



BOOK
FELAS

Where every book finds a home

Προδιαγραφές Συστήματος

Del.1.2

Version 0.1

10/05/24

Μέλη της Ομάδας Ανάπτυξης

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	3
1 Στατική Μοντελοποίηση	6
1.1 <Library Package>	6
1.1.1 Οντότητες.....	7
1.1.1.1 Entity <Book>.....	7
1.1.1.2 Entity <Library>	8
1.1.2 Ελεγχτές.....	9
1.1.2.1 Controller <BookController>.....	9
1.1.2.2 Controller <LibraryController>	9
1.1.3 Οριακά αντικείμενα	10
1.1.3.1 Boundary <LibraryGUI>	10
1.1.3.2 Boundary <BookPageGUI>.....	10
1.1.3.3 Boundary <BookDetailsPageGUI>	11
1.1.3.4 Boundary <EditBookPageGUI>	12
1.1.3.5 Boundary <BookDBProxy>.....	12
1.2 <Rating Package>.....	13
1.2.1 Οντότητες.....	14
1.2.1.1 Entity <Rating>.....	14
1.2.2 Ελεγχτές.....	15
1.2.2.1 Controller <RatingController>.....	15
1.2.3 Οριακά αντικείμενα	15
1.2.3.1 Boundary <CreateRatingGUI>	15
1.2.3.2 Boundary <EditRatingGUI>	16
1.2.3.3 Boundary <RatingCardGUI>	16
1.2.3.4 Boundary <RatingDBProxy>.....	17
1.3 <Like Package>	17
1.3.1 Οντότητες.....	17
1.3.1.1 Entity <LikeList>.....	17
1.3.1.2 Entity <Book>.....	18
1.3.1.3 Entity <User>.....	18
1.3.1.4 Entity <Library>	18
1.3.2 Ελεγχτές.....	18
1.3.2.1 Controller <LikeListController>.....	18

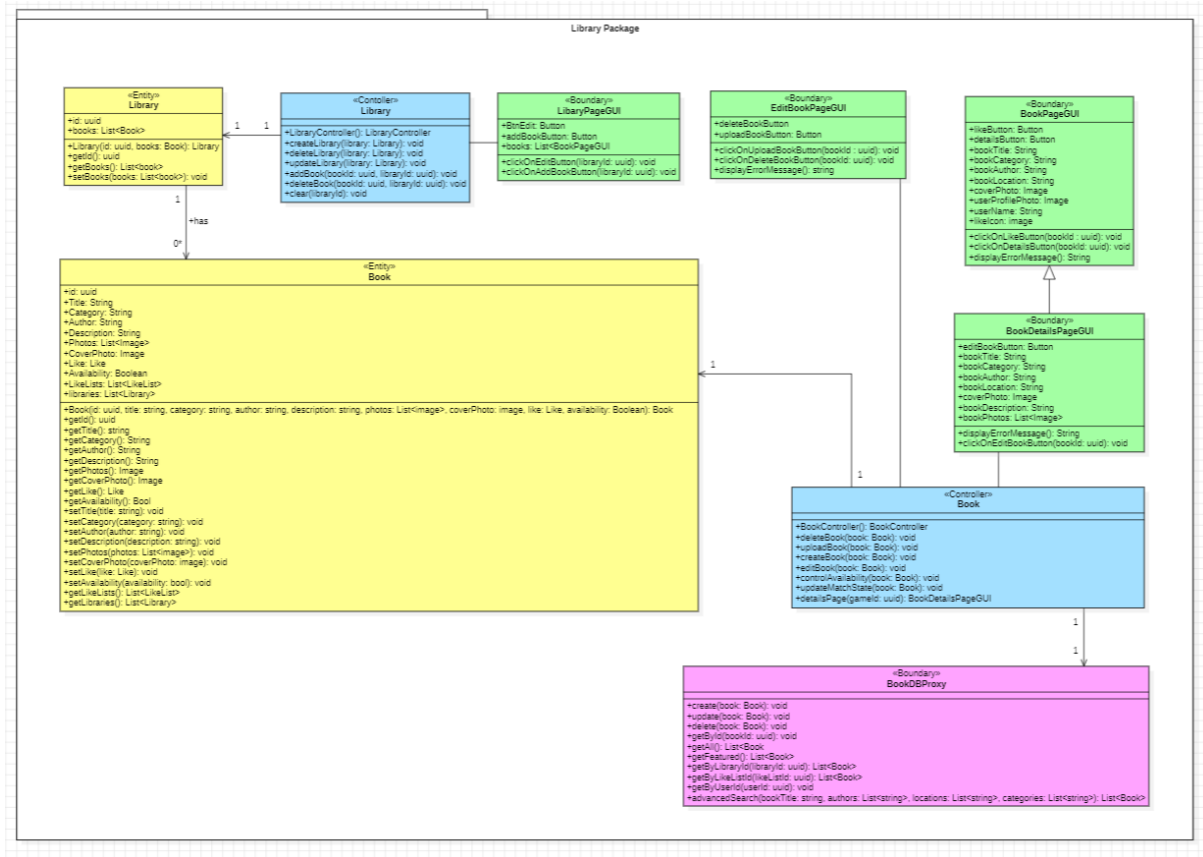
1.3.2.2 Controller <LikeController>	18
1.3.3 Οριακά αντικείμενα	19
1.3.3.1 Boundary <LikeListGUI>	19
1.3.3.2 Boundary <BookPageGUI>	19
1.4 <Search Package>	19
1.4.1 Ελεγχτές	20
1.4.1.1 Controller <SearchController>	20
1.4.1.2 Controller <BookController>	20
1.4.2 Οριακά αντικείμενα	20
1.4.2.1 Boundary <FilterListPageGUI>	20
1.4.2.2 Boundary <SortOptionsPageGUI>	21
1.4.2.3 Boundary <SearchPageGUI>	22
1.4.2.4 Boundary <BookDBProxy>	22
1.5 <Exchange Process Package>	23
1.5.1 Οντότητες	23
1.5.1.1 Entity <Chat>	23
1.5.2 Ελεγχτές	24
1.5.2.1 Controller <ChatController>	24
1.5.2.2 Controller <NotificationController>	24
1.5.3 Οριακά αντικείμενα	24
1.5.3.1 Boundary <ChatPageGUI>	24
1.5.3.2 Boundary <NotificationPageGUI>	25
1.5.3.3 Boundary <ExchangeDialog>	26
1.6 <Navigate Package>	26
1.6.1 Οντότητες	26
1.6.1.1 Entity <User>	26
1.6.2 Ελεγχτές	27
1.6.2.1 Controller <UserController>	27
1.6.2.2 Controller <HomeController>	28
1.6.2.3 Controller <BookController>	28
1.6.2.4 Controller <ChatController>	28
1.6.2.5 Controller <LibraryController>	28
1.6.2.6 Controller <LikeListController>	28
1.6.3 Οριακά αντικείμενα	28
1.6.3.1 Boundary <UserProfilePageGUI>	28

1.6.3.2 Boundary <EditProfilePageGUI>	29
1.6.3.3 Boundary <HomePageGUI>	29
1.6.3.4 Boundary <LibraryGUI>	30
1.6.3.5 Boundary <RatingCardGUI>	30
1.6.3.6 Boundary <BookDetailsPageGUI>	30
1.6.3.7 Boundary <LikeListGUI>	30
1.6.3.8 Boundary <BookPageGUI>	30
1.6.3.9 Boundary <UserDBProxy>	30
2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν	31
2.1 Πρότυπο Abstract Factory	31
2.2 Πρότυπο Proxy	31
2.3 Πρότυπο Adapter	32
3 Δυναμική Μοντελοποίηση	33
3.1 Exchange Process Package	33
3.1.1 Αφήγηση Σεναρίου	33
3.1.2 Διάγραμμα ακολουθιών	33
3.2 Library Package	33
3.2.1 Αφήγηση Σεναρίου	33
3.2.2 Διάγραμμα ακολουθιών	34
3.2 Search Package	34
3.2.1 Αφήγηση Σεναρίου	34
3.2.2 Διάγραμμα ακολουθιών	35
Παράρτημα Ι – Γλωσσάριο	36

1 Στατική Μοντελοποίηση

1.1 <Library Package>

Το παρόν πακέτο αναλύει τις κλάσεις του **Feature 1: Edit Library**, **Feature 10: System controls book availability**.



1.1.1 Οντότητες

1.1.1.1 Entity <Book>

«Entity» Book
+id: uuid +Title: String +Category: String +Author: String +Description: String +Photos: List<Image> +CoverPhoto: Image +Like: Like +Availability: Boolean +LikeLists: List<LikeList> +libraries: List<Library>
+Book(id: uuid, title: string, category: string, author: string, description: string, photos: List<image>, coverPhoto: image, like: Like, availability: Boolean): Book +getId(): uuid +getTitle(): string +getCategory(): String +getAuthor(): String +getDescription(): String +getPhotos(): Image +getCoverPhoto(): Image +getLike(): Like +getAvailability(): Bool +setTitle(title: string): void +setCategory(category: string): void +setAuthor(author: string): void +setDescription(description: string): void +setPhotos(photos: List<image>): void +setCoverPhoto(coverPhoto: image): void +setLike(like: Like): void +setAvailability(availability: bool): void +getLikeLists(): List<LikeList> +getLibraries(): List<Library>

Χαρακτηριστικά κλάσης

- <id>:<uuid> : Το βασικό κλειδί κι ευρετήριο της οντότητας.
- <title>:<string>: Ο τίτλος του βιβλίου.
- <category>:<string>: Η κατηγορία στην οποία ανήκει το βιβλίο.
- <author>:<string>: Ο συγγραφέας του βιβλίου.
- <description>:<string>: Η περιγραφή του βιβλίου.
- <photos>:<List<image>> : Η λίστα με τις φωτογραφίες του βιβλίου.
- <coverPhoto>:<image>: Η φωτογραφία εξωφύλλου του βιβλίου.
- <like>:<boolean>: Αν έχει γίνει ή όχι like στο βιβλίο.
- <availability>:<boolean>: Η διαθεσιμότητα του βιβλίου.
- <match>:<boolean>: Δηλώνει αν το βιβλίο είναι σε matching process ή όχι.
- <likeLists>: List<LikeList>: Η λίστα με τις λίστες αγαπημένων στις οποίες βρίσκεται το βιβλίο.
- <libraries>:List<Library>: Η λίστα με τις βιβλιοθήκες στις οποίες βρίσκεται το βιβλίο.

Μέθοδοι κλάσης

- <Book>(id: uuid, title: string, category: string, author: string, description: string, photos: List<image>, coverPhoto: image, like: boolean, availability: boolean, match: boolean, <likeLists>: List<LikeList>, <libraries>:List<Library>):<BookEntity>: Η συνάρτηση δόμησης της κλάσης. Αρχικοποιεί τα χαρακτηριστικά της κλάσης με βάση τα ορίσματα.
- <getId()>: <uuid>: Επιστρέφει το ευρετήριο της οντότητας.
- <getTitle()>: <string>: Επιστρέφει τον τίτλο του βιβλίου.
- <setTitle>(title: string):<void>: Αλλάζει την τιμή του τίτλου με αυτήν που δίνεται ως όρισμα.
- <getCategory>():<string>: Επιστρέφει την κατηγορία του βιβλίου.
- <setCategory>(category:string):<void>: Αλλάζει την τιμή της κατηγορίας με αυτήν που δίνεται ως όρισμα.
- <getAuthor>():<string>: Επιστρέφει το συγγραφέα του βιβλίου.

- `<setAuthor>(author: string):<void>`: Αλλάζει την τιμή του συγγραφέα με αυτήν που δίνεται ως όρισμα.
- `<getDescription>():<string>`: Επιστρέφει την περιγραφή του βιβλίου.
- `<setDescription>(description: string):<void>`: Αλλάζει την τιμή της περιγραφής με αυτήν που δίνεται ως όρισμα.
- `<getPhotos>():<List<image>>`: Επιστρέφει τη λίστα με τις φωτογραφίες του βιβλίου.
- `<setPhotos>(photos: List<image>):<void>`: Αλλάζει τη λίστα φωτογραφιών με αυτή που δίνεται ως όρισμα.
- `<getCoverPhoto>():<image>`: Επιστρέφει τη φωτογραφία εξωφύλλου.
- `<setCoverPhoto>(coverPhoto: image):<void>`: Αλλάζει τη φωτογραφία εξωφύλλου με αυτή που δίνεται ως όρισμα.
- `<getLike>():<boolean>`: Επιστρέφει την τιμή της μεταβλητής like.
- `<setLike>(like: boolean):<void>`: Αλλάζει την τιμή της μεταβλητής like με αυτή που δίνεται ως όρισμα.
- `<getAvailability>():<boolean>`: Επιστρέφει τη μεταβλητή διαθεσιμότητας.
- `<setAvailability>(availability: boolean):<void>`: Αλλάζει την τιμή της μεταβλητής διαθεσιμότητας με αυτή που δίνεται ως όρισμα.
- `<getLikeLists>():<List<LikeList>>`: Επιστρέφει τη λίστα με τις λίστες αγαπημένων στις οποίες βρίσκεται το βιβλίο.
- `<getLibraries>():<List<Library>>`: Επιστρέφει τη λίστα με τις βιβλιοθήκες στις οποίες βρίσκεται το βιβλίο.

