



# CAPSTONE PROJECT REPORT

## Report 4 – Software Design Document

– Hanoi, June 29, 2021 –

# Table of Contents

I. Project Report	6
1. Status Report	6
2. Team Involvements	6
3. Issues/Suggestions	6
II. Software Design Document	7
1. Overall Description	7
1.1 Assumptions	7
1.2 Design Constraints	7
1.3 Technology Suggestion	7
2. System Architecture Design	8
2.1 Overall Architecture	8
2.1.1 Diagram	8
2.2.2 Component Explanation	9
2.2 System Architecture	12
2.2.1 SAFO Front-end Architecture	12
2.2.2 SAFO Back-end Architecture	13
2.3 Package Diagram	14
2.3.1 SAFO Package Diagram	14
2.3.2 SAFO Front-end	15
2.3.2.1 Diagram	15
2.3.2.2 Package Explanation	16
2.3.3 SAFO Back-end	17
2.3.3.1 Diagram	17
2.3.3.2 Package Explanation	18
3. System Detailed Design	19
3.1 Home	19
3.1.1 Log In	19
3.1.2 Sign Up	20
3.1.3 Log out	22
3.2 User Profile	25
3.2.1 Reset Password	25

3.2.2 Change Password	27
3.2.3 Create Card	28
3.2.4 View Card	30
3.2.5 Update Card	32
3.2.6 Delete Card	34
3.2.7 Scan QR Code	36
3.2.8 Enter Share Code	36
3.2.9 Share Card	36
3.2.10 View List	37
3.2.11 Update List	38
3.2.12 Delete List	40
3.2.13 Search List	42
3.2.14 Create Group	44
3.2.15 View Group	46
3.2.16 Update Group	48
3.2.17 Delete Group	50
3.2.18 Search Group	52
3.2.19 Create Information Personal	54
3.2.20 View Information Personal	56
3.2.21 Update Information Personal	58
3.2.22 Delete Information Personal	60
3.2.23 View History	62
3.2.24 Send Report	62
3.2.25 Generate Email Signature	64
3.2.26 Contact Safo	66
3.2.27 Change Language	68
3.2.28 Upgrade Premium	69
3.2.29 Buy Theme	71
3.3 Admin Management	73
3.3.1 View Dashboard	73
3.3.2 View History	75
3.3.3 Create Template Card	77

3.3.4 View Template Card	79
3.3.5 Update Template Card	81
3.3.6 Search Template Card	83
3.3.7 Ban User	85
3.3.8 Unban User	87
3.3.9 Search User	89
3.3.10 View Report	91
3.3.11 Update Report	92
3.3.12 Search Report	94
3.3.13 View Transaction	96
3.3.14 Generate Report Chart	97
3.3.15 Create Discount	99
3.3.16 View Discount	101
3.3.17 Update Discount	103
3.3.18 Create Banner	105
3.3.19 View Banner	107
3.3.20 Update Banner	109
3.3.21 Create Premium	111
3.3.22 View Premium	113
3.3.23 Update Premium	115
<b>4. Class Specifications</b>	<b>117</b>
<b>4.1 SAFO Back-end</b>	<b>117</b>
<b>4.1.1 User Controller</b>	<b>117</b>
<b>4.1.2 Admin Controller</b>	<b>120</b>
<b>4.1.3 Common Controller</b>	<b>122</b>
<b>4.1.4 Service</b>	<b>123</b>
<b>4.2 SAFO Front-end</b>	<b>128</b>
<b>4.2.1 SAFO User</b>	<b>128</b>
<b>4.2.2 SAFO Admin</b>	<b>130</b>
<b>5. Data &amp; Database Design</b>	<b>132</b>
<b>5.1 Database Design</b>	<b>132</b>
<b>5.1.1. Account</b>	<b>133</b>

5.1.2. AccountGroup	133
5.1.3. AccountReport	133
5.1.4. AdminLog	134
5.1.5. ApplyingDiscount	134
5.1.6. Banner	134
5.1.7. CardBox	135
5.1.8. Card Group	135
5.1.9. Card Information	135
5.1.10. DataTagConfig	136

# I. Project Report

## 1. Status Report

#	Work Item	Status	Notes (Work Item in Details)
1	Overall Description	Completed	
2	System Architecture Design	Completed	
3	System Detailed Design	Completed	
4	Object Specifications	Completed	
5	Data & Database Design	Completed	

## 2. Team Involvements

#	Task	Member	Notes (Task Details, etc.)
1	Overall Description	TuND	
2	System Architecture Design	TuND	
3	System Detailed Design	TuND	
4	Object Specifications	TuND	
5	Data & Database Design	TuND, AnhTHP, DucNM	

## 3. Issues/Suggestions

#	Issue	Status	Notes (Solution, Suggestion, etc.)
1		Pending	
2		In Progress	
3		Completed	

## **II. Software Design Document**

### **1. Overall Description**

#### **1.1 Assumptions**

This system is designed based on the following assumptions:

- Window 10
- NodeJS v14.17.0
- MySQL
- Google Chrome

#### **1.2 Design Constraints**

This system should comply with the following items:

- It has 2 websites at the client-side to serve two types of actors: normal users and admin. They can use a web browser and application to send requests and connect to the front-end application.
- End-user's Environment: Windows.
- Database using MySQL.
- Support languages: Vietnamese, English.
- The web application must be responsive and snappy.
- The user must have a stable connection to the internet.

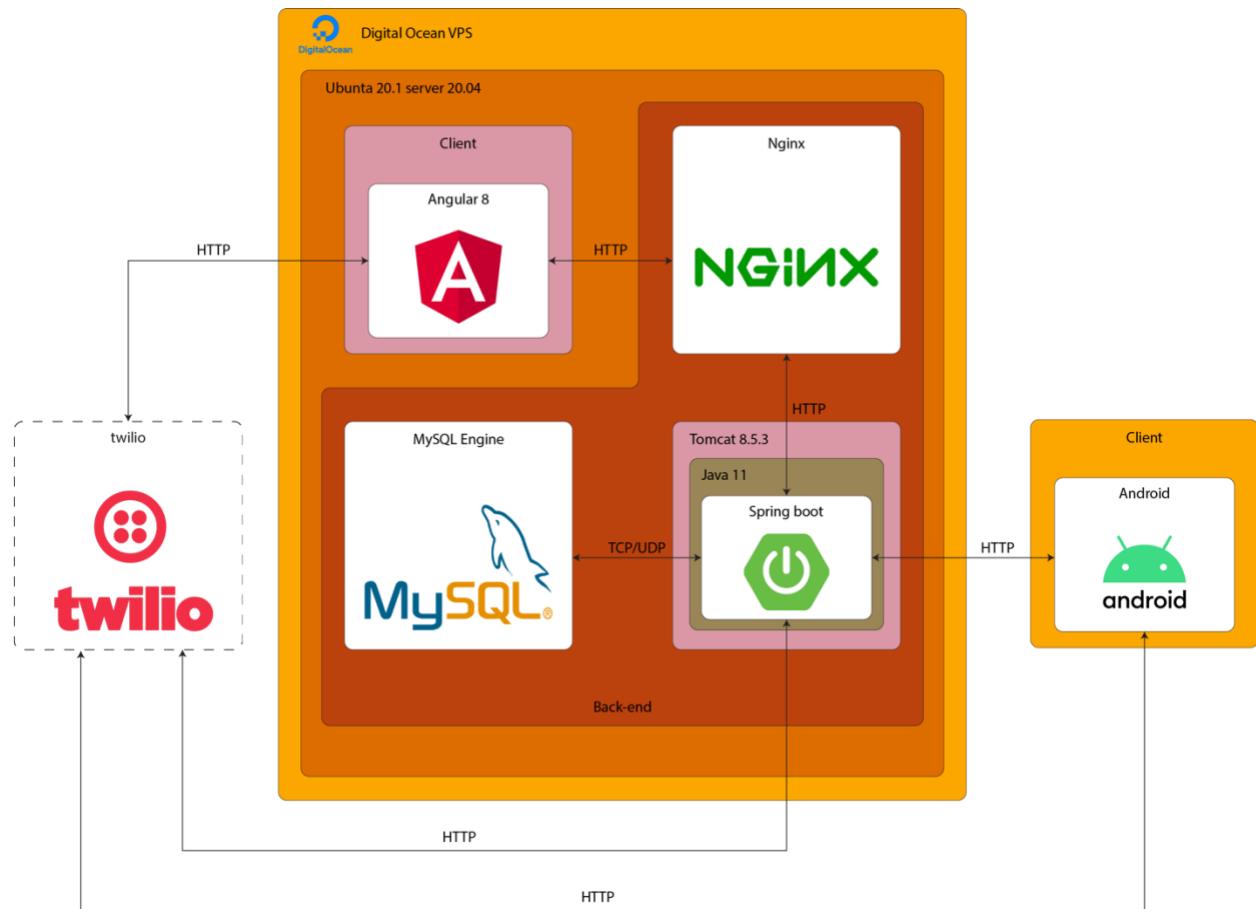
#### **1.3 Technology Suggestion**

- Database: MySQL
- Web Application: Nginx, Tomcat 8.5.3
- Server: VPS, Local server

## 2. System Architecture Design

### 2.1 Overall Architecture

#### 2.1.1 Diagram



## **2.2.2 Component Explanation**

### **2.2.2.1 Angular**



Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps. With Angular, you're taking advantage of a platform that can scale from single-developer projects to enterprise-level applications. Angular is designed to make updating as easy as possible, so you can take advantage of the latest developments with a minimum of effort. Best of all, the Angular ecosystem consists of a diverse group of over 1.7 million developers, library authors, and content creators.

