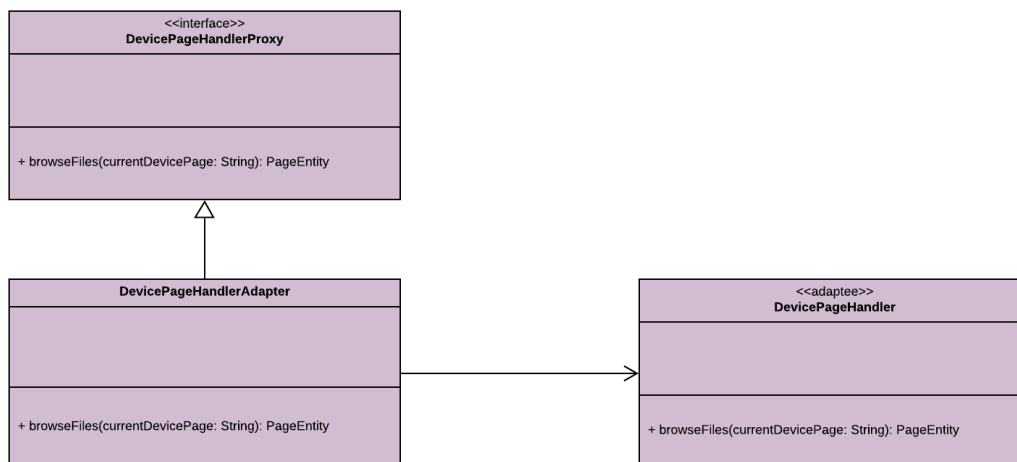
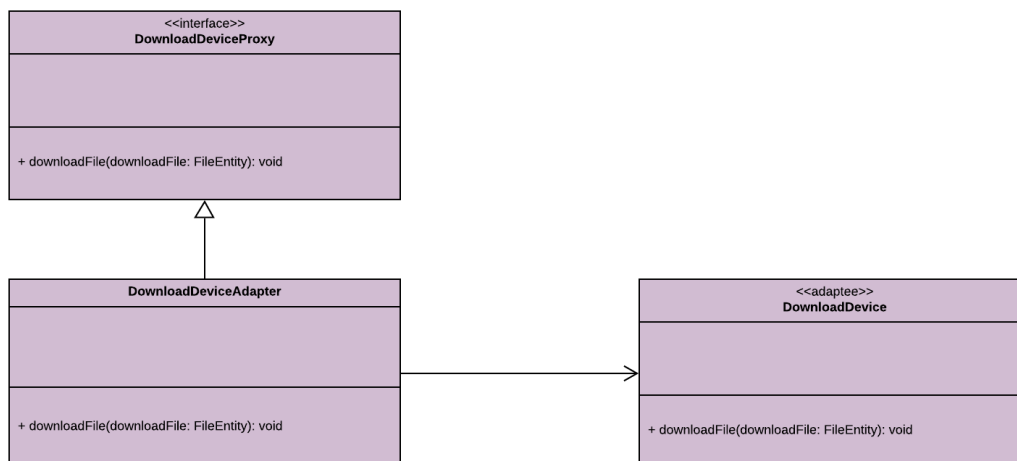
Στο πρότυπο Adapter (προσαρμοστής), ένα αντικείμενο υλοποιεί τη διεπαφή που χρησιμοποιείται από τα άλλα αντικείμενα με ενιαίο και γενικό τρόπο, έτσι ώστε τα αντικείμενα που προσπελαίνουν μια συσκευή (device) ή ένα άλλο συστατικό (component) να μην χρειάζεται να γνωρίζουν την συγκεκριμένη υλοποίηση αλλά μόνο τη διεπαφή (interface) αυτής.

- Πρόβλημα που αντιμετωπίστηκε: Συγκεκριμένα το πρότυπο adapter χρησιμοποιήθηκε για να διευκολύνει την επικοινωνία του συστήματός μας με τη συσκευή του χρήστη. Εφόσον, λοιπόν, εξασφαλίζει τη συνεργασία της εφαρμογής με τη συσκευή, ικανοποιείται η ΜΛΑ-3.