UUID: A **Universally Unique Identifier** (UUID) is a label used to uniquely identify a resource among all other resources of that type ([What's a uuid?](#)).

ΣΗΜΕΙΩΣΗ: Σκοπίμως παραλείπεται setter για το id καθώς αυτό είναι κάτι που γίνεται από τη βάση.

1.1.1.2 Entity <Library>

«Entity» Library
+id: uuid +books: List<Book>
+Library(id: uuid, books: Book): Library +getId(): uuid +getBooks(): List<book> +setBooks(books: List<book>): void

Χαρακτηριστικά κλάσης

- `<id: uuid>`: Αποτελεί το βασικό κλειδί και ευρετήριο της οντότητας.
- `<books: List<Book>>`: Αποτελεί τη λίστα με τα βιβλία της βιβλιοθήκης του χρήστη.

Μέθοδοι κλάσης

- `<getLibraryId(): uuid>`: Επιστρέφει το διακριτικό χαρακτηριστικό αριθμό της βιβλιοθήκης.
- `<getBooks(): List<Book>>`: Επιστρέφει τα βιβλία της βιβλιοθήκης.
- `<setBooks(books: List<Book>): void>`: Ανανεώνει τα βιβλία της βιβλιοθήκης με αυτά που δίνονται ως όρισμα.
- `<Library(id: uuid, books: Book):Library>`: Η συνάρτηση δόμησης της κλάσης.

1.1.2 Ελεγκτές

1.1.2.1 Controller <BookController >

«Controller» Book
+BookController(): Book +deleteBook(book: Book): void +uploadBook(book: Book): void +createBook(book: Book): void +editBook(book: Book): void +controlAvailability(book: Book): void +updateMatchState(book: Book): void +detailsPage(gameId: uuid): BookDetailsPageGUI

Μέθοδοι κλάσης

- <BookController():<BookController>: Η συνάρτηση δόμησης της κλάσης.
- <createBook>(book: Book): <void>: Προσθέτει ένα καινούριο βιβλίο στη βάση δεδομένων.
- <updateBook>(book: Book):<void>: Αλλάζει τα χαρακτηριστικά ενός υπάρχοντος βιβλίου στη βάση δεδομένων.
- <deleteBook>(book: Book):<void>: Διαγράφει ένα βιβλίο από τη βάση δεδομένων.
- <controlAvailability>(book: Book):<void>: Μεταβάλλει τη διαθεσιμότητα του βιβλίου.
- <updateMatchState(book:Book):void>: Ανανεώνει το match state του βιβλίου.
- <detailsPage(gameId: uuid): BookDetailsPageGUI>: Επιστρέφει το οριακό αντικείμενο της σελίδας με τις λεπτομέρειες του βιβλίου.
- <editPage(gameId: uuid): EditBookPageGUI>: Επιστρέφει το οριακό αντικείμενο της σελίδας επεξεργασίας του βιβλίου.

1.1.2.2 Controller <LibraryController >

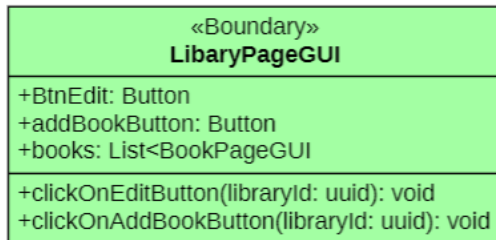
«Contollor» Library
+LibraryController(): LibraryController +createLibrary(library: Library): void +deleteLibrary(library: Library): void +updateLibrary(library: Library): void +addBook(bookId: uuid, libraryId: uuid): void +deleteBook(bookId: uuid, libraryId: uuid): void +clear(libraryId): void

Μέθοδοι κλάσης

- <LibraryController():LibraryController>: Η συνάρτηση δόμησης της κλάσης.
- <createLibrary>(library: Library):void>: Δημιουργεί μια νέα βιβλιοθήκη στη βάση δεδομένων.
- <updateLibrary>(library: Library): void> : Ανανεώνει βιβλιοθήκη που ήδη υπάρχει.
- <deleteLibrary>(library: Library): void>: Διαγράφει μια βιβλιοθήκη από τη βάση δεδομένων.
- <addBook>(bookId:uuid, libraryId: uuid): void>: Προσθέτει ένα καινούριο βιβλίο στη βιβλιοθήκη.
- <deleteBook>(bookId: uuid, libraryId: uuid): void>: Διαγράφει ένα υπάρχον βιβλίο από τη βιβλιοθήκη.
- <clear>(libraryId):void>: Αφαιρεί όλα τα παιχνίδια από τη βιβλιοθήκη.

1.1.3 Οριακά αντικείμενα

1.1.3.1 Boundary <LibraryGUI >



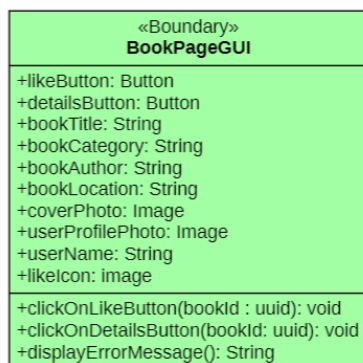
Χαρακτηριστικά κλάσης

- <editButton: [button](#)> :Κουμπί επιλογής τροποποίησης βιβλιοθήκης.
- <addBookButton: [button](#)> : Κουμπί επιλογής προσθήκης νέου βιβλίου.
- <books: List<[BookPageGUI](#)>: Η λίστα με τα οριακά αντικείμενα της σελίδας βιβλίου που φαίνονται στη βιβλιοθήκη.

Μέθοδοι κλάσης

- <clickOnEditButton(libraryId: [uuid](#)): [void](#)>: Καλεί την [updateLibrary\(library: \[Library\]\(#\)\)](#) του [LibraryController](#).
- <clickOnAddBookButton(libraryId: [uuid](#)): [void](#)>: Καλεί την [addBook\(bookId:\[uuid\]\(#\), libraryId: \[uuid\]\(#\)\)](#) του [LibraryController](#).

1.1.3.2 Boundary <BookPageGUI >



Χαρακτηριστικά κλάσης

- <likeButton: [button](#)>: Το κουμπί για να κάνεις like στο βιβλίο.
- <detailsButton: [button](#)>: Το κουμπί για να δεις τις λεπτομέρειες του βιβλίου.
- <bookTitle: [string](#)>: Ο επίσημος τίτλος του βιβλίου.
- <bookCategory: [string](#)>: Η κατηγορία που ανήκει το βιβλίο.
- <bookAuthor: [string](#)>: Ο συγγραφέας του βιβλίου.
- <bookLocation: [string](#)>: Η τοποθεσία του βιβλίου.
- <coverPhoto: [image](#)>: Το εξώφυλλο του βιβλίου που εμφανίζεται ως η επίσημη εικόνα.
- <userProfilePhoto: [image](#)>: Η εικόνα προφίλ του χρήστη που έχει το βιβλίο.

- <userName: **string**>: Το username του χρήστη που έχει το βιβλίο.
- <likeIcon: **image**>: Το εικονίδιο του like.

Μέθοδοι κλάσης

- <clickOnLikeButton(bookId : **uuid**): **void**>: Καλεί τη συνάρτηση **updateLike**(bookId: **uuid**): **void** του **LikeController** και αλλάζει το εικονίδιο likeIcon: **image** (σε έγχρωμο ή άχρωμο).
- <clickOnDetailsButton(bookId : **uuid**): **void**>: Καλεί την **detailsPage**(gameId: **uuid**): **BookDetailsPageGUI** του **BookController**.
- <displayErrorMessage(): **string**>: Εμφανίζει μήνυμα σφάλματος.

1.1.3.3 Boundary <BookDetailsPageGUI >

«Boundary» BookDetailsPageGUI
+editBookButton: Button +bookTitle: String +bookCategory: String +bookAuthor: String +bookLocation: String +coverPhoto: Image +bookDescription: String +bookPhotos: List<Image>
+displayErrorMessage(): String +clickOnEditBookButton(bookId: uuid): void

Χαρακτηριστικά κλάσης

- <editBookButton: **button**> : Το κουμπί για να επεξεργαστείς τις λεπτομέρειες του βιβλίου.
- <bookTitle: **string**>: Ο επίσημος τίτλος του βιβλίου.
- <bookCategory: **string**>: Η κατηγορία που ανήκει το βιβλίο.
- <bookAuthor: **string**>: Ο συγγραφέας του βιβλίου.
- <bookLocation: **string**>: Η τοποθεσία του βιβλίου.
- <coverPhoto: **image**>: Το εξώφυλλο του βιβλίου που εμφανίζεται ως η επίσημη εικόνα.
- <bookDescription: **string**>: Η περιγραφή του βιβλίου.
- <bookPhotos: List<**image**>>: Η λίστα με τις υπόλοιπες εικόνες του βιβλίου.

Μέθοδοι κλάσης

- <clickOnEditBookButton(bookId : **uuid**): **void**>: Καλεί την **editPage**(gameId: **uuid**): **EditBookPageGUI** του **BookController**.

1.1.3.4 Boundary <EditBookPageGUI>

«Boundary» EditBookPageGUI
+deleteBookButton +uploadBookButton: Button
+clickOnUploadBookButton(bookId : uuid): void +clickOnDeleteBookButton(bookId: uuid): void +displayErrorMessage(): string

Χαρακτηριστικά κλάσης

- <deleteBookButton: **button**>: Το κουμπί για να διαγράψεις το βιβλίο.
- <uploadBookButton: **button**>: Το κουμπί για να ανεβάσεις το βιβλίο (είτε καινούριο είτε ανανεωμένο).

Μέθοδοι κλάσης

- <clickOnUploadBookButton(bookId : **uuid**): **void**>: Καλεί την **updateBook**(book: **Book**) του **BookController**.
- <clickOnDeleteBookButton(bookId : **uuid**): **void**>: Καλεί την **deleteBook**(book: **Book**) του **BookController** και την **deleteBook**(bookId: **uuid**, libraryId: **uuid**) του **LibraryController** και την **deleteBook**(bookId: **uuid**, likeListId: **uuid**) από το **LikeListController**.
- <displayErrorMessage(): **string**>: Παρατίθεται μήνυμα σφάλματος από το σύστημα.

1.1.3.5 Boundary <BookDBProxy>

«Boundary» BookDBProxy
+create(book: Book): void +update(book: Book): void +delete(book: Book): void +getById(bookId: uuid): void +getAll(): List<Book> +getFeatured(): List<Book> +getByLibraryId(libraryId: uuid): List<Book> +getByLikeListId(likeListId: uuid): List<Book> +getByUserId(userId: uuid): void +advancedSearch(bookTitle: string, authors: List<string>, locations: List<string>, categories: List<string>): List<Book>

- <create(book: **Book**): **void**>: Καλεί query για τη δημιουργία ενός βιβλίου στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος book.
- <update(book: **Book**): **void**>: Καλεί query για ανανέωση ενός βιβλίου στη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος book.
- <delete(book: **Book**): **void**>: Καλεί query για διαγραφή ενός βιβλίου από τη βάση δεδομένων με τα χαρακτηριστικά του αντικειμένου ορίσματος book.
- <getById(bookId: **uuid**): **void**>: Καλεί query για την εύρεση ενός βιβλίου στη βάση δεδομένων με το κλειδί που δίνεται ως όρισμα.
- <getAll(): List<**Book**>: Καλεί query για την εύρεση όλων των βιβλίων της βάσης.
- <getFeatured(): List<**Book**>: Καλεί query για την εύρεση των καινούριων βιβλίων στη βάση με σκοπό να τα διαφημίσει στην αρχική σελίδα.