### **2.2.2.2 MySQL**



MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

### **2.2.2.3 Spring Framework**



The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise Javabeans (EJB) model. The Spring Framework is open source.

### **2.2.2.4 Nginx**



Nginx, stylized as NGINX, Nginx, or Nginx, is a web server that can also be used as a reverse proxy, load balancer, mail proxy, and HTTP cache. The software was created by Igor Sysoev and publicly released in 2004. Nginx is free and open-source software, released under the terms of the 2-clause BSD license.

## **2.2.2.5 twilio**



**twilio**

Twilio is an American cloud communications platform as a service company based in San Francisco, California. Twilio allows software developers to programmatically make and receive phone calls, send and receive text messages, and perform other communication functions using its web service APIs.

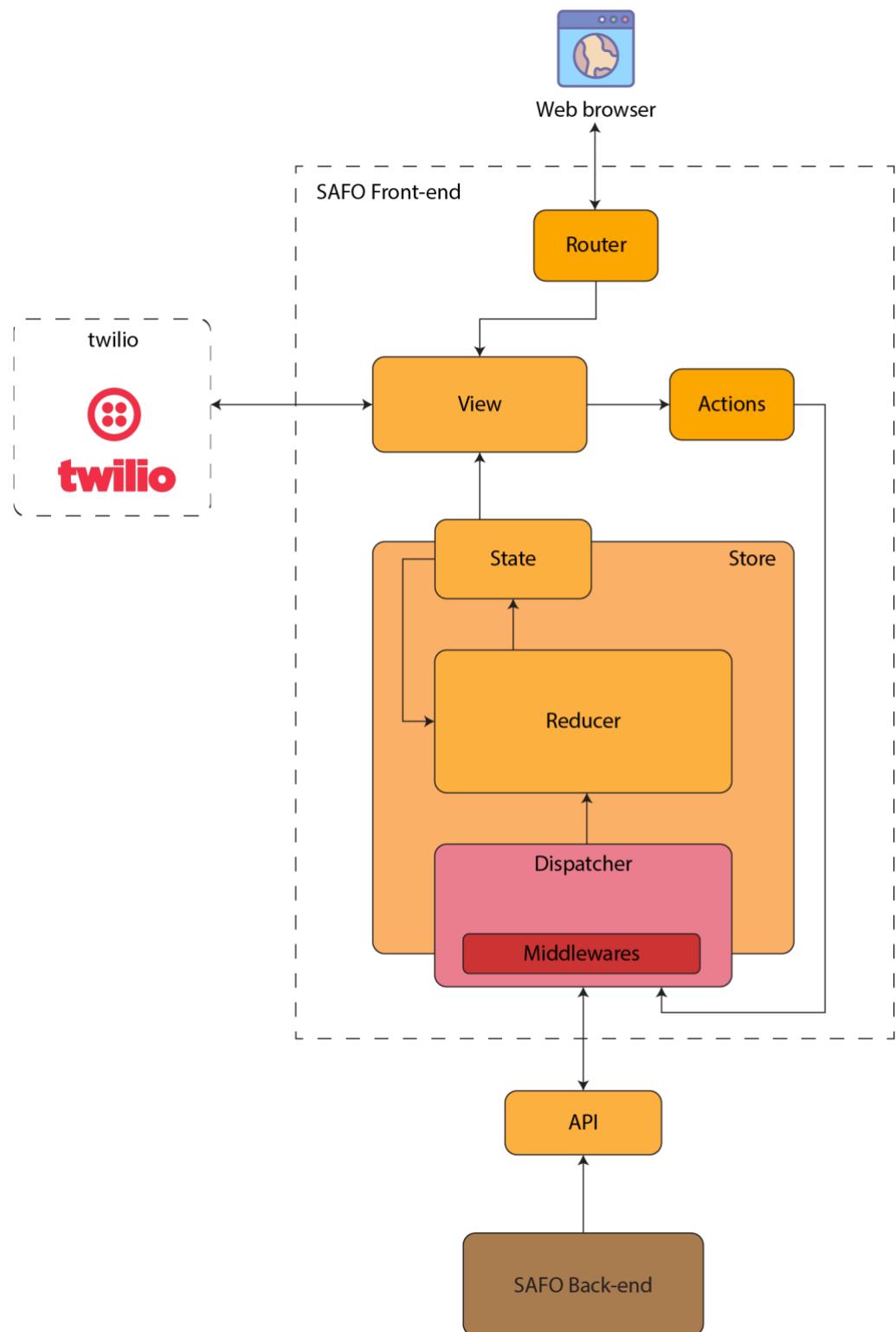
## **2.2.2.6 Android**



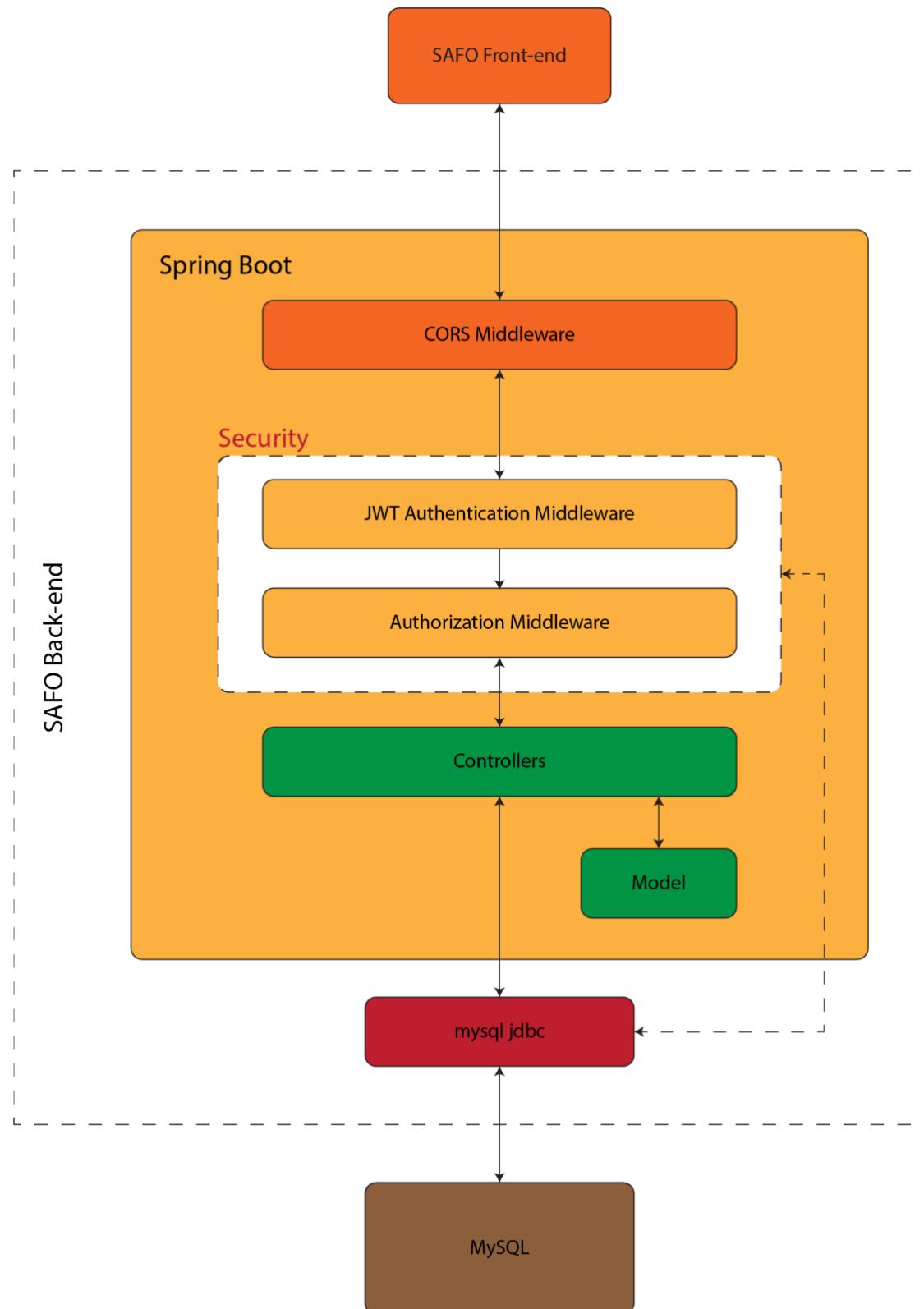
Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It was unveiled in November 2007, with the first commercial Android device, the HTC Dream, being launched in September 2008.