Σχήμα 4: Adapter Design Pattern

Παρακάτω εμφανίζεται η εφαρμογή του προτύπου στις κλάσεις DevicePageHandlerProxy και DownloadDeviceProxy.

Σχήμα 5: Adapter Pattern για την αρχική κλάση `DevicePageHandlerProxy`.Σχήμα 6: Adapter Pattern για την αρχική κλάση `DownloadDeviceProxy`.

4.2 Δημιουργικά Πρότυπα

Τα δημιουργικά πρότυπα αφορούν τυποποιημένους τρόπους δυναμικής κατασκευής αντικειμένων κατά τον χρόνο εκτέλεσης. Ο απλός τρόπος δημιουργίας αντικειμένων μπορεί να οδηγήσει σε σχεδιαστικά προβλήματα, είτε ακόμα και να αυξήσει την πολυπλοκότητα της εφαρμογής. Απώτερος στόχος της χρήσης τους είναι η ανεξαρτητοποίηση του κώδικα που χρησιμοποιεί κάποια αντικείμενα από τις κλάσεις που ορίζουν τα αντικείμενα αυτά και τον τρόπο που κατασκευάζονται στη μνήμη, σύμφωνα με την αρχή ανοιχτότητας-κλειστότητας για ορθή αντικειμενοστραφή σχεδίαση.

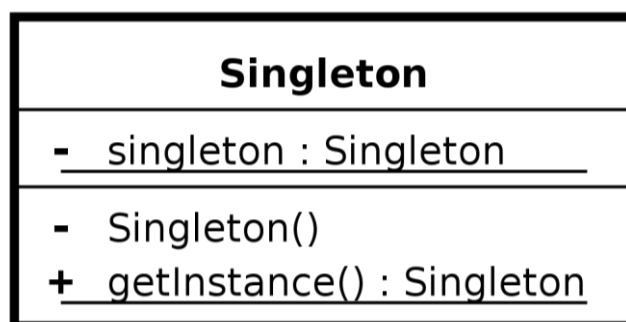
4.2.1 Singleton Design Pattern

Το πρότυπο Singleton εξασφαλίζει ότι μία κλάση θα έχει μόνο ένα στιγμιότυπο και παρέχει ένα καθολικό σημείο πρόσβασης. Συνήθως, μεταξύ των κλάσεων και των στιγμιότυπων τους υπάρχει μια σχέση ένα προς πολλά. Κατά τη διαδικασία ανάλυσης, η ύπαρξη πολλών στιγμιότυπων της ίδιας έννοιας στο

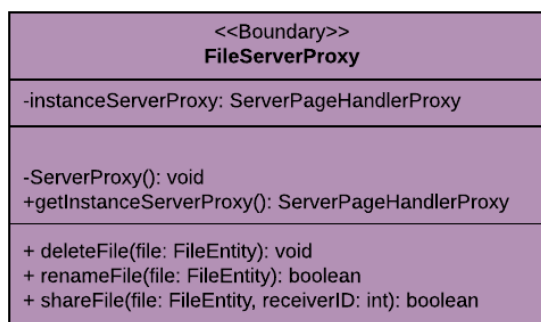
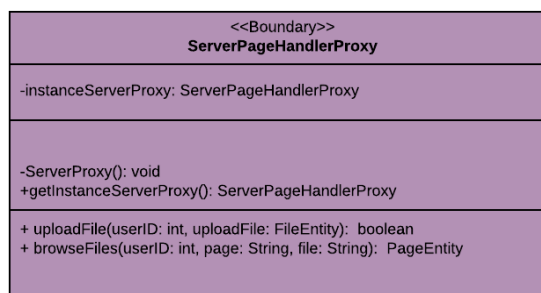


σύστημα υποδηλώνει την αναγκαιότητα μιας κλάσης. Το πρότυπο περιλαμβάνει μία ειδική μέθοδο κατασκευής στιγμιότυπων, που με την κλήση της, ελέγχει αν κάποιο αντικείμενο έχει ήδη δημιουργηθεί. Αν έχει όντως δημιουργηθεί, επιστρέφει απλώς ένα δείκτη προς το υπάρχον αντικείμενο, διαφορετικά, το δημιουργεί και έπειτα επιστρέφει ένα δείκτη προς αυτό.