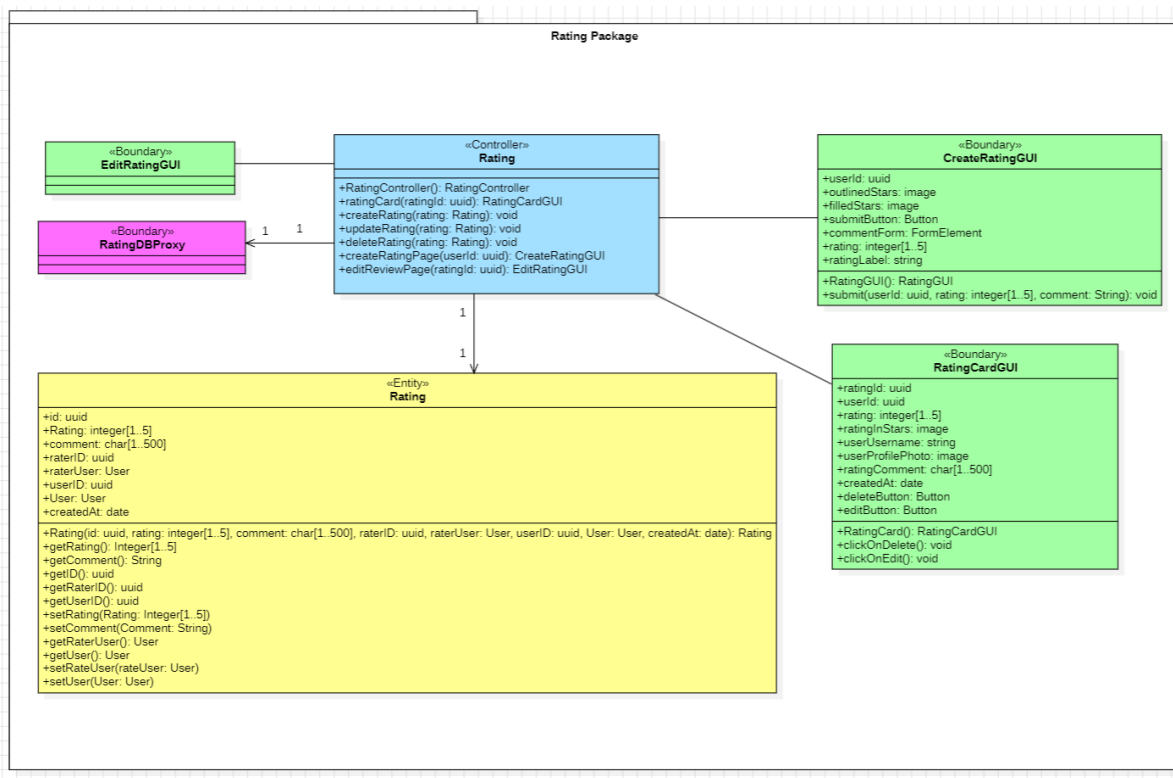
- `<getByLibraryId(libraryId: uuid): List<Book>>`: Καλεί query για την εύρεση όλων των βιβλίων που ανήκουν σε μια λίστα `Library` με βάση το κλειδί της βιβλιοθήκης που δίνεται ως όρισμα.
- `<getByLikeListId(likeListId: uuid): List<Book>>`: Καλεί query για την εύρεση όλων των βιβλίων που ανήκουν σε μια λίστα `LikeList` με βάση το κλειδί της λίστας που δίνεται ως όρισμα.
- `<getByUserId(userId: uuid): void>`: Καλεί query για την εύρεση ενός βιβλίου στη βάση δεδομένων με βάση το κλειδί του χρήστη στον οποίο ανήκει, που δίνεται ως όρισμα.
- `<advancedSearch(bookTitle: string, authors: List<string>, locations: List<string>, categories: List<string>): List<Book>`: Καλεί query για την εύρεση των βιβλίων που αντιστοιχούν στα φίλτρα που δίνονται μέσω των ορισμάτων.

ΣΗΜΕΙΩΣΗ 1: query είναι ένα data request στη βάση, αίτημα δηλαδή για να λάβει κάποια πληροφορία.

ΣΗΜΕΙΩΣΗ 2: Για τη δόμηση της κλάσης ακολουθούμε τις μεθόδους CRUD.

1.2 <Rating Package>

Το πακέτο αυτό αναλύει τις κλάσεις των **Feature 7: Get rating notification** και **Feature 9: Rate user**.



1.2.1 Οντότητες

1.2.1.1 Entity <Rating >

«Entity» Rating
+id: uuid +Rating: integer[1..5] +comment: char[1..500] +raterID: uuid +raterUser: User +userID: uuid +User: User +createdAt: date
+Rating(id: uuid, rating: integer[1..5], comment: char[1..500], raterID: uuid, raterUser: User, userID: uuid, User: User, createdAt: date): Rating +getRating(): Integer[1..5] +getComment(): String +getID(): uuid +getRaterID(): uuid +getUserID(): uuid +setRating(Rating: Integer[1..5]) +setComment(Comment: String) +getRaterUser(): User +getUser(): User +setRateUser(rateUser: User) +setUser(User: User)

Χαρακτηριστικά κλάσης

- <id: **uuid**>: Το βασικό κλειδί και ευρετήριο της οντότητας.
- <rating: **integer**[1..5]>: Η βαθμολογία της κριτικής.
- <comment: **char**[1..500]>: Σχόλιο κριτικής.
- <raterID: **uuid**>: Το ξένο κλειδί για την σύνδεση με την οντότητα του χρήστη που κάνει την κριτική (rateUser).
- <raterUser: **User**>: Ο χρήστης **User** για τον οποίο γράφεται η κριτική .
- <userID: **uuid**>: Το ξένο κλειδί για την σύνδεση με την οντότητα του χρήστη που δέχεται την κριτική.
- <User: **User**>: Ο χρήστης **User** που κάνει την κριτική.
- <createdAt: **date**>: Η ημερομηνία που έγινε η κριτική.

Μέθοδοι κλάσης

- <Rating(id: **uuid**, rating: **integer**[1..5], comment: **char**[1..500], raterID: **uuid**, raterUser: **User**, userID: **uuid**, User: **User**, createdAt: **date**): **Rating**>: Η συνάρτηση δόμησης της κλάσης.
- <getRating():**Integer** [1..5]>: Επιστρέφει την μεταβλητή rating.
- <getComment(): **char**[1..500]>: Επιστρέφει την μεταβλητή comment
- <getID(): **uuid**>: Επιστρέφει την μεταβλητή ID
- <getRaterID(): **uuid**>: Επιστρέφει την μεταβλητή RaterID.
- <getUserID(): **uuid**>: Επιστρέφει την μεταβλητή UserID.
- <getRaterUser(): **User**>: Επιστρέφει την μεταβλητή raterUser
- <getUser(): **User**>: Επιστρέφει την μεταβλητή **User**
- <setRating(Rating:**Integer** [1..5])>: Αλλάζει την μεταβλητή rating με την τιμή που δίνεται ως όρισμα
- <setComment(Comment:**char**[1..500])>:Αλλάζει την μεταβλητή comment με την τιμή που δίνεται ως όρισμα.
- <setRateUser(rateUser:**User**)>:Αλλάζει την μεταβλητή rateUser με την τιμή που δίνεται ως όρισμα.
- <setUser(user:**User**)>:Αλλάζει την μεταβλητή user με την τιμή που δίνεται ως όρισμα.

1.2.2 Ελεγκτές

1.2.2.1 Controller <RatingController >

«Controller» Rating
+RatingController(): RatingController +ratingCard(ratingId: uuid): RatingCardGUI +createRating(rating: Rating): void +updateRating(rating: Rating): void +deleteRating(rating: Rating): void +createRatingPage(userId: uuid): CreateRatingGUI +editReviewPage(ratingId: uuid): EditRatingGUI

Μέθοδοι κλάσης

- <RatingController(): RatingController>: Η συνάρτηση δόμησης της κλάσης.
- <ratingCard(ratingId: uuid): RatingCardGUI>: Επιστρέφει τη σελίδα με τις λεπτομέρειες της κριτικής.
- <createRating(rating: Rating): void>: Προσθέτει μια καινούρια κριτική στη βάση δεδομένων.
- <updateRating(rating: Rating): void>: Ανανεώνει τα χαρακτηριστικά μιας υπάρχουσας κριτικής στη βάση δεδομένων.
- <deleteRating(rating: Rating): void>: Διαγράφει μια κριτική από τη βάση δεδομένων.
- <createRatingPage(userId: uuid): CreateRatingGUI>: Επιστρέφει το οριακό αντικείμενο της σελίδας όπου ο χρήστης μπορεί να αφήσει κριτική.
- <editRatingPage(ratingId: uuid): EditRatingGUI>: Επιστρέφει το οριακό αντικείμενο της σελίδας όπου ο χρήστης μπορεί να επεξεργαστεί την κριτική.

1.2.3 Οριακά αντικείμενα

1.2.3.1 Boundary <CreateRatingGUI >

«Boundary» CreateRatingGUI
+userId: uuid +outlinedStars: image +filledStars: image +submitButton: Button +commentForm: FormElement +rating: integer[1..5] +ratingLabel: string +RatingGUI(): RatingGUI +submit(userId: uuid, rating: integer[1..5], comment: String): void

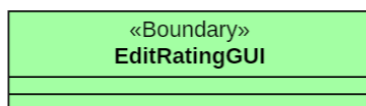
Χαρακτηριστικά κλάσης

- <userId : uuid>: Το βασικό κλειδί κι ευρετήριο της οντότητας User.
- <outlinedStars: image>: Η εικόνα με τα μη-χρωματισμένα αστέρια της κριτικής.
- <filledStars: image>: Η εικόνα με τα γεμισμένα αστέρια της κριτικής.
- <submitButton: button>: Το κουμπί που πατάει ο χρήστης για να υποβάλει την κριτική.
- <commentForm: FormElement>: Η φόρμα στην οποία ο χρήστης γράφει τα σχόλια της κριτικής του.
- <rating: integer[1..5]>: Η βαθμολογία της κριτικής.
- <ratingLabel: string>: Η ετικέτα της σελίδας.

Μέθοδοι κλάσης

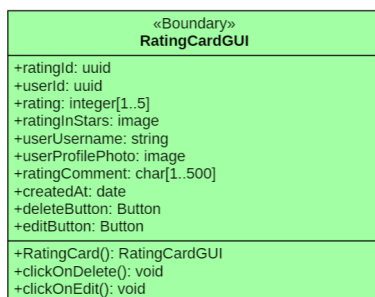
- `<RatingGUI():RatingGUI`
- `<submit(userId : uuid, rating: integer[1..5], comment: String):void>`: Υποβάλλει την κριτική με βάση το βασικό κλειδί του User, την βαθμολογία και τα σχόλια της κριτικής, καλώντας τη συνάρτηση `createRating(rating: Rating): void` του `RatingController`.

1.2.3.2 Boundary <EditRatingGUI >



Σε αυτό το οριακό αντικείμενο μπορείς να επεξεργαστείς τις λεπτομέρειες του rating, φυσικά καλώντας την `updateRating(rating: Rating): void` της `RatingController`. Δεν αναλύεται για εξοικονόμηση χώρου.

1.2.3.3 Boundary <RatingCardGUI >



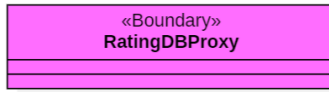
Χαρακτηριστικά κλάσης

- `<ratingId: uuid>`: Το βασικό κλειδί κι ευρετήριο της οντότητας Rating.
- `<userId: uuid>`: Το βασικό κλειδί κι ευρετήριο της οντότητας User.
- `<rating: integer[1..5]>`: Η βαθμολογία της κριτικής.
- `<ratingInStars: image>`: Η αντίστοιχη βαθμολογία σε εικόνα με αστέρια.
- `<username: string>`: Το username του χρήστη.
- `<profilePhoto: image>`: Η εικόνα προφίλ του χρήστη.
- `<comment: char[1..500]>`: Τα σχόλια της κριτικής.
- `<createdAt: date>`: Η ημερομηνία που έγινε η κριτική.
- `<deleteButton: button>`: Το κουμπί για διαγραφή της κριτικής.
- `<editButton: button>`: Το κουμπί για επεξεργασία της κριτικής.

Μέθοδοι κλάσης

- `<RatingCard():RatingCardGUI>`: Η συνάρτηση δόμησης της κλάσης.
- `<clickOnDelete():void>`: Καλεί τη συνάρτηση `deleteRating(rating: Rating): void` του `RatingController`.
- `<clickOnEdit():void>`: Καλεί τη συνάρτηση `editRatingPage(ratingId: uuid): EditRatingGUI` του `RatingController`.