## 2.2 System Architecture

### 2.2.1 SAFO Front-end Architecture

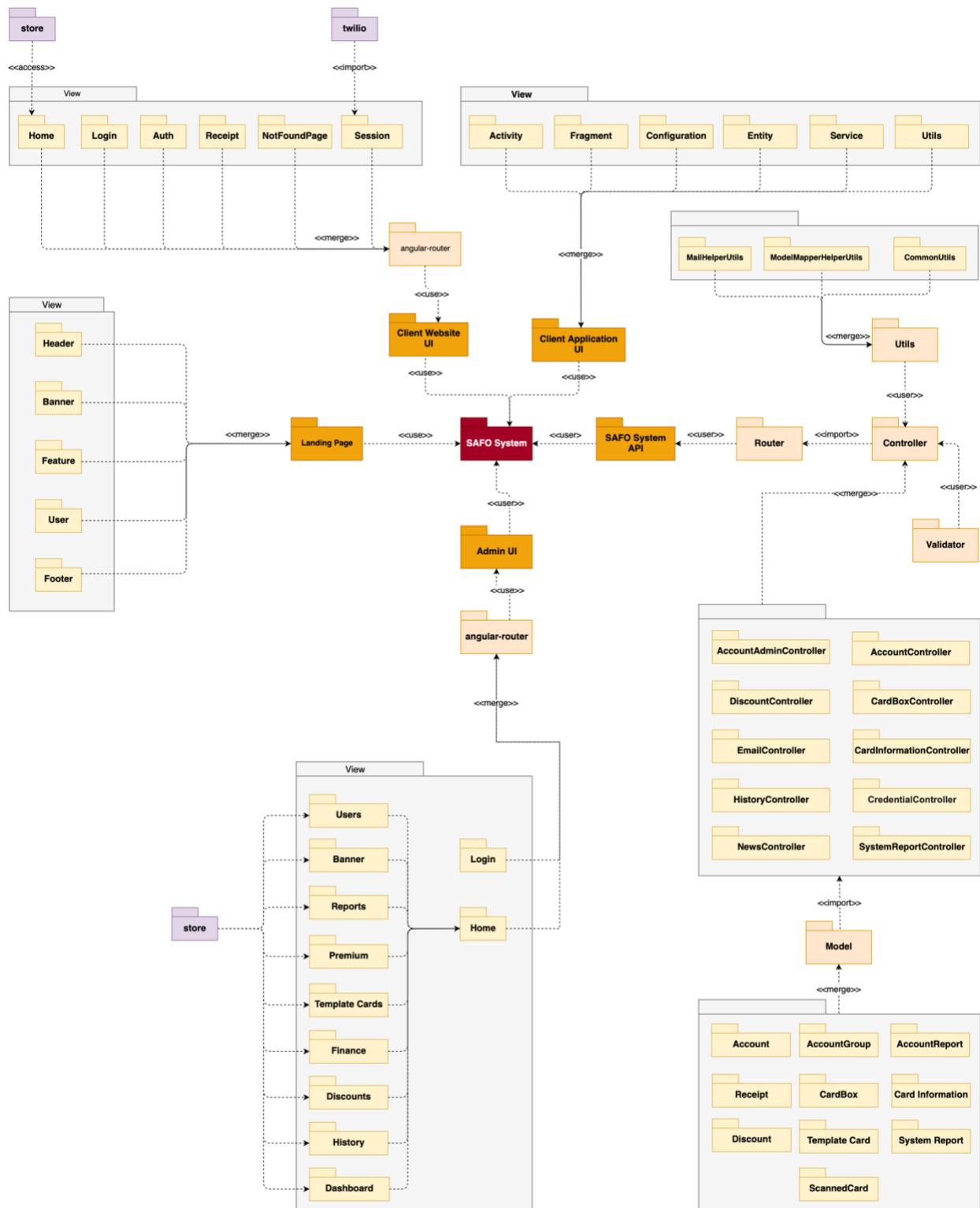


## 2.2.2 SAFO Back-end Architecture



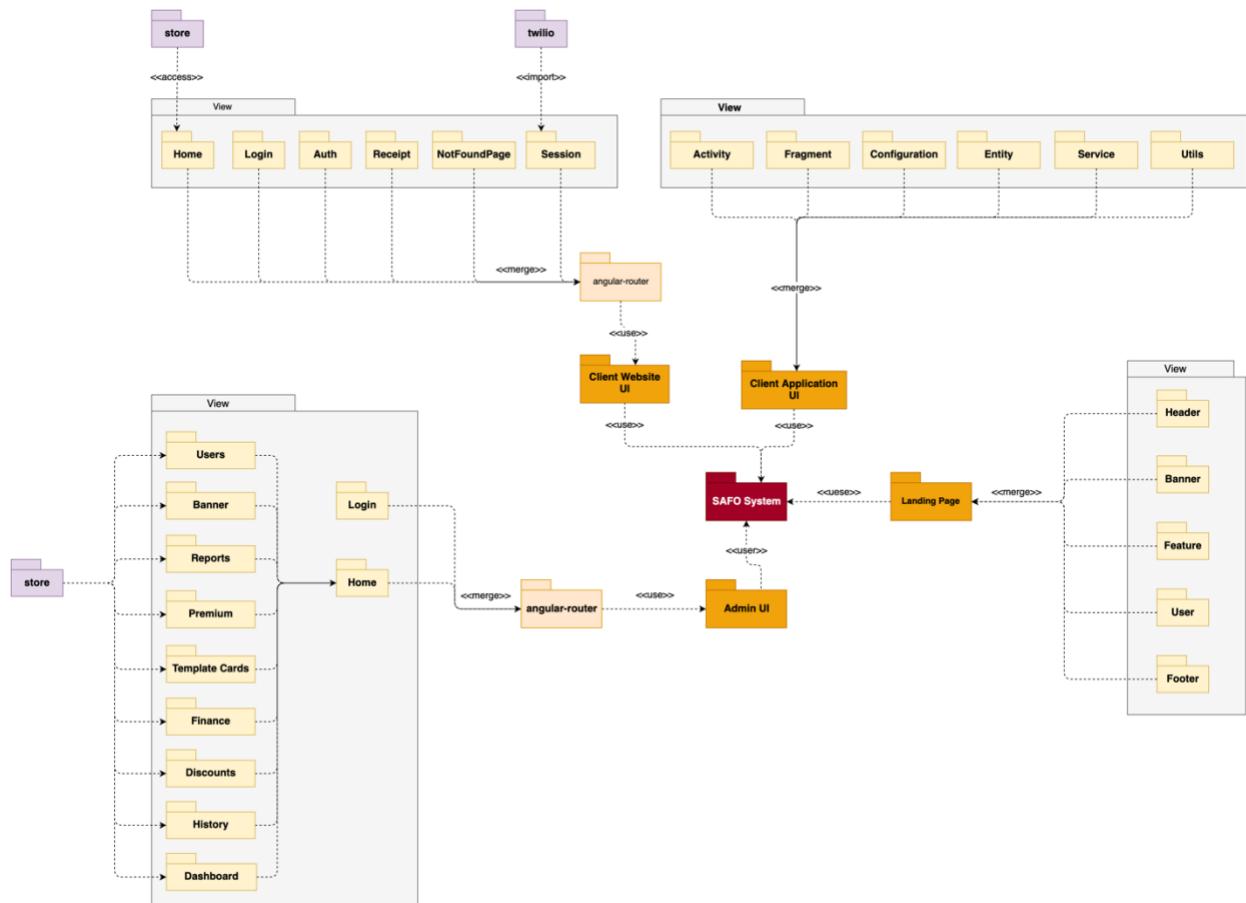
## 2.3 Package Diagram

### 2.3.1 SAFO Package Diagram



## 2.3.2 SAFO Front-end

### 2.3.2.1 Diagram

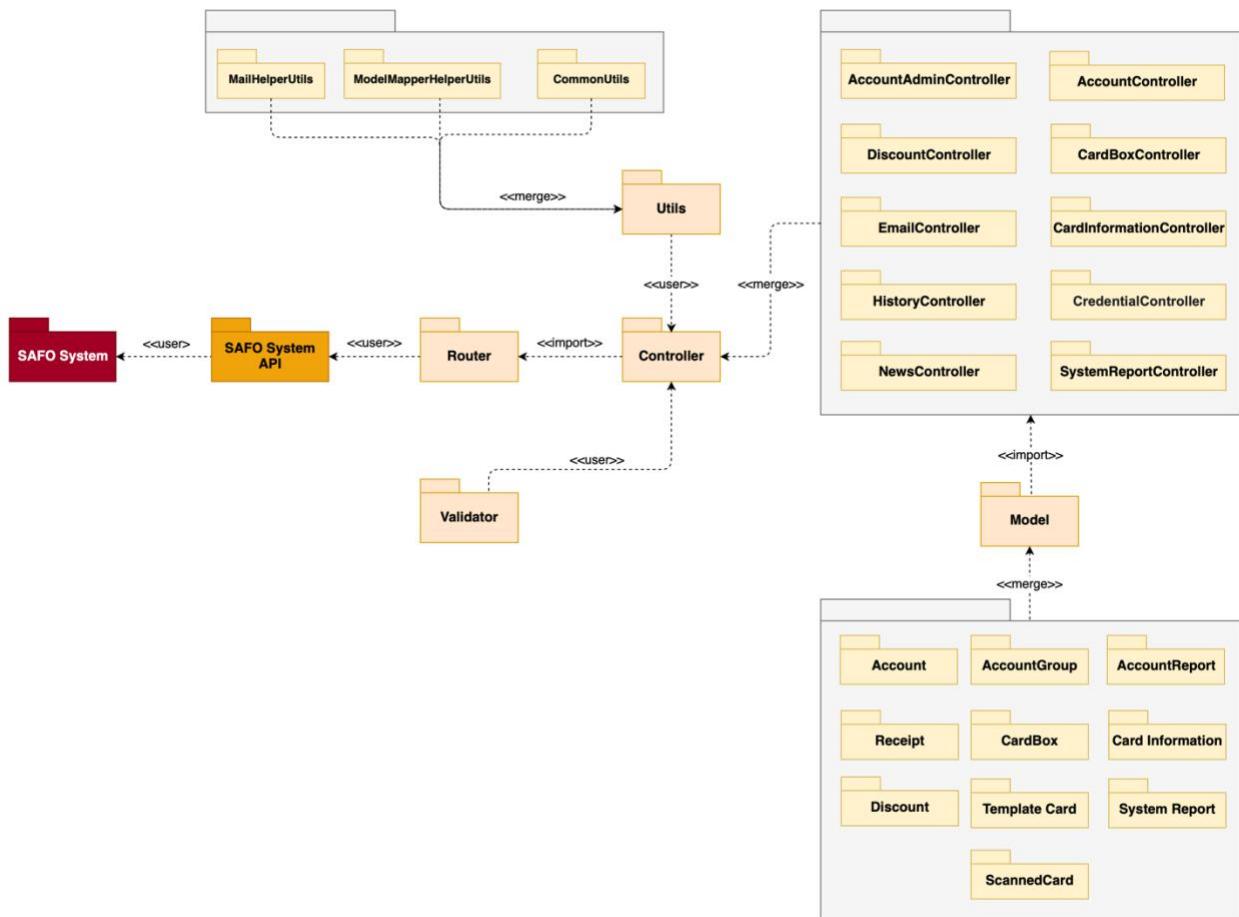


### **2.3.2.2 Package Explanation**

No	Package	Description
1	SAFO System	Contain all code of SAFO system.
2	Client Website UI	Contain all client code for the main website.
3	Landing Page	Contain all code of website appear for the first time.
4	Admin UI	Contain all client code for the managed app
5	Client Application UI	Contain all client code for the main application.
6	angular-router	Declarative routing for the entire system.