- Πρόβλημα που αντιμετωπίστηκε: Συγκεκριμένα το πρότυπο Singleton χρησιμοποιήθηκε για την υλοποίηση των οριακών (boundary) κλάσεων ServerPageHandlerProxy και FileServerProxy. Με αυτόν τον τρόπο αποτρέπεται η συνεχής δημιουργία και καταστροφή αυτών των στιγμιότυπων της κλάσης, καθώς καλούνται αρκετές φορές κατά την εκτέλεση της εφαρμογής. Έτσι, αποφεύγεται η δημιουργία περισσότερων από μία κλάση-διαχειριστή, το οποίο αποτελεί λογικό λάθος.



Σχήμα 7: Singleton Pattern Design Pattern.



Σχήμα 8: Το πρότυπο Singleton όπως εφαρμόστηκε στις οριακές κλάσεις.

4.3 Πρότυπα Συμπεριφοράς

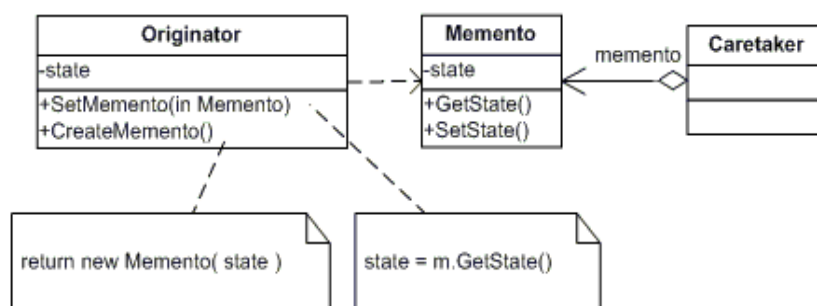
Τα πρότυπα συμπεριφοράς (behavioral design patterns) αφορούν την αναγνώριση κοινών επικοινωνιακών μοτίβων ανάμεσα στα αντικείμενα και την υλοποίηση τους. Με τον τρόπο αυτό, αυξάνεται η ευελιξία στην εκτέλεση αυτών των επικοινωνιών.