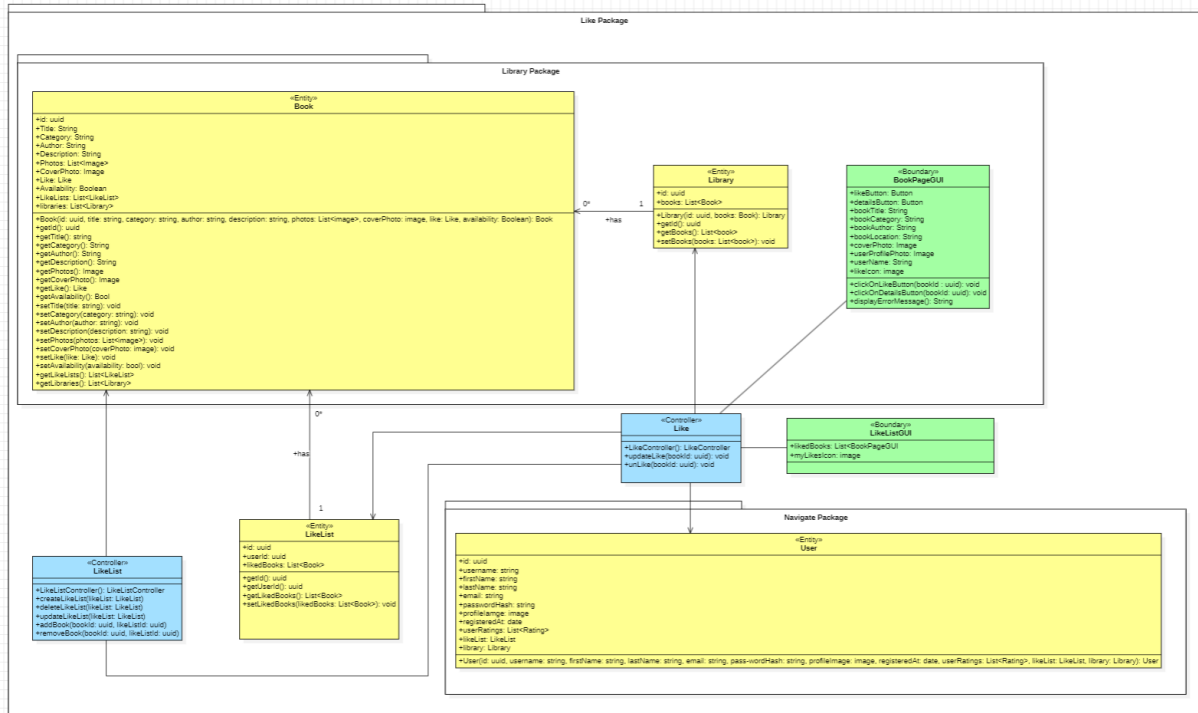
1.2.3.4 Boundary <RatingDBProxy >



Ακολουθεί τη λογική του **BookDBProxy** του **Library Package**.

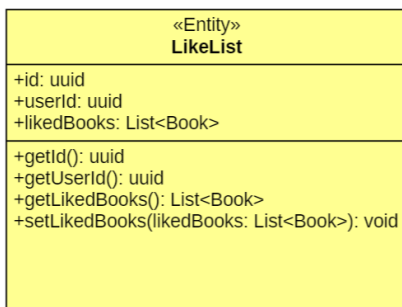
1.3 <Like Package>

Το πακέτο αναλύει τις κλάσεις του **Feature 5: Like Books**.



1.3.1 Οντότητες

1.3.1.1 Entity <LikeList >



Χαρακτηριστικά κλάσης

- <id: **uuid**>: Το βασικό κλειδί κι ευρετήριο της οντότητας.
- <userId: **uuid**>: Το κλειδί του χρήστη στον οποίο αναφέρεται η λίστα.
- <likedBooks: List<**Book**>>: Η λίστα με τα βιβλία στα οποία έχει πατήσει like ο χρήστης.

Μέθοδοι κλάσης

- `<LikeList(id: uuid, userId: uuid, likedBooks: List<Book>): LikeList>`: Η συνάρτηση δόμησης της κλάσης.
- `<getId(): uuid>`: Επιστρέφει το κλειδί της οντότητας.
- `<getUserId(): uuid>`: Επιστρέφει το κλειδί του χρήστη στον οποίο αναφέρεται η λίστα.
- `<getLikedBooks(): List<Book>>`: Επιστρέφει τη λίστα με τα βιβλία που αρέσουν στο χρήστη.
- `<setLikedBooks(likedBooks: List<Book>): void>`: Ανανεώνει τη λίστα βιβλίων που αρέσουν στο χρήστη με τη λίστα που δίνεται ως όρισμα.

1.3.1.2 Entity <Book>

Αναλύθηκε στο [Library Package](#).

1.3.1.2 Entity <User>

Αναλύεται στο [Navigate Package](#).

1.3.1.2 Entity <Library>

Αναλύθηκε στο [Library Package](#).

1.3.2 Ελεγκτές

1.3.2.1 Controller <LikeListController >

«Controller» LikeList
+LikeListController(): LikeListController +createLikeList(likeList: LikeList) +deleteLikeList(likeList: LikeList) +updateLikeList(likeList: LikeList) +addBook(bookId: uuid, likeListId: uuid) +removeBook(bookId: uuid, likeListId: uuid)

Μέθοδοι κλάσης

- `<LikeListController(): LikeListController>`: Η συνάρτηση δόμησης της κλάσης.
- `<createLikeList(likeList: LikeList)>`: Δημιουργεί μια νέα λίστα αγαπημένων βιβλίων στη βάση δεδομένων.
- `<deleteLikeList(likeList: LikeList)>`: Διαγράφει μια λίστα αγαπημένων από τη βάση δεδομένων.
- `<updateLikeList(likeList: LikeList)>`: Ανανεώνει το περιεχόμενο μιας λίστας αγαπημένων στη βάση δεδομένων.
- `<addBook(bookId: uuid, likeListId: uuid)>`: Προσθέτει ένα βιβλίο στη λίστα.
- `<removeBook(bookId: uuid, likeListId: uuid)>`: Αφαιρεί ένα βιβλίο από τη λίστα.

1.3.2.2 Controller <LikeController >

«Controller» Like
+LikeController(): LikeController +updateLike(bookId: uuid): void +unLike(bookId: uuid): void

Μέθοδοι κλάσης

- `<LikeController(): LikeController>`: Η συνάρτηση δόμησης της κλάσης.
- `<updateLike(bookId: uuid): void>`:Κάνει like καλώντας την `addBook(bookId: uuid, likeListId: uuid)` του `LikeListController` να προσθέσει το βιβλίο στη λίστα.
- `<unLike(bookId: uuid): void>`:Κάνει unlike καλώντας την `removeBook(bookId: uuid, likeListId: uuid)` του `LikeListController` για να αφαιρέσει το βιβλίο από τη λίστα.

1.3.3 Οριακά αντικείμενα

1.3.3.1 Boundary <LikeListGUI >

«Boundary» LikeListGUI
+likedBooks: List<BookPageGUI
+myLikesIcon: image

Χαρακτηριστικά κλάσης

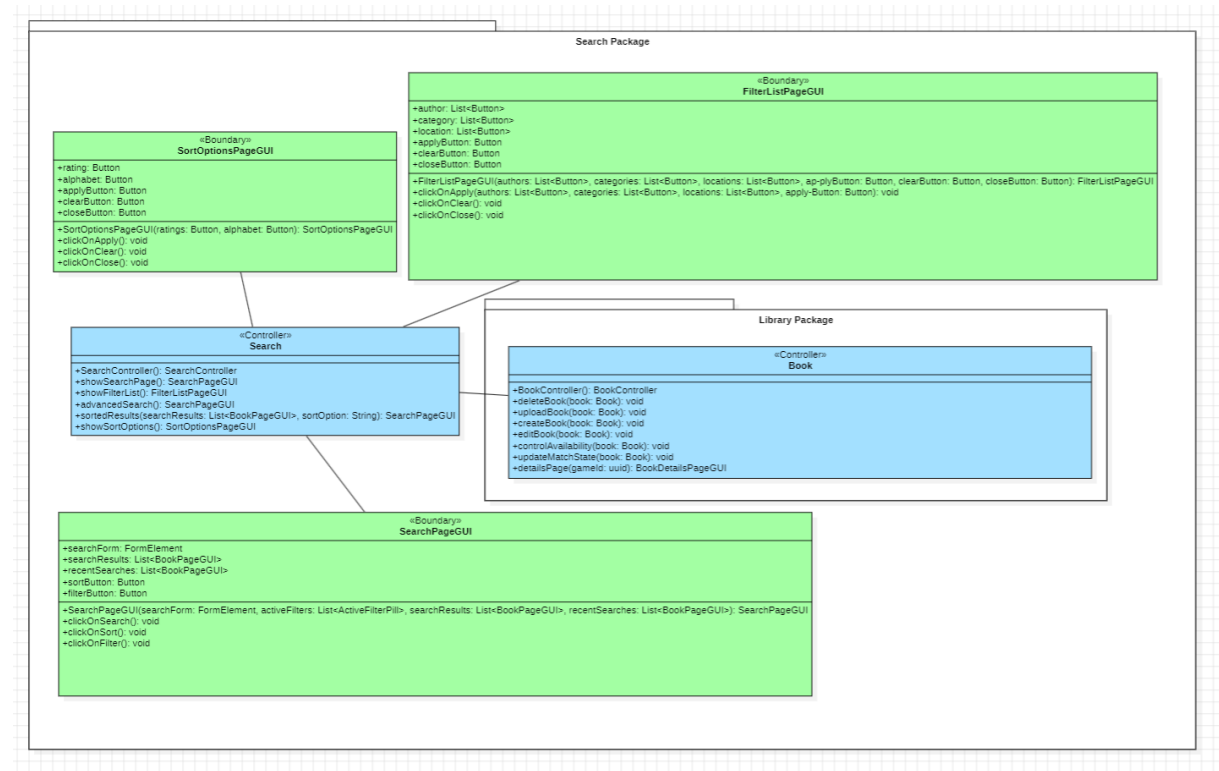
- `<likedBooks: List<BookPageGUI>`: Η λίστα με τα liked βιβλία.
- `<myLikesIcon: image>`: Το εικονίδιο της λίστας.

1.3.3.2 Boundary <BookPageGUI >

Αναλύεται στο [Library Package](#).

1.4 <Search Package>

Το πακέτο αναλύει τις κλάσεις των **Feature 2: Search for books**, **Feature 3: Filter book results**, **Feature 4: Sort Book results**.



1.4.1 Ελεγκτές

1.4.1.1 Controller <SearchController >

«Controller» Search
+SearchController(): SearchController +showSearchPage(): SearchPageGUI +showFilterList(): FilterListPageGUI +advancedSearch(): SearchPageGUI +sortedResults(searchResults: List<BookPageGUI>, sortOption: String): SearchPageGUI +showSortOptions(): SortOptionsPageGUI

Μέθοδοι κλάσης

- <SearchController(): SearchController>: Η συνάρτηση δόμησης της κλάσης.
- <showSearchPage(): SearchPageGUI>: Επιστρέφει το οριακό αντικείμενο της σελίδας αναζήτησης SearchPageGUI.
- <showFilterList(): FilterListPageGUI>: Επιστρέφει το οριακό αντικείμενο της σελίδας της λίστας φίλτρων.
- <advancedSearch():SearchPageGUI>: Επιστρέφει ανανεωμένη SearchPageGUI. Καλεί τη συνάρτηση advancedSearch() του BookDBProxy. Αφού λάβει τα παιχνίδια σύμφωνα με τα φίλτρα που έχει ορίσει ο χρήστης, ανανεώνει τη σελίδα με μια λίστα από το οριακό αντικείμενο BookPageGUI.
- <sortedResults(searchResults: List<BookPageGUI>, sortOption: string): SearchPageGUI>: Παίρνει τη λίστα με τα αποτελέσματα και την επιστρέφει με βάση την ταξινόμηση που έχει επιλεγεί από το χρήστη.
- <showSortOptions(): SortOptionsPageGUI>: Επιστρέφει το οριακό αντικείμενο της σελίδας της λίστας με της επιλογές ταξινόμησης.

1.4.1.2 Controller <BookController >

Αναλύθηκε στο Library Package.

1.4.2 Οριακά αντικείμενα

1.4.2.1 Boundary <FilterListPageGUI >

«Boundary» FilterListPageGUI
+author: List<Button> +category: List<Button> +location: List<Button> +applyButton: Button +clearButton: Button +closeButton: Button
+FilterListPageGUI(authors: List<Button>, categories: List<Button>, locations: List<Button>, applyButton: Button, clearButton: Button, closeButton: Button): FilterListPageGUI +clickOnApply(authors: List<Button>, categories: List<Button>, locations: List<Button>, applyButton: Button): void +clickOnClear(): void +clickOnClose(): void

Χαρακτηριστικά κλάσης

- <author: List<button>>: Λίστα από τα κουμπιά που αντιστοιχούν της συγγραφείς
- <category: List<button>>: Λίστα από τα κουμπιά που αντιστοιχούν της κατηγορίες
- <location: List<button>>: Λίστα από τα κουμπιά που αντιστοιχούν της τοποθεσίες χρηστών

- <applyButton: [button](#)>: Το κουμπί για την εφαρμογή των επιλεγμένων φίλτρων
- <clearButton: [button](#)>: Το κουμπί για τη διαγραφή όλων των ενεργών φίλτρων
- <closeButton: [button](#)>: Το κουμπί για το κλείσιμο της σελίδας και την επιστροφή στην προηγούμενη

Μέθοδοι κλάσης

- <FilterListPageGUI(authors: List<[button](#)>, categories: List<[button](#)>, locations: List<[Button](#)>, applyButton: [button](#), clearButton: [button](#), closeButton: [button](#)): [FilterListPageGUI](#)>: Η συνάρτηση δόμησης της κλάσης.
- <clickOnApply(authors: List<[button](#)>, categories: List<[button](#)>, locations: List<[Button](#)>, applyButton: [button](#)): void> : Καλεί τη συνάρτηση [advancedSearch](#)(bookTitle: [string](#), authors: List<[string](#)>, categories: List<[string](#)>, locations: List<[string](#)>) του [SearchController](#), προκειμένου να εφαρμοστούν τα επιλεγμένα φίλτρα του χρήστη στην αναζήτηση.
- <clickOnClear(): void>: Καλεί την συνάρτηση [advancedSearch](#)(bookTitle: [string](#), authors: List<[string](#)>, categories: List<[string](#)>, locations: List<[string](#)>): [SearchPage](#) του [SearchController](#), προκειμένου να αφαιρεθούν όλα τα ενεργά φίλτρα.
- <clickOnClose(): void>: Καλεί την συνάρτηση [advancedSearch](#)(bookTitle: [string](#), authors: List<[string](#)>, categories: List<[string](#)>, locations: List<[string](#)>): [SearchPage](#) του [SearchController](#), με τις προηγούμενες επιλογές του χρήστη.