7	views	Contain all components used to render interface for the entire system.
8	store	State management
9	twilio	Communication provider

## 2.3.3 SAFO Back-end

### 2.3.3.1 Diagram



### 2.3.3.2 Package Explanation

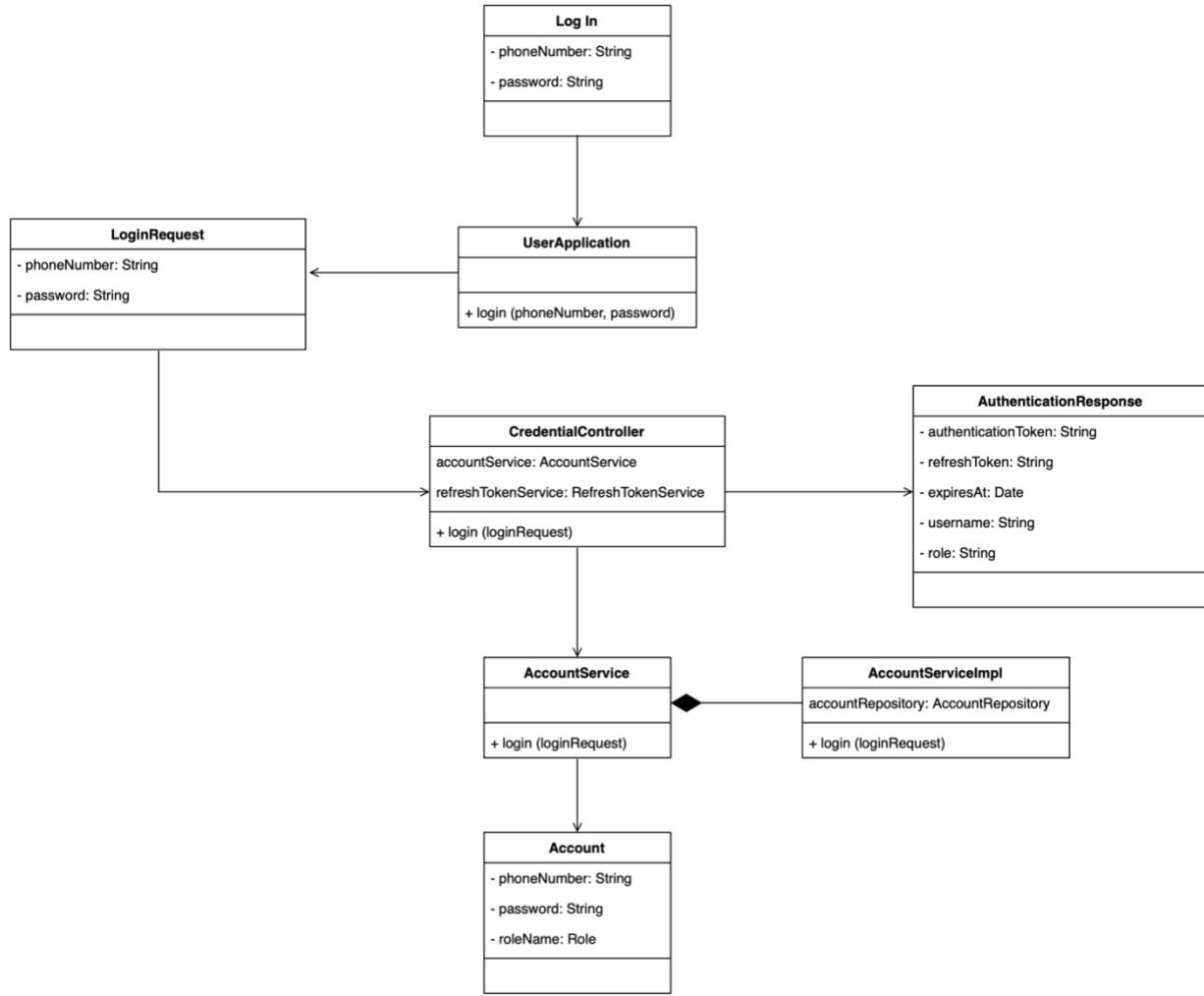
No	Package	Description
1	Controller	Contain classes that are defined as controller, the main function is connected between client and server through API endpoints.
2	Router	A concept that defines how an application will respond when a user requests a particular endpoint. That endpoint is usually a URI or a path with a specific Request method (POST, PUT, GET, ...).
3	Model	Defines the properties of an object.
4	Validator	Contains various types of input check functions.
5	Utils	Contains standard form formats.