4.3.1 Memento Design Pattern

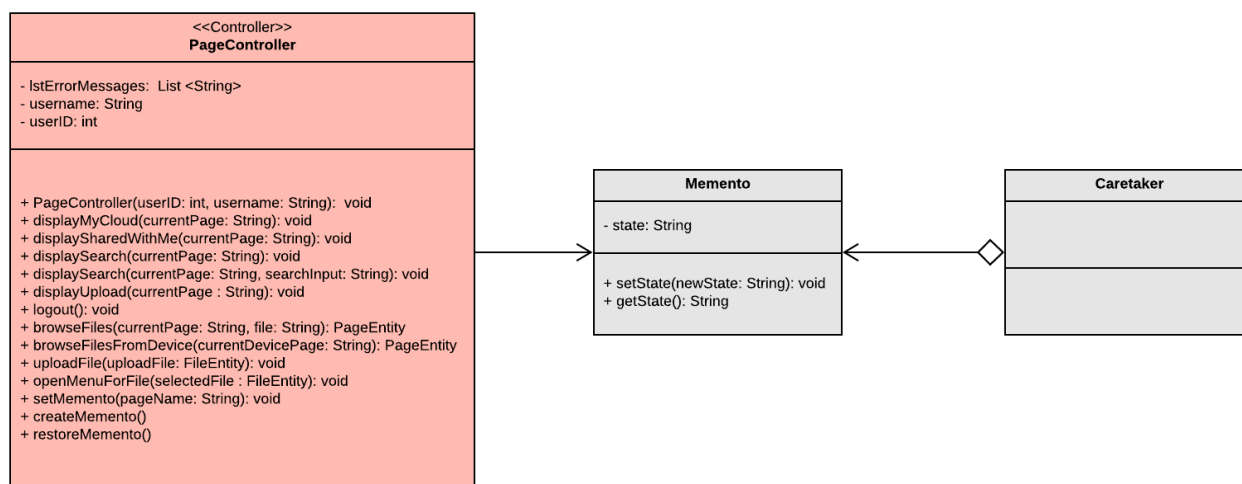
Το πρότυπο Memento χρησιμοποιείται για την επαναφορά της κατάστασης ενός αντικειμένου σε μια προηγούμενη κατάσταση. Αποτελείται από τρεις κλάσεις, που ονομάζονται Memento, Originator και CareTaker. Η κλάση Memento περιέχει την κατάσταση του αντικειμένου προς αποκατάσταση. Η κλάση Originator δημιουργεί και αποθηκεύει καταστάσεις στα αντικείμενα Memento, ενώ το αντικείμενο της κλάσης Caretaker είναι υπεύθυνο για την επαναφορά της κατάστασης από το Memento.

- Πρόβλημα που αντιμετωπίστηκε: Συγκεκριμένα το πρότυπο memento χρησιμοποιήθηκε για να διευκολύνει την εμπειρία του χρήστη, δίνοντάς του τη δυνατότητα να πλοηγείται σε προηγούμενες σελίδες του συστήματος. Αποθηκεύει, δηλαδή, το όνομα της εκάστοτε σελίδας που εμφανίζεται, ώστε να μπορεί το σύστημα, σε μετέπειτα κατάσταση, να επιστρέψει σε αυτήν εφόσον ο χρήστης το επιθυμεί.



Σχήμα 9: Memento Design Pattern

Παρακάτω εμφανίζεται η εφαρμογή του προτύπου στην κλάση PageController.



Σχήμα 10: Memento Pattern για την αρχική κλάση PageController.



Παράρτημα Ι – Πίνακας Ιχνηλασιμότητας

Η μεταβολή από το Έγγραφο Απαιτήσεων Χρηστών στο Έγγραφο Απαιτήσεων Λογισμικού δεν εμπεριείχε κάποιες μεταβολές του αρχικού, συνεπώς δεν παρατίθεται Πίνακας Ιχνηλασιμότητας. Οι διαφοροποιήσεις που οφείλονται στα πρότυπα σχεδιασμού, έχουν ήδη αναλυθεί στο Κεφάλαιο 4 και συνεπώς δεν επαναλαμβάνονται.



Παράρτημα II – Ανοιχτά Θέματα

- Δεδομένων των μετρίων τιμών της σταθερότητας (stability) των μη λειτουργικών απαιτήσεων ΜΛΑ-1, ΜΛΑ-3 και λιγότερο της ΜΛΑ-2, σε περίπτωση που αυτές τελικά μεταβληθούν και το σύστημα υπόκειται πλέον σε νέους περιορισμούς, το έγγραφο αυτό πρέπει να ανανεωθεί σύμφωνα με τα νέα δεδομένα.
- Ο τύπος ορισμένων ιδιοτήτων των κλάσεων πιθανώς να αλλάξει στο μέλλον ή να προστεθούν νέες.
- Οι συναρτήσεις οι οποίες έχουν δημιουργηθεί περιγράφουν τις βασικές λειτουργικότητες όπως εντοπίστηκαν από την ομάδα ανάπτυξης. Κατά την υλοποίηση του συστήματος είναι πιθανό να χρειαστούν επιπλέον συναρτήσεις για την ορθότερη λειτουργία του συστήματος.