1.4.2.2 Boundary <SortOptionsPageGUI >

«Boundary» SortOptionsPageGUI
+rating: Button +alphabet: Button +applyButton: Button +clearButton: Button +closeButton: Button
+SortOptionsPageGUI(ratings: Button, alphabet: Button): SortOptionsPageGUI +clickOnApply(): void +clickOnClear(): void +clickOnClose(): void

Χαρακτηριστικά κλάσης

- <rating: [button](#)>: Κουμπί που αντιστοιχεί σε ταξινόμηση με βάση κριτικές χρηστών.
- <alphabet: [button](#)>: Κουμπί που αντιστοιχεί σε ταξινόμηση με βάση το αλφάβητο.
- <applyButton: [button](#)>: Το κουμπί για την εφαρμογή της επιλεγμένης ταξινόμησης.
- <clearButton: [button](#)>: Το κουμπί για την αφαίρεση της επιλογής ταξινόμησης.
- <closeButton: [button](#)>: Το κουμπί για το κλείσιμο της σελίδας ταξινόμησης.

Μέθοδοι κλάσης

- <SortOptionsPageGUI(ratings: [button](#), alphabet: [button](#)): [SortOptionsPageGUI](#)>: Συνάρτηση δόμησης της κλάσης.
- <clickOnApply():void>: Ταξινομεί τη λίστα με τα οριακά αντικείμενα [BookPageGUI](#). Καλεί την [sortedResults](#)(searchResults: List<[BookPageGUI](#)>, sortOption: [string](#)) του [SearchController](#).
- <clickOnClear():void>: Ίδια με αυτήν που περιγράφεται στο [FilterListPageGUI](#).
- <clickOnClose():void>: Ίδια με αυτήν που περιγράφεται στο [FilterListPageGUI](#).

1.4.2.3 Boundary <SearchPageGUI >

«Boundary» SearchPageGUI
+searchForm: FormElement +searchResults: List<BookPageGUI> +recentSearches: List<BookPageGUI> +sortButton: Button +filterButton: Button
+SearchPageGUI(searchForm: FormElement, activeFilters: List<ActiveFilterPill>, searchResults: List<BookPageGUI>, recentSearches: List<BookPageGUI>): SearchPageGUI +clickOnSearch(): void +clickOnSort(): void +clickOnFilter(): void

Χαρακτηριστικά κλάσης

- <searchForm: [FormElement](#)>: Η φόρμα αναζήτησης.
- <searchResults: List<[BookPageGUI](#)>>: Η λίστα με τα αποτελέσματα της αναζήτησης, ως οριακό αντικείμενο [BookPageGUI](#).
- <recentSearches: List<[BookPageGUI](#)>>: Η λίστα με τις πρόσφατες αναζητήσεις του χρήστη.
- <sortButton: [button](#)>: Το κουμπί για να κάνεις ταξινόμηση.
- <filterButton> [button](#)>: Το κουμπί για να επιλέξεις (ή να αφαιρέσεις φίλτρα).

Μέθοδοι κλάσης

- <[SearchPageGUI](#)(searchForm: [FormElement](#), activeFilters: List<[ActiveFilterPill](#)>, searchResults: List<[BookPageGUI](#)>, recentSearches: List<[BookPageGUI](#)>): [SearchPageGUI](#)>: Η συνάρτηση δόμησης της κλάσης.
- <[clickOnSearch\(\)](#): [void](#)>: Επιλέγει τη φόρμα αναζήτησης, ώστε ο χρήστης να αρχίσει να πληκτρολογεί.
- <[clickOnSort\(\)](#): [void](#)>: Καλεί την [showSortOptions\(\)](#) της [SearchController](#).
- <[clickOnFilter\(\)](#): [void](#)>: Καλεί την [showFilterList\(\)](#) της [SearchController](#).

ΣΗΜΕΙΩΣΗ : Το αντικείμενο [FormElement](#) έχει τα χαρακτηριστικά των

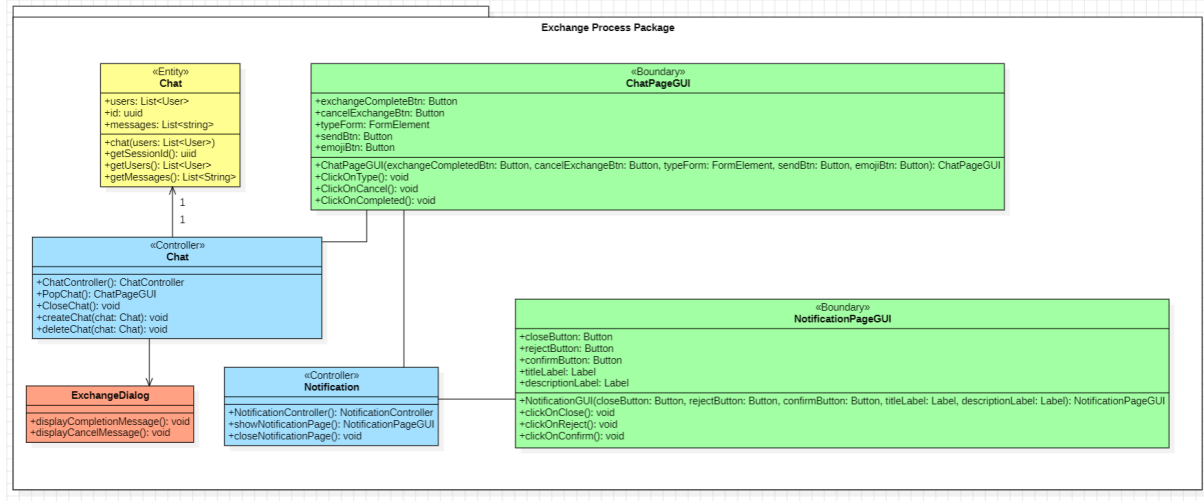
- <input: [string](#)>: Το κείμενο που εισήγε ο χρήστης.
- <entityType: [enum](#)>: Η οντότητα με την οποία σχετίζεται (αν σχετίζεται).
- <entityId: [uuid](#)>: Το id της οντότητας με την οποία σχετίζεται (αν σχετίζεται).
- <label: [string](#)>: Η ετικέτα του.

1.4.2.4 Boundary <BookDBProxy >

Αναλύθηκε στο [Library Package](#).

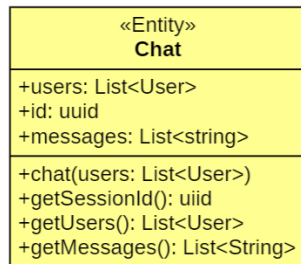
1.5 <Exchange Process Package>

Το πακέτο αναλύει τις κλάσεις των **Feature 6: Get exchange process notifications**, **Feature 8: System directs through exchange process**.



1.5.1 Οντότητες

1.5.1.1 Entity <Chat>



Χαρακτηριστικά κλάσης

- `<users: List <User>>`: Η λίστα με της χρήστες που χρησιμοποιούν το chat.
- `<id: uuid>`: Το id του chat session.
- `<messages: List<string>>`: Η λίστα με τα μηνύματα που έχει το chat session.

Μέθοδοι κλάσης

- `<Chat(users: List<User>): Chat >`: Η συνάρτηση δόμησης της κλάσης
- `<getSessionId(): uuid>`: Επιστρέφει το id του chat session
- `<getUsers(): List<User>>`: Επιστρέφει τους χρήστες που χρησιμοποιούν το chat session.
- `<getMessages(): List<String>>`: Επιστρέφει τη λίστα με τα μηνύματα του chat session.

1.5.2 Ελεγκτές

1.5.2.1 Controller <ChatController>

«Controller» Chat
+ChatController(): ChatController +PopChat(): ChatPageGUI +CloseChat(): void +createChat(chat: Chat): void +deleteChat(chat: Chat): void

Μέθοδοι κλάσης

- <ChatController(): ChatController> : Η συνάρτηση δόμησης της κλάσης
- <PopChat(chatId: uuid): ChatPageGUI> : Επιστρέφει το οριακό αντικείμενο της σελίδας ChatPageGUI.
- <CloseChat(chatId: uuid): void> : Κλείνει το chat.
- <createChat(chat: Chat): void>: Δημιουργεί ένα καινούριο chat στη βάση δεδομένων.
- <deleteChat(chat: Chat): void>: Διαγραφεί το chat από τη βάση δεδομένων.

1.5.2.2 Controller <NotificationController>

«Controller» Notification
+NotificationController(): NotificationController +showNotificationPage(): NotificationPageGUI +closeNotificationPage(): void

Μέθοδοι κλάσης

- <NotificationController(): NotificationController>: Η συνάρτηση δόμησης της κλάσης.
- <showNotificationPage(): NotificationPageGUI> : Επιστρέφει το οριακό αντικείμενο της σελίδας της ειδοποίησης.
- <closeNotificationPage(): void> Κλείνει το παράθυρο της ειδοποίησης κι επιστρέφει στην προηγούμενη σελίδα.

1.5.3 Οριακά αντικείμενα

1.5.3.1 Boundary <ChatPageGUI>

«Boundary» ChatPageGUI
+exchangeCompleteBtn: Button +cancelExchangeBtn: Button +typeForm: FormElement +sendBtn: Button +emojiBtn: Button +ChatPageGUI(exchangeCompletedBtn: Button, cancelExchangeBtn: Button, typeForm: FormElement, sendBtn: Button, emojiBtn: Button): ChatPageGUI +ClickOnType(): void +ClickOnCancel(): void +ClickOnCompleted(): void

Χαρακτηριστικά κλάσης

- <exchangeCompletedButton: **button**>: Κουμπί που δηλώνει ότι ολοκληρώθηκε η συναλλαγή
- <cancelExchangeButton: **button**> : Κουμπί για την ακύρωση της συναλλαγής
- <typeForm: **FormElement**> : Η φόρμα που μπορείς να πληκτρολογήσεις μήνυμα
- <sendBtn: **button**> : Κουμπί για να στείλεις το μήνυμα
- <emojiBtn: **button**>: Κουμπί για να στείλεις emoji

Μέθοδοι κλάσης

- <ChatPageGUI(exchangeCompletedBtn: **button**, cancelExchangeBtn: **button**, typeForm: **FormElement**, sendBtn: **button**, emojiBtn: **button**): **ChatPageGUI**>: Η συνάρτηση δόμησης της κλάσης
- <clickOnType():void>: Επιλέγει τη φόρμα ώστε να πληκτρολογήσει ο χρήστης.
- <clickOnCancel():void>: Επιλέγει το κουμπί ακύρωσης της συναλλαγής.
- <clickOnCompleted():void>: Επιλέγει το κουμπί επιβεβαίωσης της ολοκλήρωσης συναλλαγής.

1.5.3.2 Boundary <NotificationPageGUI>

«Boundary» NotificationPageGUI	
+closeButton: Button +rejectButton: Button +confirmButton: Button +titleLabel: Label +descriptionLabel: Label	
+NotificationGUI(closeButton: Button, rejectButton: Button, confirmButton: Button, titleLabel: Label, descriptionLabel: Label): NotificationPageGUI +clickOnClose(): void +clickOnReject(): void +clickOnConfirm(): void	

Χαρακτηριστικά κλάσης

- <closeButton: **button**>: Το κουμπί που κλείνει την ειδοποίηση
- <rejectButton: **button**>: Το κουμπί που απορρίπτει την συνέχιση της διαδικασίας συναλλαγής
- <confirmButton: **button**>: Το κουμπί που επιβεβαιώνει τη συνέχιση της διαδικασίας
- <titleLabel: **label**>: Η ετικέτα με τον τίτλο της ειδοποίησης
- <descriptionLabel: **label**>: Η ετικέτα με την περιγραφή της ειδοποίησης

Μέθοδοι κλάσης

- <NotificationGUI(closeButton: **button**, rejectButton: **button**, confirmButton: **button**, titleLabel: **label**, descriptionLabel: **label**): **NotificationPageGUI**>: Η συνάρτηση δόμησης της κλάσης.
- <clickOnClose()>: Καλεί τη συνάρτηση **closeNotificationPage()**: **void** του NotificationController.
- <clickOnReject()>: Καλεί πάλι τη συνάρτηση **closeNotificationPage()**: **void** του NotificationController.
- <clickOnConfirm()>: Καλεί τη συνάρτηση **PopChat(chatId: uuid)**: **ChatPageGUI** του **ChatController** και την **createChat(chat: Chat)**: **void**.