### 3. System Detailed Design

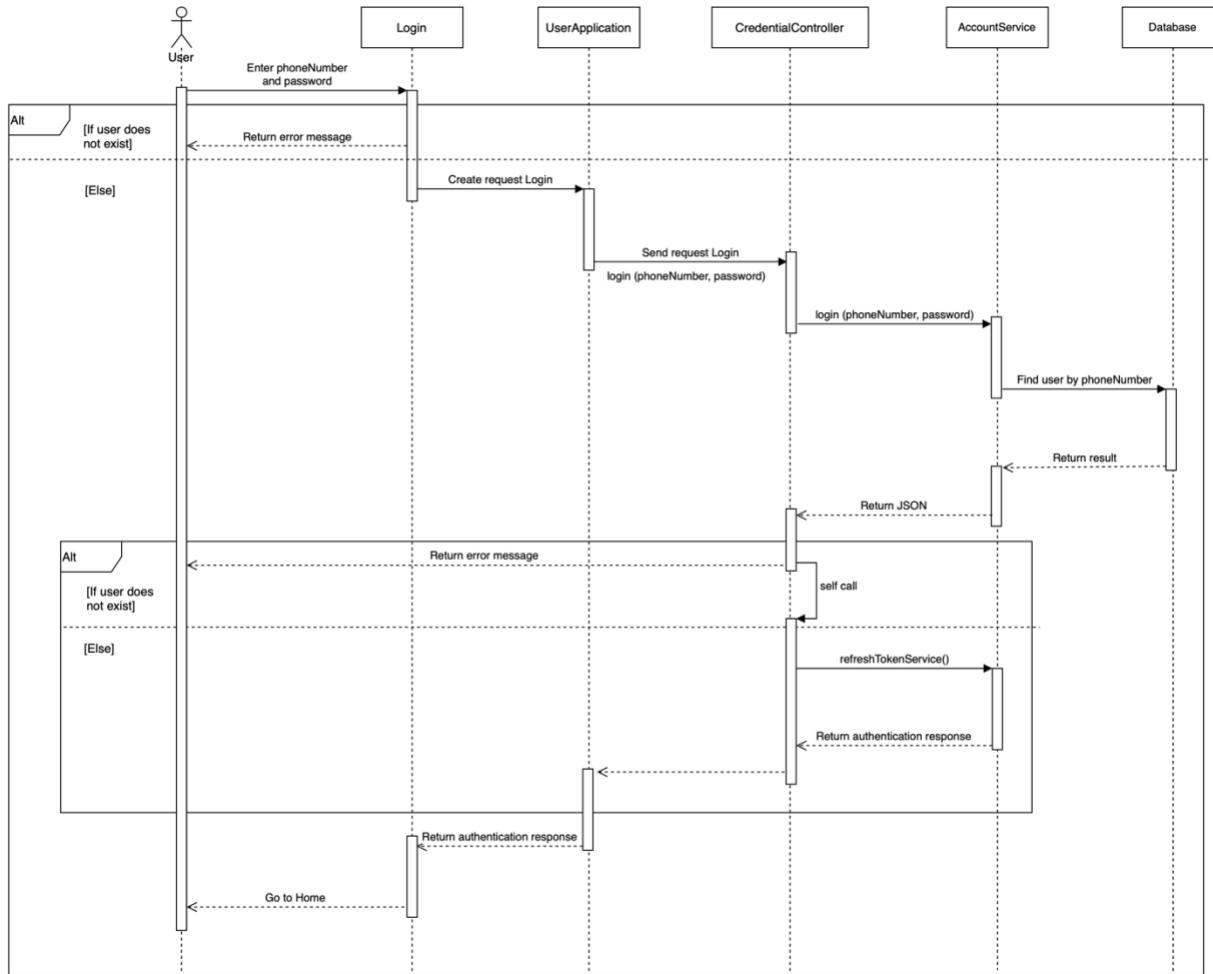
#### 3.1 Home

##### 3.1.1 Log In

a. Class diagram

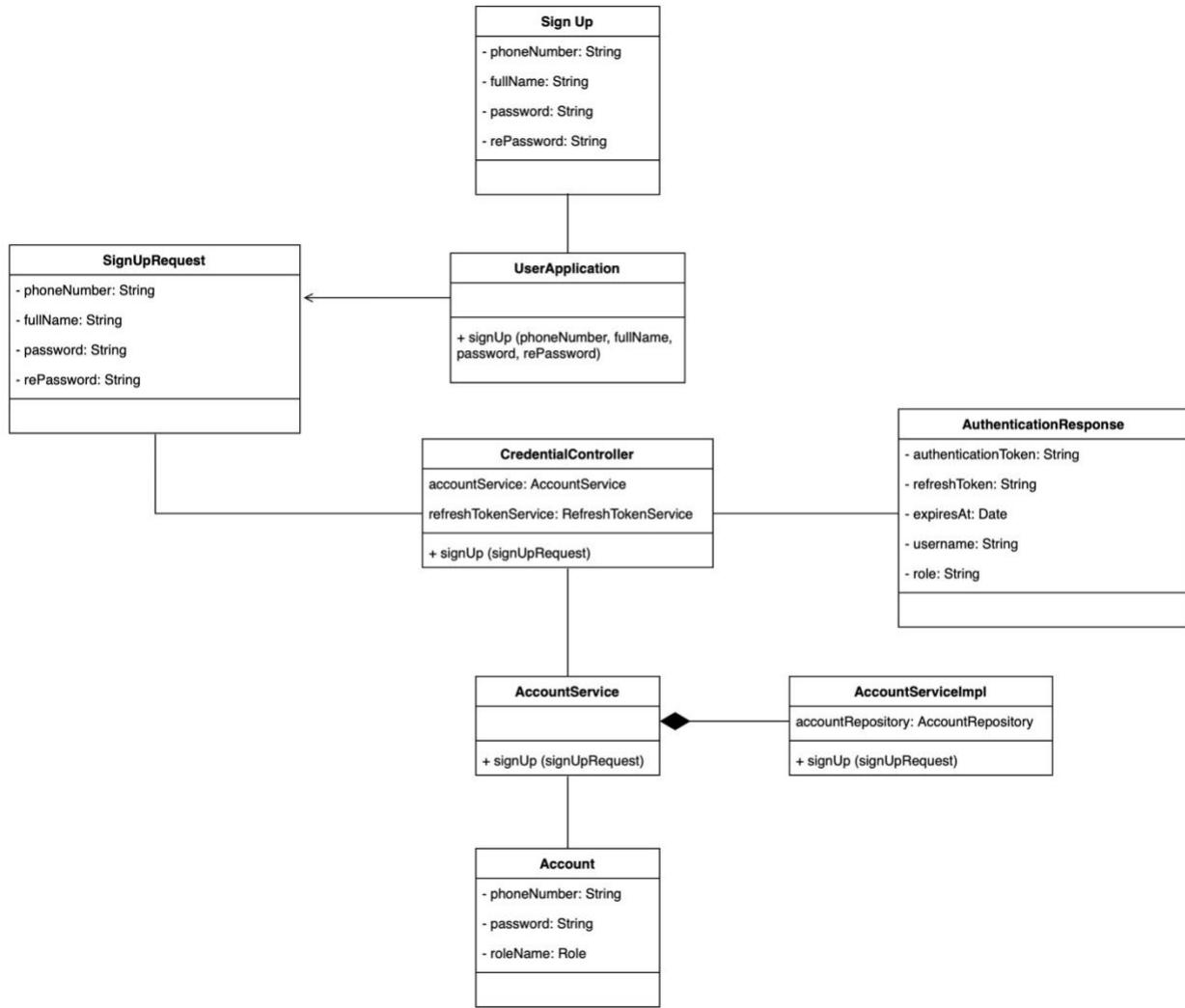


## b. Sequence diagram

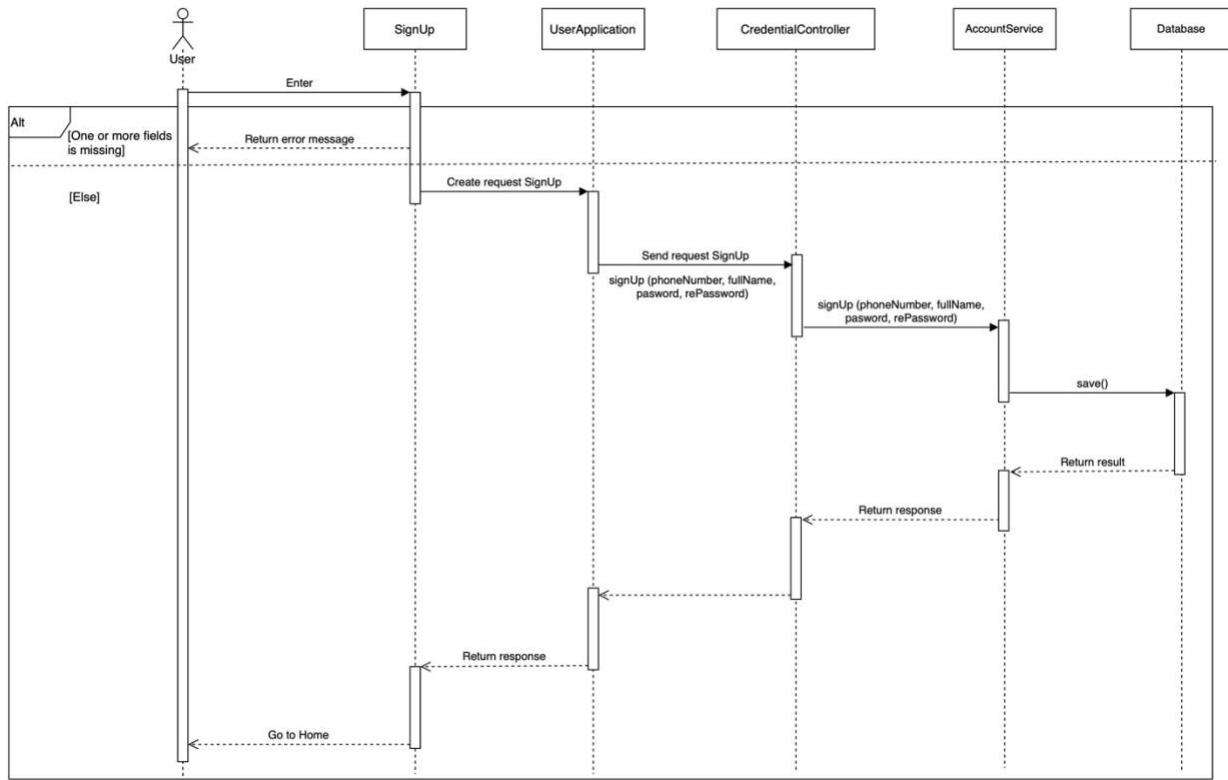


### 3.1.2 Sign Up

#### a. Class Diagram

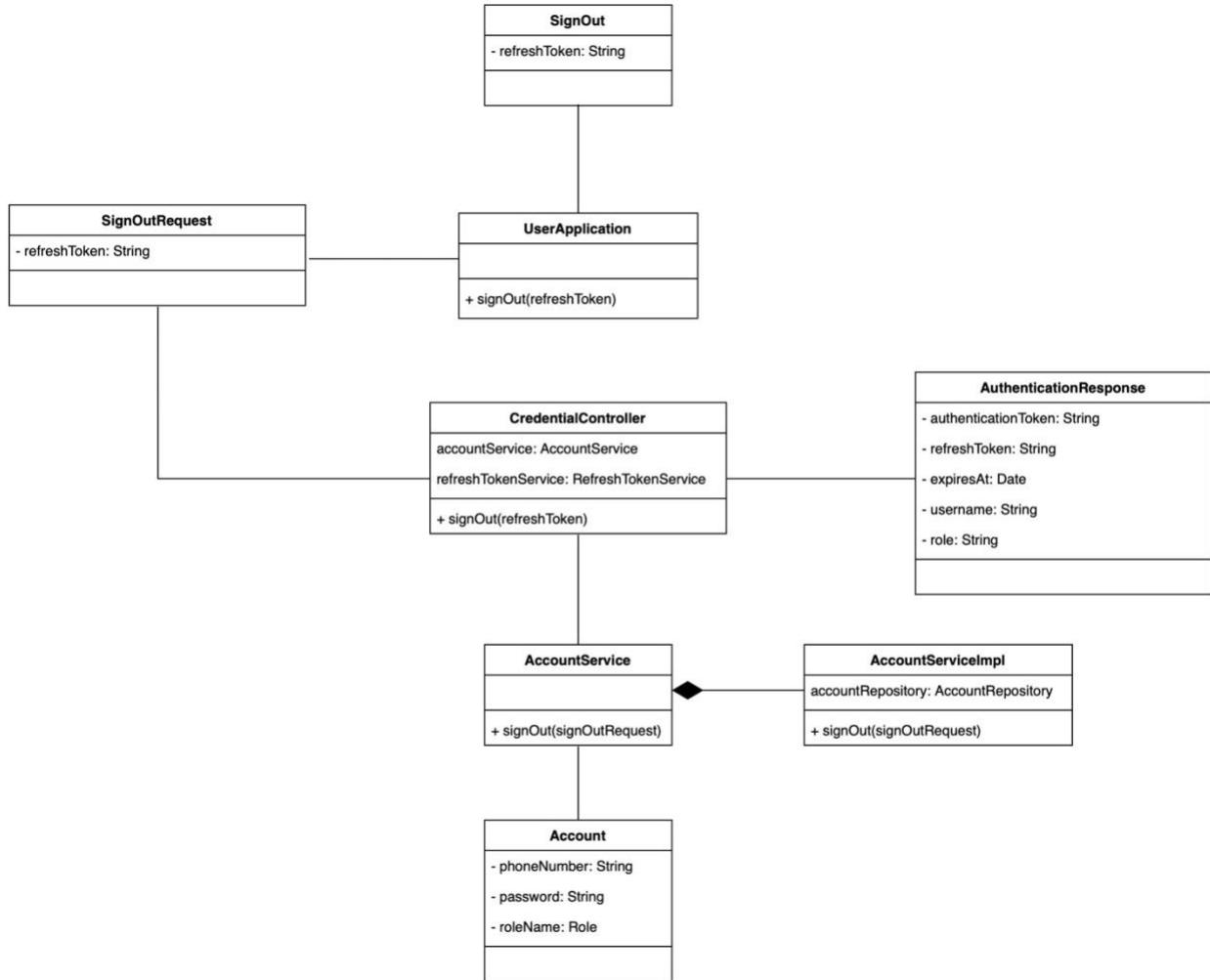


b. Sequence Diagram

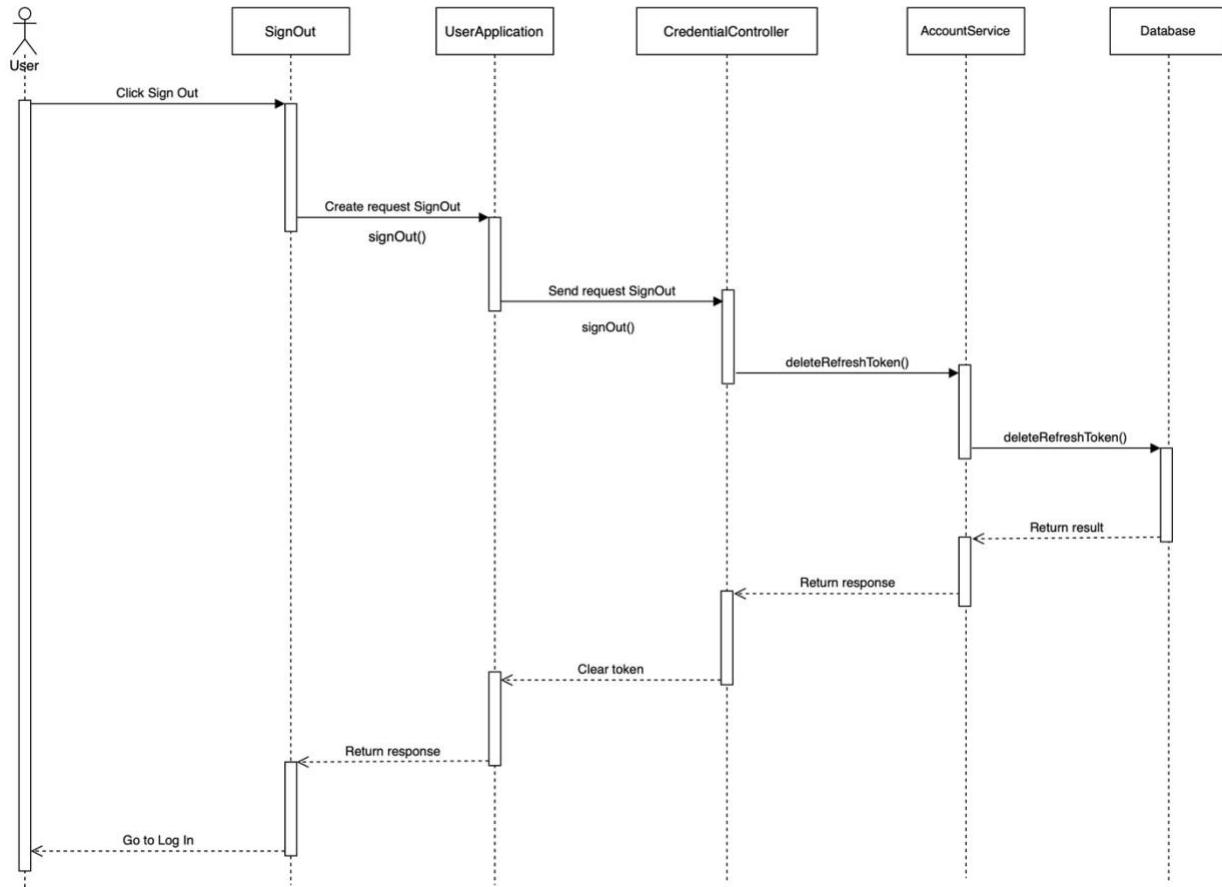


### 3.1.3 Log out

a. Class Diagram