1.5.3.3 Boundary <ExchangeDialog>

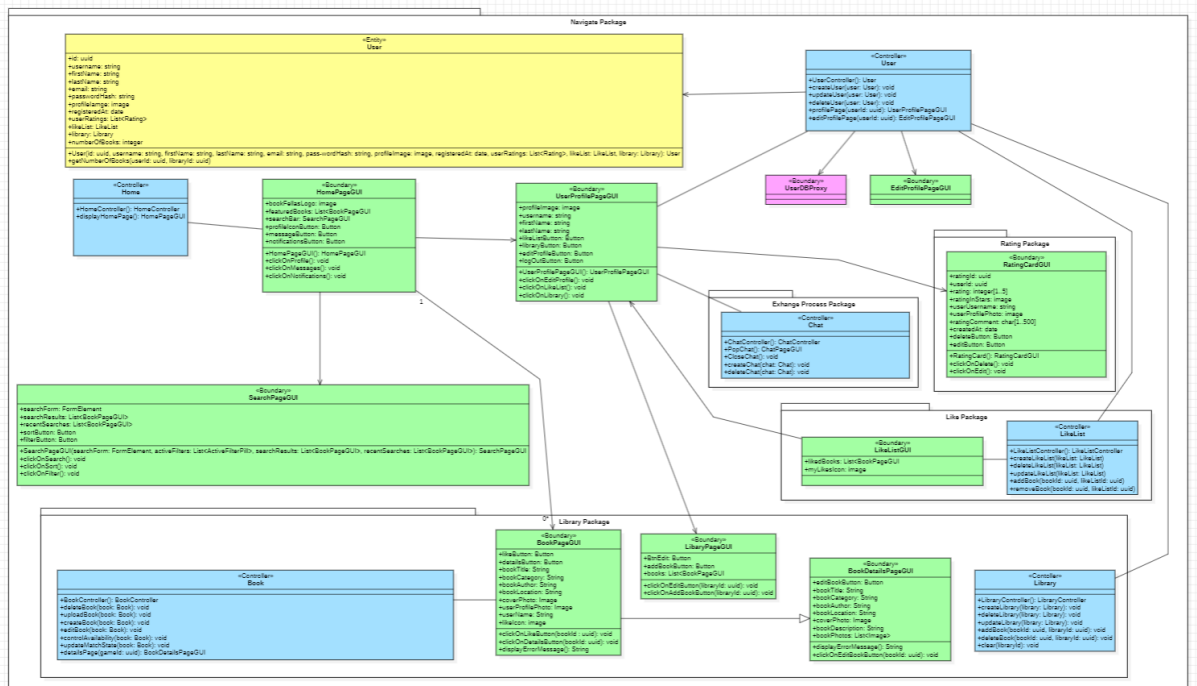


Μέθοδοι κλάσης

- <displayCompletionMessage(): void>: Εμφανίζει το μήνυμα ότι η συναλλαγή ήταν επιτυχής.
- <displayCancelMessage(): void>: Εμφανίζει το μήνυμα ότι η συναλλαγή ματαιώθηκε.

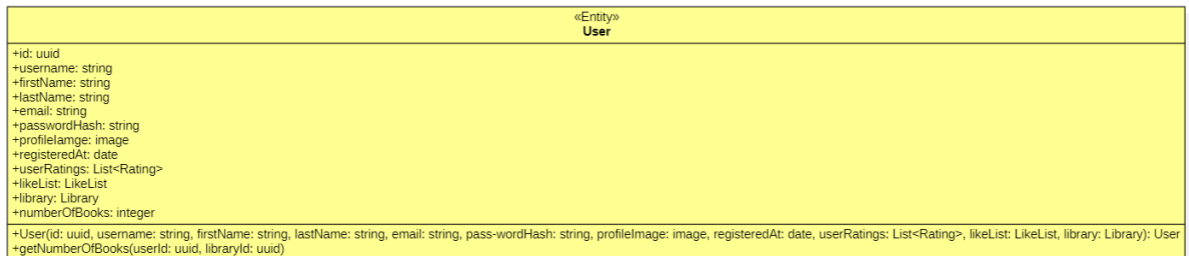
1.6 <Navigate Package>

Το πακέτο αναλύει τις κλάσεις του **Feature 11: Navigate to the application.**



1.6.1 Οντότητες

1.6.1.1 Entity <User>



Χαρακτηριστικά κλάσης

- <id: **uuid**>: Το βασικό κλειδί κι ευρετήριο της οντότητας.
- <username: **string**>: Το username του χρήστη.
- <firstName: **string**>: Το όνομα του χρήστη.

- <lastName: **string**>: Το επώνυμο του χρήστη.
- <email: **string**>: Το email του χρήστη.
- <passwordHash: **string**>: Ο κωδικός του χρήστη.
- <profileImage: **image**> : Η εικόνα προφίλ του χρήστη.
- <registeredAt: **date**> : Η ημερομηνία εγγραφής του χρήστη.
- <userRatings: List<**Rating**>>: Η λίστα με τις κριτικές που έχουν γίνει στον χρήστη.
- <hisRatings: List<**Rating**>>: Η λίστα με τις κριτικές που έχει κάνει ο χρήστης.
- <likeList: **LikeList**>: Η λίστα με τα βιβλία που έχει κάνει like.
- <library: **Library**>: Η προσωπική του βιβλιοθήκη.
- <numberOfBooks: **integer**>: Ο αριθμός των βιβλίων που υπάρχουν στη βιβλιοθήκη του.

Μέθοδοι κλάσης

- <User(id: **uuid**, username: **string**, firstName: **string**, lastName: **string**, email: **string**, passwordHash: **string**, profileImage: **image**, registeredAt: **date**, userRatings: List<**Rating**>, likeList: **LikeList**, library: **Library**): User>: Η συνάρτηση δόμησης της κλάσης.
- <getNumberOfBooks(userId: **uuid**, libraryId: **uuid**>: Επιστρέφει τον αριθμό των βιβλίων που βρίσκονται στην βιβλιοθήκη του χρήστη.

ΣΗΜΕΙΩΣΗ: Οι υπόλοιποι getters και setters παραλείπονται για εξοικονόμηση χώρου.

1.6.2 Ελεγκτές

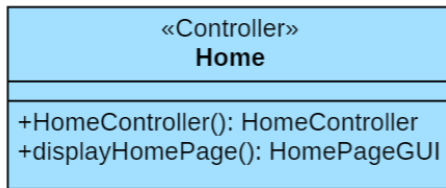
1.6.2.1 Controller <UserController>

«Controller» User
+UserController(): User +createUser(user: User): void +updateUser(user: User): void +deleteUser(user: User): void +profilePage(userId: uuid): UserProfilePageGUI +editProfilePage(userId: uuid): EditProfilePageGUI

Μέθοδοι κλάσης

- <UserController(): **UserController**>: Η συνάρτηση δόμησης της κλάσης.
- <createUser(user: **User**): **void**> : Προσθέτει ένα καινούριο χρήστη στη βάση δεδομένων.
- <updateUser(user: **User**): **void**> : Αλλάζει τα χαρακτηριστικά ενός υπάρχοντος χρήστη στη βάση δεδομένων.
- <deleteUser(user: **User**): **void**> : Διαγράφει ένα χρήστη από τη βάση δεδομένων.
- <profilePage(userId: **uuid**): **UserProfilePageGUI**>: Επιστρέφει το οριακό αντικείμενο της σελίδας του προσωπικού προφίλ του χρήστη.
- <editProfilePage(userId: **uuid**): **EditProfilePageGUI**>: Επιστρέφει το οριακό αντικείμενο της σελίδας επεξεργασίας των προσωπικών στοιχείων του χρήστη.

1.6.2.2 Controller <HomeController>



Μέθοδοι κλάσης

- <HomeController():HomeController>: Η συνάρτηση δόμησης της κλάσης.
- <displayHomePage():HomePageGUI>: Επιστρέφει το οριακό αντικείμενο της αρχικής σελίδας HomePageGUI.

1.6.2.3 Controller <BookController>

Αναλύθηκε στο Library Package.

1.6.2.4 Controller <ChatController>

Αναλύθηκε στο Exchange Process Package.

1.6.2.5 Controller <LibraryController>

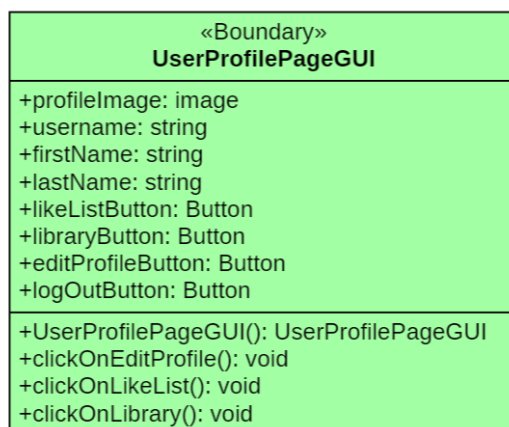
Αναλύθηκε στο Library Package.

1.6.2.6 Controller <LikeListController>

Αναλύθηκε στο Like Package.

1.6.3 Οριακά αντικείμενα

1.6.3.1 Boundary <UserProfilePageGUI>



Χαρακτηριστικά κλάσης

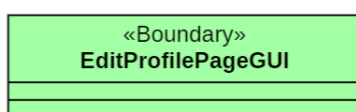
- <profileImage: image>: Η εικόνα προφίλ του χρήστη.
- <username: string>: Το username του χρήστη.
- <firstName: string>: Το όνομα του χρήστη.
- <lastName: string>: Το επώνυμο του χρήστη.
- <likeListButton: Button>: Το κουμπί για να δει τη λίστα με τα βιβλία που έχει κάνει like.
- <libraryButton: Button>: Το κουμπί για να δει τη βιβλιοθήκη του.

- <editProfileButton: **Button**>: Το κουμπί που πατάει ο χρήστης για να κάνει επεξεργασία το προφίλ του.
- <logoutButton: **Button**>: Το κουμπί που πατάει ο χρήστης για να αποσυνδεθεί.

Μέθοδοι κλάσης

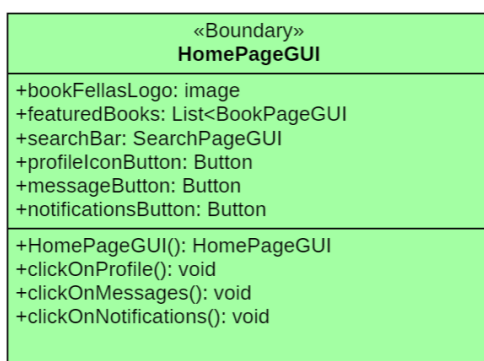
- <UserProfilePageGUI():**UserProfilePageGUI**>: Η συνάρτηση δόμησης της κλάσης.
- <clickOnEditProfile():**void**>: Καλεί τη συνάρτηση **editProfilePage(userId:uuid): EditProfilePageGUI** του **UserController**.
- <clickOnLikeList() : **void**>: Καλεί τη συνάρτηση **showLikeList()** του **LikeListController**.
- <clickOnLibrary() : **void**>: Καλεί τη συνάρτηση **showLibrary()** του **LibraryController**.

1.6.3.2 Boundary <EditProfilePageGUI>



Είναι η σελίδα όπου ο χρήστης μπορεί να επεξεργαστεί τα στοιχεία του. Δεν αναλύεται για εξοικονόμηση χώρου.

1.6.3.3 Boundary <HomePageGUI>



Χαρακτηριστικά κλάσης

- <bookFellasLogo: **image**>: Το logo της ιστοσελίδας.
- <featuredBooks: List<**BookPageGUI**>>: Η λίστα με τα βιβλία που εμφανίζονται στην αρχική.
- <searchBar: **SearchPageGUI**>: Η σελίδα για να κάνεις αναζήτηση.
- <profileIconButton : **Button**>: Το κουμπί για να μεταβείς στο προφίλ.
- <messageButton: **Button**>: Το κουμπί για να μεταβείς στα μηνύματα.
- <notificationsButton : **Button**>: Το κουμπί για να μεταβείς στις ειδοποιήσεις.
-

Μέθοδοι κλάσης

- <HomePageGUI():**HomePageGUI**>: Η συνάρτηση δόμησης της κλάσης.
- <clickOnProfile(): **void**>: Καλεί την συνάρτηση **profilePage(userId: uuid): UserProfilePageGUI** του **UserController**.
- <clickOnMessages(): **void**>: Καλεί τη συνάρτηση του **popChat(chatId: uuid)** του **ChatController**.
- <clickOnNotifications(): **void**>: Καλεί τη συνάρτηση του **showNotificationPage(): NotificationPageGUI** του **NotificationController**.

1.6.3.4 Boundary <LibraryGUI>

Αναλύθηκε στο Library Package.

1.6.3.5 Boundary <RatingCardGUI>

Αναλύθηκε στο Rating Package.

1.6.3.6 Boundary <BookDetailsPageGUI>

Αναλύθηκε στο Library Package.

1.6.3.7 Boundary <LikeListGUI>

Αναλύθηκε στο πακέτο Like Package.

1.6.3.8 Boundary <BookPageGUI>

Αναλύθηκε στο Library Package.

1.6.3.9 Boundary <UserDBProxy>

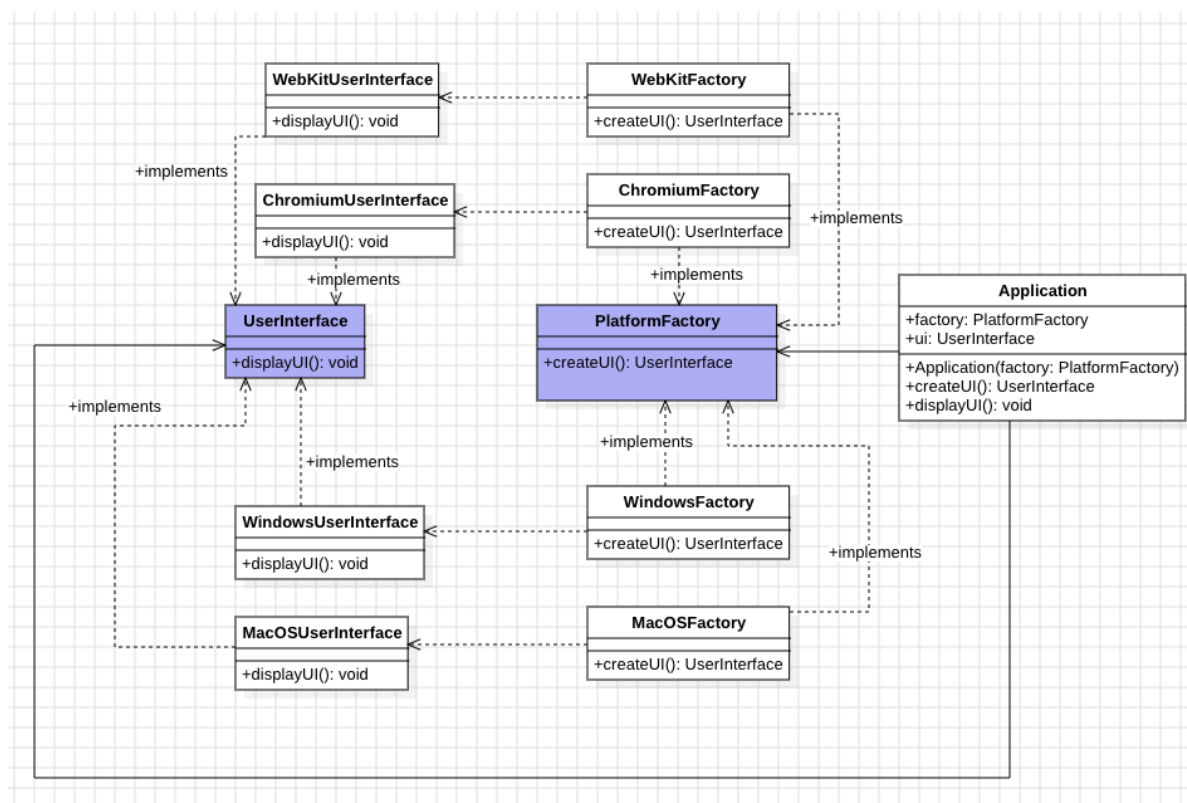
Ακολουθεί παρόμοια λογική με το BookDBProxy του Library Package.

2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

2.1 Πρότυπο Abstract Factory

Πρόκειται για ένα **δημιουργικό πρότυπο**. Με τη χρήση αυτού του προτύπου, το σύστημα μπορεί να υποστηρίξει διάφορα λειτουργικά συστήματα και πλατφόρμες (cross-platform) και επιπλέον πολλούς μοντέρνους web browsers. Τα **WebKitFactory**, **ChromiumFactory**, **MacOSFactory**, **WindowsFactory**, λειτουργούν ως concrete factories και υλοποιούν τη συνάρτηση **createUI()**: **UserInterface** του interface **PlatformFactory**. Ανάλογα με το concrete factory, χρησιμοποιείται το αντίστοιχο product που υλοποιεί το interface **UserInterface** για την κάθε πλατφόρμα. Η πλατφόρμα, εμφανίζει το αντίστοιχο user interface στον τελικό χρήστη μέσω της συνάρτησης **displayUI():void**. Η κλάση **Application** είναι υπεύθυνη για την εκκίνηση και την παραμετροποίηση του περιβάλλοντος του συστήματος. Τα factories **WebKitFactory**, **ChromiumFactory** αναφέρονται σε browser engines έτσι ώστε να καλύπτουν μεγάλη ποικιλία από browsers (WebKit: Safari, Orion, κλπ - Chromium: Chrome, Firefox, Opera, κλπ).

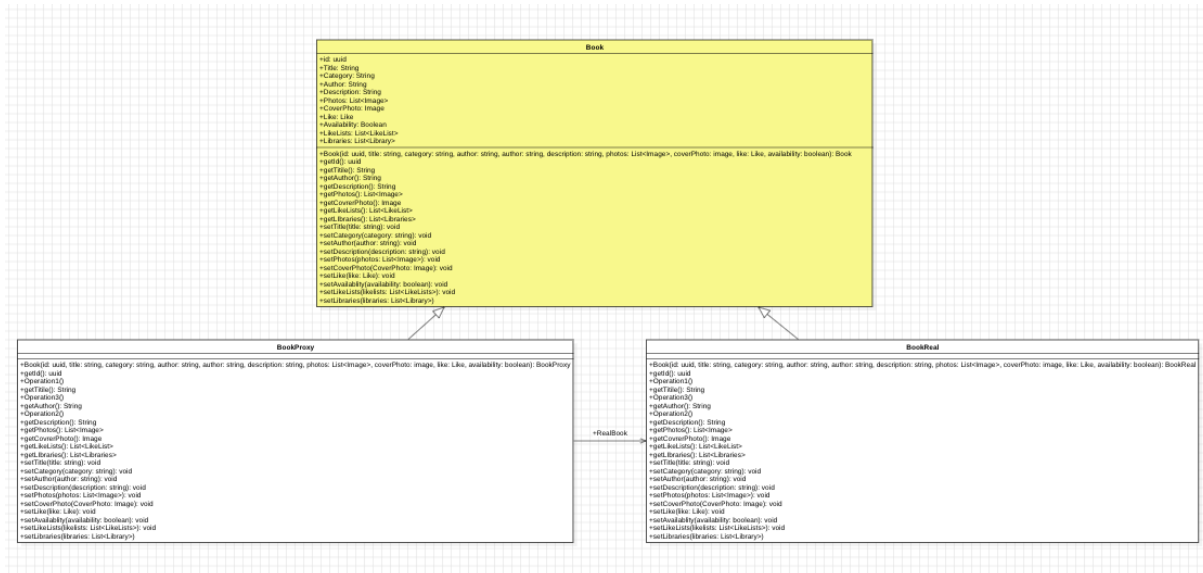
ΣΗΜΕΙΩΣΗ: Το συγκεκριμένο πρότυπο καλύπτει μια πολύ σημαντική ΜΛΑ (να είναι δηλαδή το σύστημα cross-platform), αν και τετριμμένη, γι' αυτό και δεν αναφέρεται στο πρώτο παραδοτέο.



2.2 Πρότυπο Proxy

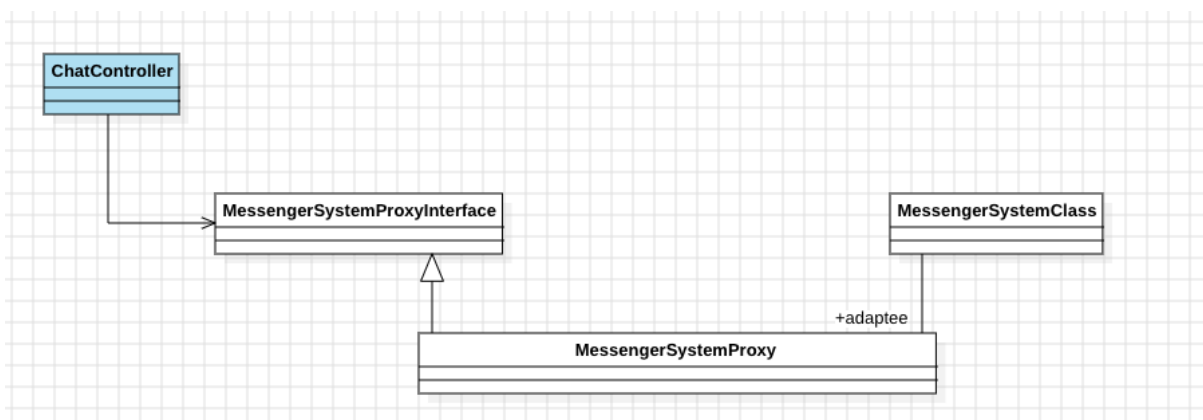
Πρόκειται για ένα **δομικό πρότυπο**. Το πρότυπο proxy χρησιμοποιείται για την κάλυψη της ΜΛΑ-3 του συστήματος Bookfellas, που αφορά την προστασία των προσωπικών δεδομένων των χρηστών (κατά GDPR), όπως επίσης και την ασφάλεια του συστήματος. Αυτά επιτυγχάνονται διότι το πρότυπο proxy παρέχει έλεγχο πρόσβασης στα αντικείμενα ανάλογα με την ιδιότητα του χρήστη. Για παράδειγμα ένας χρήστης πρέπει να μπορεί να επεξεργαστεί τις πληροφορίες ενός Book entity μόνο αν είναι δικό του,

όπως επίσης ένας χρήστης πρέπει να μπορεί να δει μόνο την δική του Like List (λίστα με likes που έχει κάνει). Παρακάτω φαίνεται ένα διάγραμμα κλάσεων με την εφαρμογή του proxy pattern στην κλάση Book του συστήματος.



2.3 Πρότυπο Adapter

Πρόκειται για ένα **δομικό πρότυπο**. Στο σύστημά μας, το πρότυπο Adapter χρησιμοποιείται για την συνεργασία των διεπαφών του ίδιου με το εξωτερικό σύστημα Messenger (της Facebook). Με το πρότυπο αυτό επιτυγχάνεται η επικοινωνία με ένα ήδη εγκατεστημένο αντικείμενο (έτσι έχουμε κι επεκτάσιμο και συντηρήσιμο σύστημα). Το σύστημά μας μπορεί να κάνει χρήση της λειτουργικότητας του Messenger, που είναι επί της ουσίας μια υπηρεσία ανταλλαγής άμεσων μηνυμάτων και βοηθά τους χρήστες του BookFellas να επικοινωνούν για τις ανταλλαγές βιβλίων. Η κλάση **MessengerSystemClass**, αντιπροσωπεύει το API του εξωτερικού συστήματος, τη λειτουργικότητα του οποίου θέλουμε να ενσωματώσουμε. Επιπλέον, με τη χρήση του προτύπου αυτού, αν μεταβληθεί το εξωτερικό σύστημα (Messenger), είναι δυνατή η επαναχρησιμοποίησή του, απλά μεταβάλλοντας αναλόγως την concrete κλάση **MessengerSystemProxy**, χωρίς να απαιτείται κάποια άλλη τροποποίηση από την πλευρά του συστήματός μας.