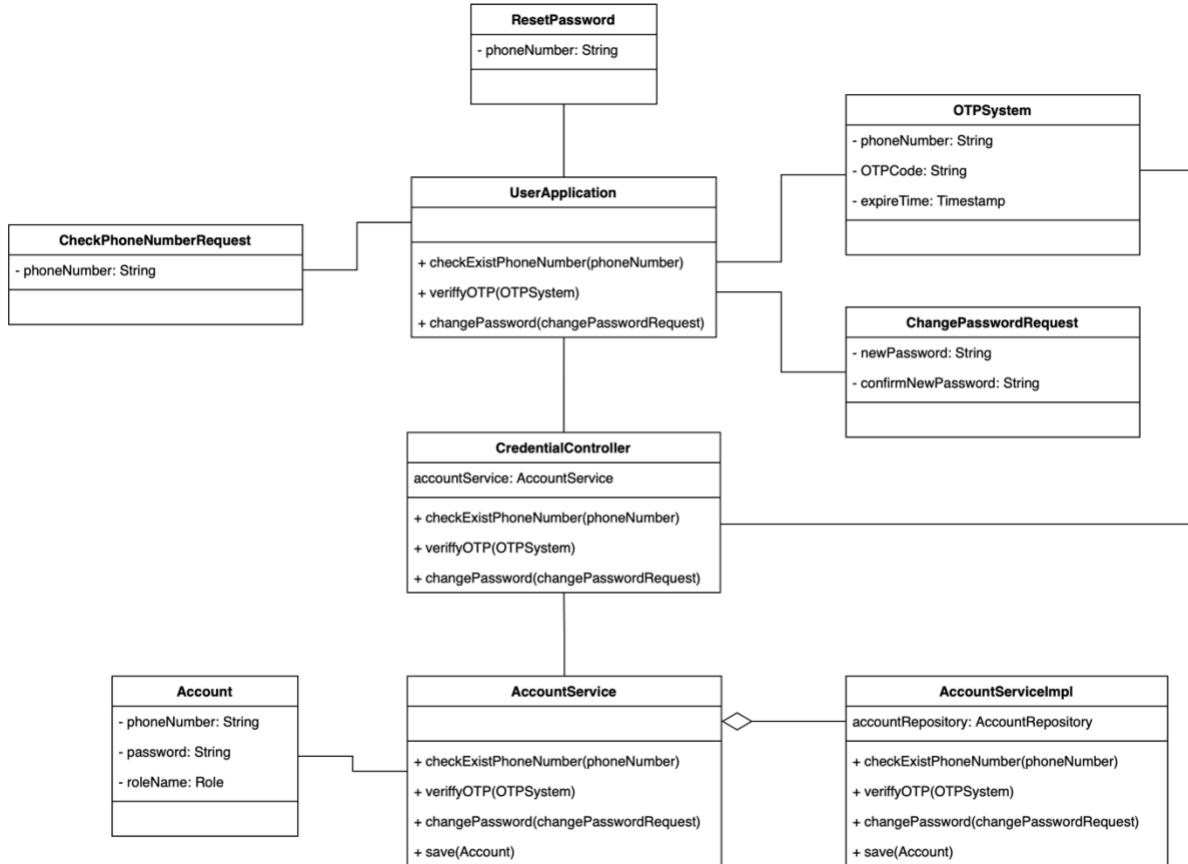
b. Sequence Diagram



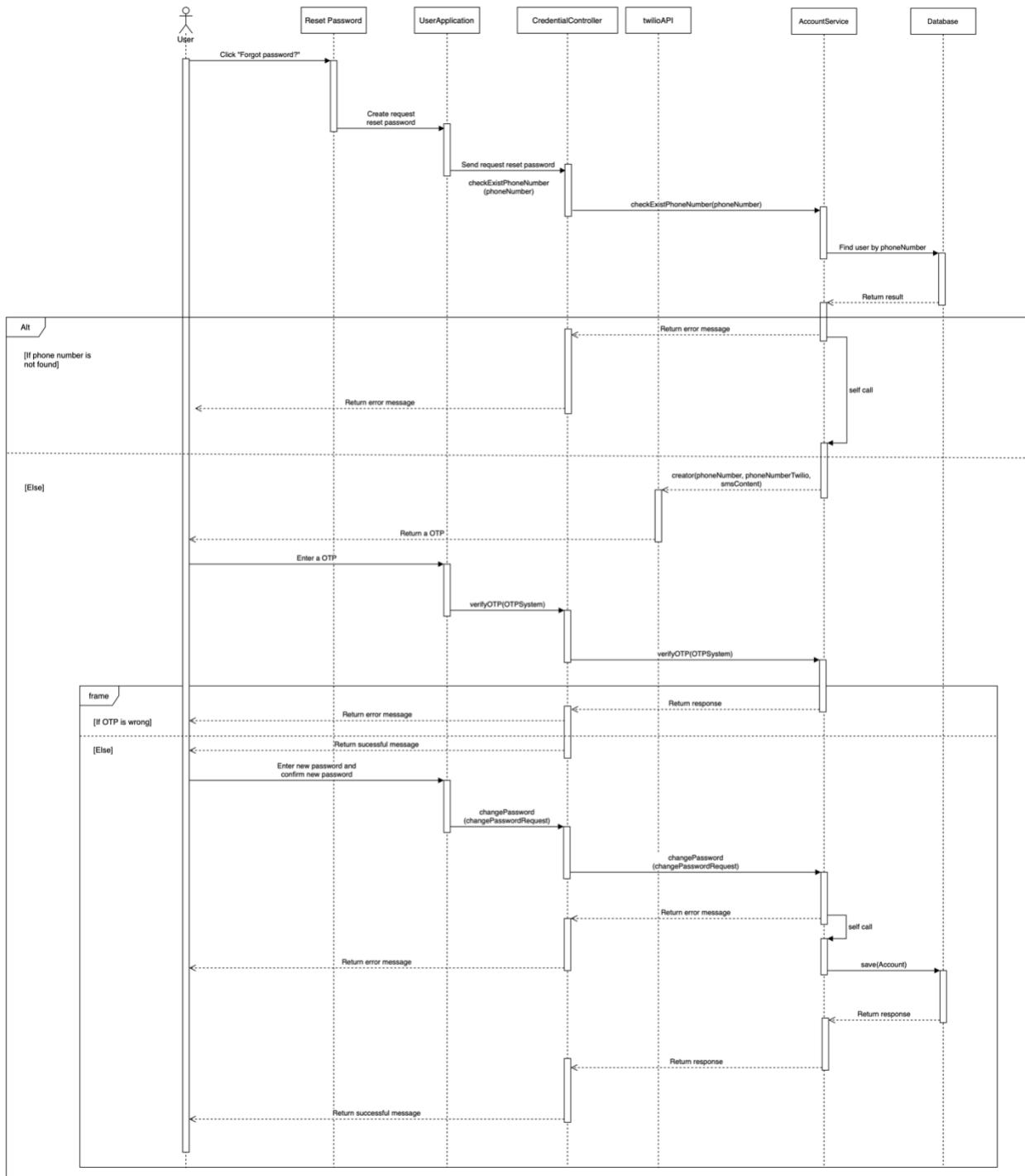
## 3.2 User Profile

### 3.2.1 Reset Password

a. Class diagram