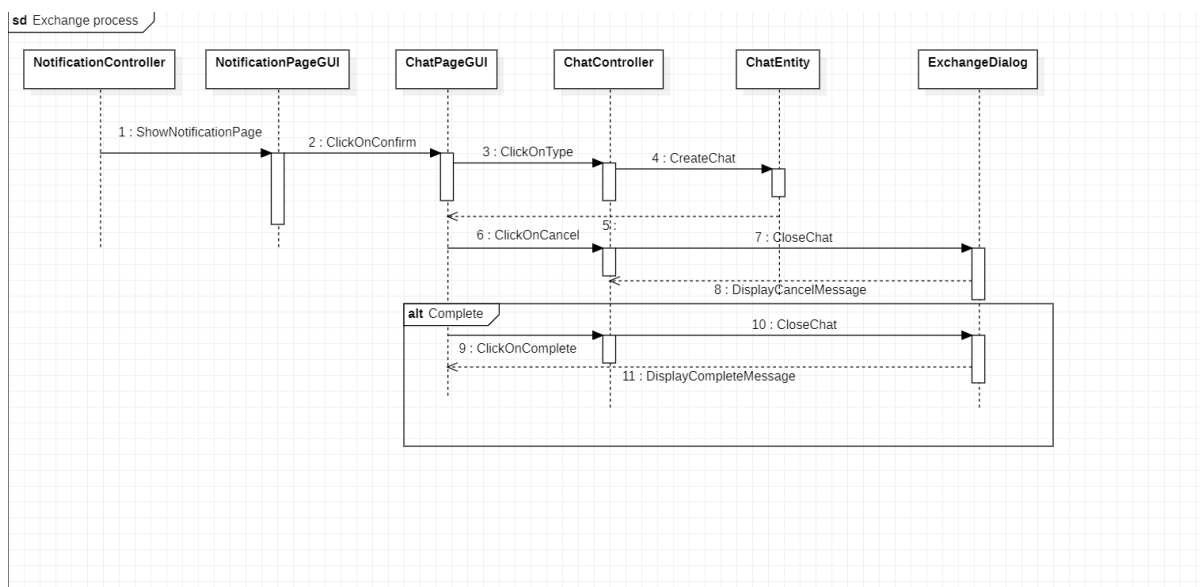
3 Δυναμική Μοντελοποίηση

3.1 Exchange Process Package

3.1.1. Αφήγηση Σεναρίου

Δεδομένου ότι υπάρχει match ανάμεσα σε δύο χρήστες εμφανίζεται η σελίδα των ειδοποιήσεων (**NotificationPageGUI**) ζητώντας επιβεβαίωση από τους χρήστες να μεταβούν στο chat μέσω του πατήματος του κουμπιού confirm (**ClickOnConfirm**). Στη συνέχεια, δημιουργείται η οντότητα του **Chat**, όταν ο χρήστης επιλέγει να πληκτρολογήσει στη σελίδα του chat. Σε περίπτωση που ο χρήστης επιθυμεί να ακυρώσει τη διαδικασία ανταλλαγής πατά το κουμπί cancel και ο controller του chat δίνει εντολή για κλείσιμο του chat (**CloseChat**). Τέλος, μέσω του boundary αντικειμένου παρατίθεται ένα μήνυμα που ενημερώνει για την ακύρωση (**DisplayCancelMessage**). Αντίστοιχα, όταν ολοκληρωθεί μία ανταλλαγή ο χρήστης πατά το κουμπί ολοκλήρωσης και ο controller του chat δίνει εντολή για κλείσιμο του chat (**CloseChat**). Τέλος, μέσω του boundary αντικειμένου παρατίθεται ένα μήνυμα που ενημερώνει για την ολοκλήρωση (**DisplayCompleteMessage**).

3.1.2 Διάγραμμα ακολουθιών



3.2 Library Package

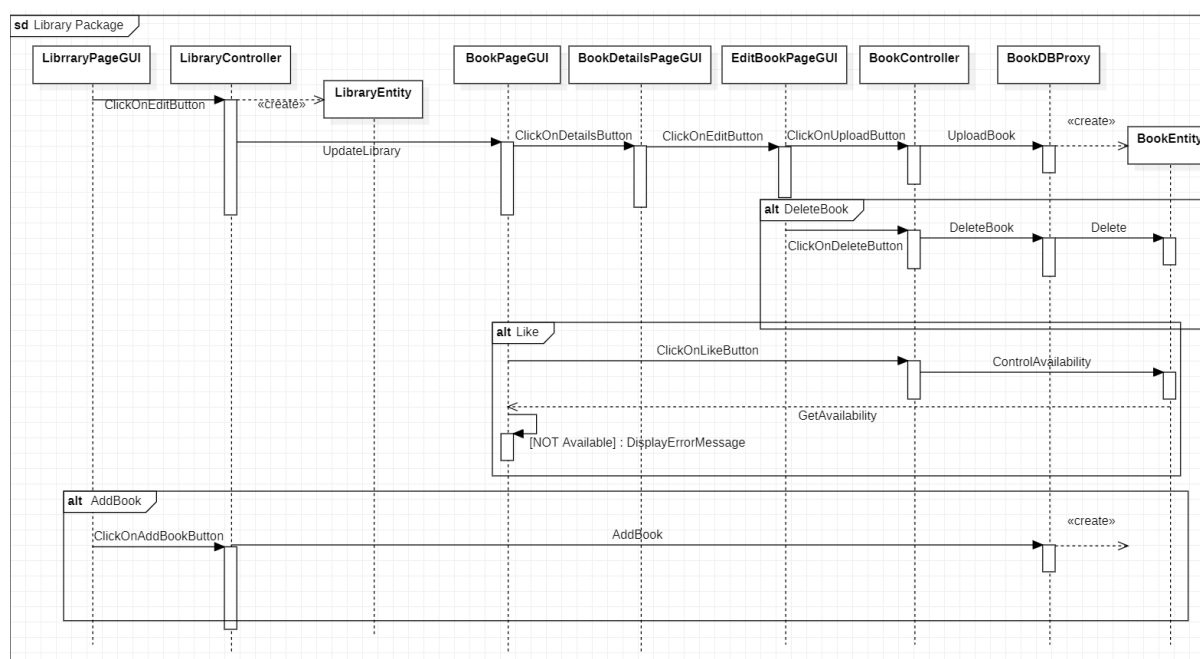
3.2.1 Αφήγηση Σεναρίου

Δεδομένου ότι ο χρήστης βρίσκεται στη σελίδα της βιβλιοθήκης πατώντας το κουμπί edit έχει τη δυνατότητα να τροποποιήσει τη βιβλιοθήκη του. Γίνεται μετάβαση στη σελίδα του βιβλίου (**BookPageGUI**) και πατώντας το κουμπί details εμφανίζεται η σελίδα με τις λεπτομέρειες του βιβλίου (**BookDetailsPageGUI**). Έπειτα, με το πάτημα του κουμπιού edit καλείται ο controller του βιβλίου

και μέσω της συνάρτησης του **uploadBook** η κλάση **BookDBProxy**, που επικοινωνεί με τη βάση, δημιουργεί την ανανεωμένη οντότητα του βιβλίου, μέσω της **update**. Στην περίπτωση που ο χρήστης επιθυμεί να πατήσει like σε ένα βιβλίο ενώ βρίσκεται στη σελίδα του (**BookPageGUI**) πατά

το αντίστοιχο κουμπί. Τότε, ο controller του βιβλίου ελέγχει τη διαθεσιμότητα του μέσω της συνάρτησης `ControlAvailability` και η οντότητα του βιβλίου επιστρέφει την αντίστοιχη τιμή. Αν το βιβλίο είναι μη διαθέσιμο παρατίθεται ένα μήνυμα λάθους στη σελίδα του βιβλίου μέσω της συνάρτησης του `DisplayErrorMessage`. Προκειμένου ο χρήστης να προβεί στο ανέβασμα ενός βιβλίου στη βιβλιοθήκη του πατά το κουμπί add που βρίσκεται στη σελίδα της βιβλιοθήκης (`LibraryPageGUI`). Στη συνέχεια, ο controller της βιβλιοθήκης καλεί την κλάση `BookDBProxy` μέσω της συνάρτησης `addBook` και εκείνη δημιουργεί την οντότητα του βιβλίου, μέσω της `create`. Τέλος, για την διαγραφή ενός βιβλίου γίνεται πάτημα του κουμπιού delete που βρίσκεται στη σελίδα επεξεργασίας του βιβλίου (`EditBookPageGUI`). Τότε καλείται ο controller του βιβλίου, ο οποίος καλεί την `BookDBProxy` μέσω της συνάρτησης `DeleteBook` και εκείνη μέσω της συνάρτησης `delete` προβαίνει στη διαγραφή του βιβλίου από τη βάση δεδομένων.

3.2.2 Διάγραμμα ακολουθιών



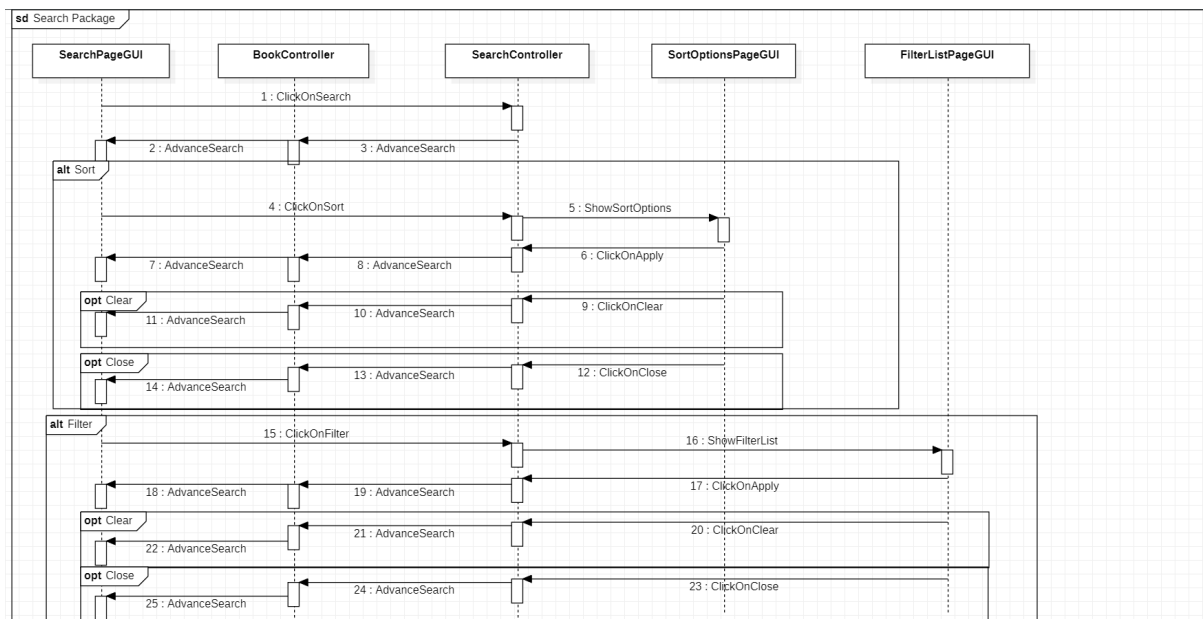
3.3 Search Package

3.3.1. Αφήγηση Σεναρίου

Δεδομένου ότι ο χρήστης βρίσκεται στη σελίδα αναζήτησης πατώντας το κουμπί αναζήτησης επιστρέφονται τα κατάλληλα αποτελέσματα μέσω του `SearchController`. Γίνεται χρήση της συνάρτησης `advancedSearch` του `SearchController` και `BookController`. Στην περίπτωση που ο χρήστης επιθυμεί να φιλτράρει τα αποτελέσματα που εμφανίζονται στην οθόνη του έχει τη δυνατότητα να πατήσει το κουμπί των φίλτρων. Τότε ο `SearchController` μέσω της `ShowFilterList` εμφανίζει την `FilterListPageGUI` με τα παρεχόμενα φίλτρα. Έχοντας επιλέξει τα επιθυμητά φίλτρα μέσω της `ClickOnApply` ο `SearchController` καλεί ξανά την `advancedSearch` και εμφανίζονται στην σελίδα αναζήτησης (`SearchPageGUI`) τα αντίστοιχα αποτελέσματα. Είναι διαθέσιμη η επιλογή εκκαθάρισης φίλτρων και η έξοδος από τα φίλτρα. Πατώντας το clear ή close, αντίστοιχα, ο `SearchController` καλεί ξανά την `advancedSearch` και εμφανίζεται η ανανεωμένη σελίδα αναζήτησης. Στην περίπτωση που ο χρήστης επιθυμεί να ταξινομήσει τα αποτελέσματα που εμφανίζονται στην οθόνη του έχει τη δυνατότητα να πατήσει το κουμπί της ταξινόμησης. Τότε ο `SearchController` μέσω της `ShowSortOptions` εμφανίζει την `SortOptionsPageGUI` με τις επιλογές ταξινόμησης. Έχοντας

επιλέξει την επιθυμητή ταξινόμηση μέσω της [ClickOnApply](#) ο Search Controller καλεί ξανά την [advancedSearch](#) και εμφανίζονται στην σελίδα αναζήτησης ([SearchPageGUI](#)) τα αντίστοιχα αποτελέσματα. Είναι διαθέσιμη η επιλογή εκκαθάρισης ταξινόμησης και η έξοδος από την επιλεγμένη ταξινόμηση, πατώντας το clear ή close, αντιστοίχως.

3.3.2. Διάγραμμα ακολουθιών



Παράρτημα Ι – Γλωσσάριο

ΛΑ - Λειτουργική Απαίτηση

ΟΑ - Ομάδα Εργασίας

ΜΛΑ – Μη Λειτουργική Απαίτηση