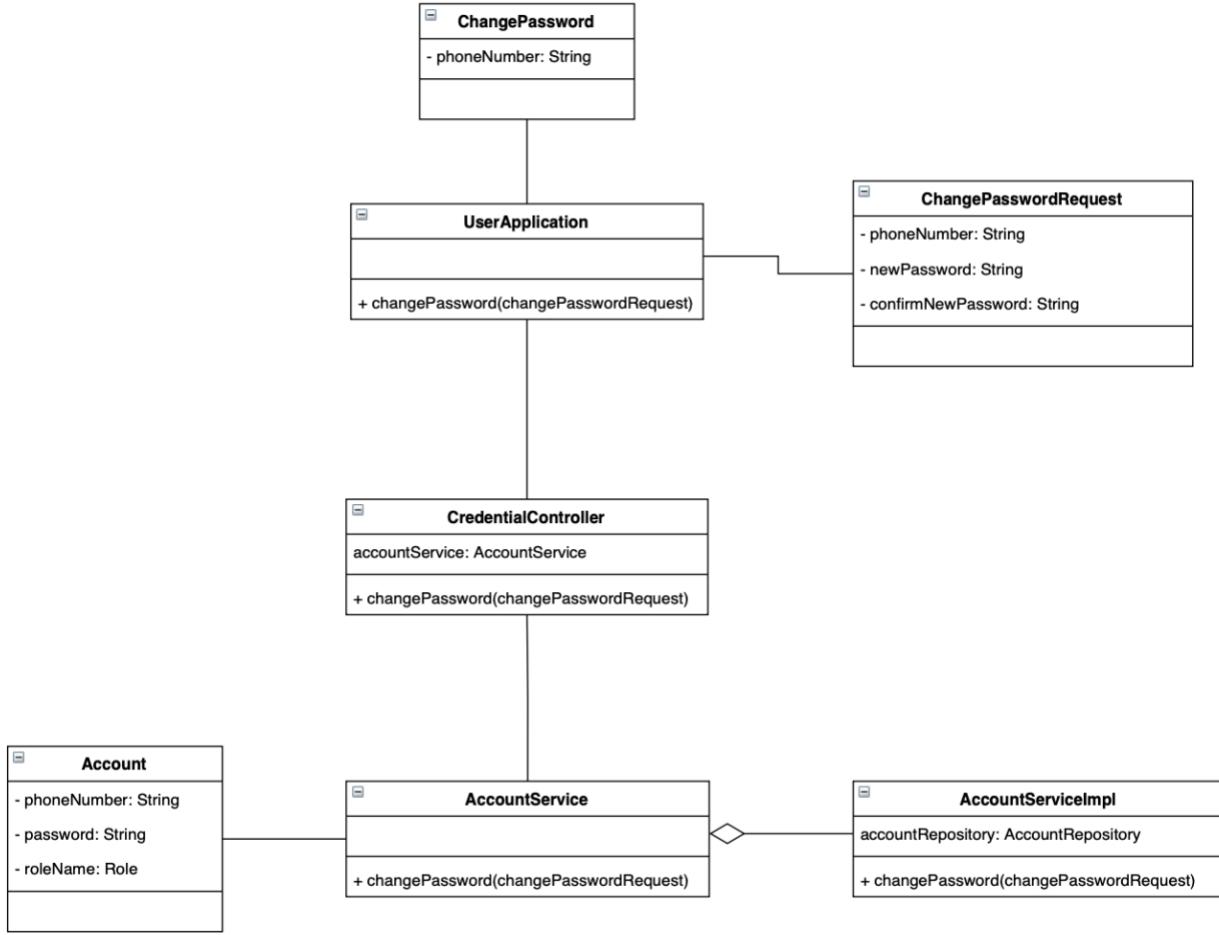


## b. Sequence diagram

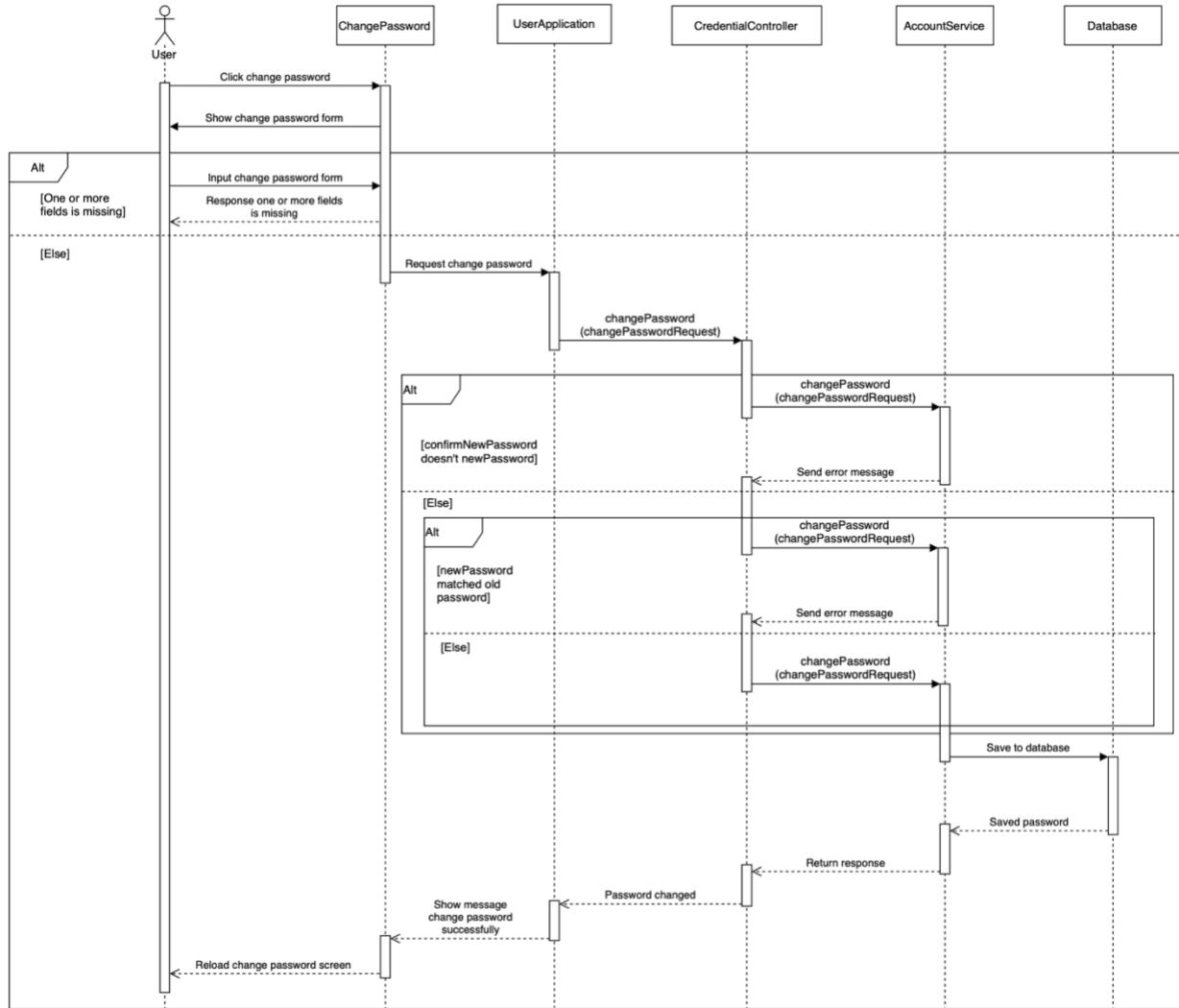


### 3.2.2 Change Password

a. Class Diagram

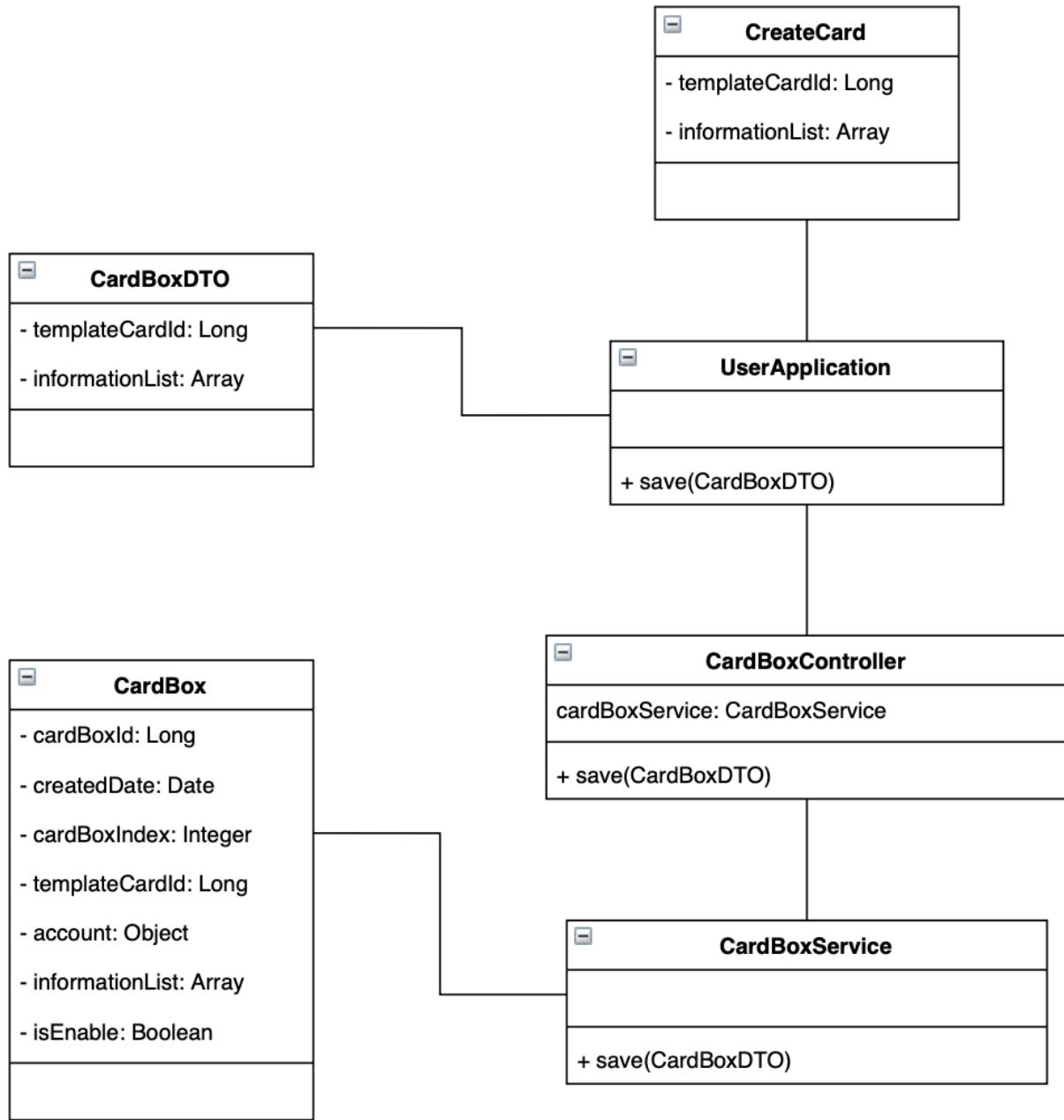


b. Sequence Diagram

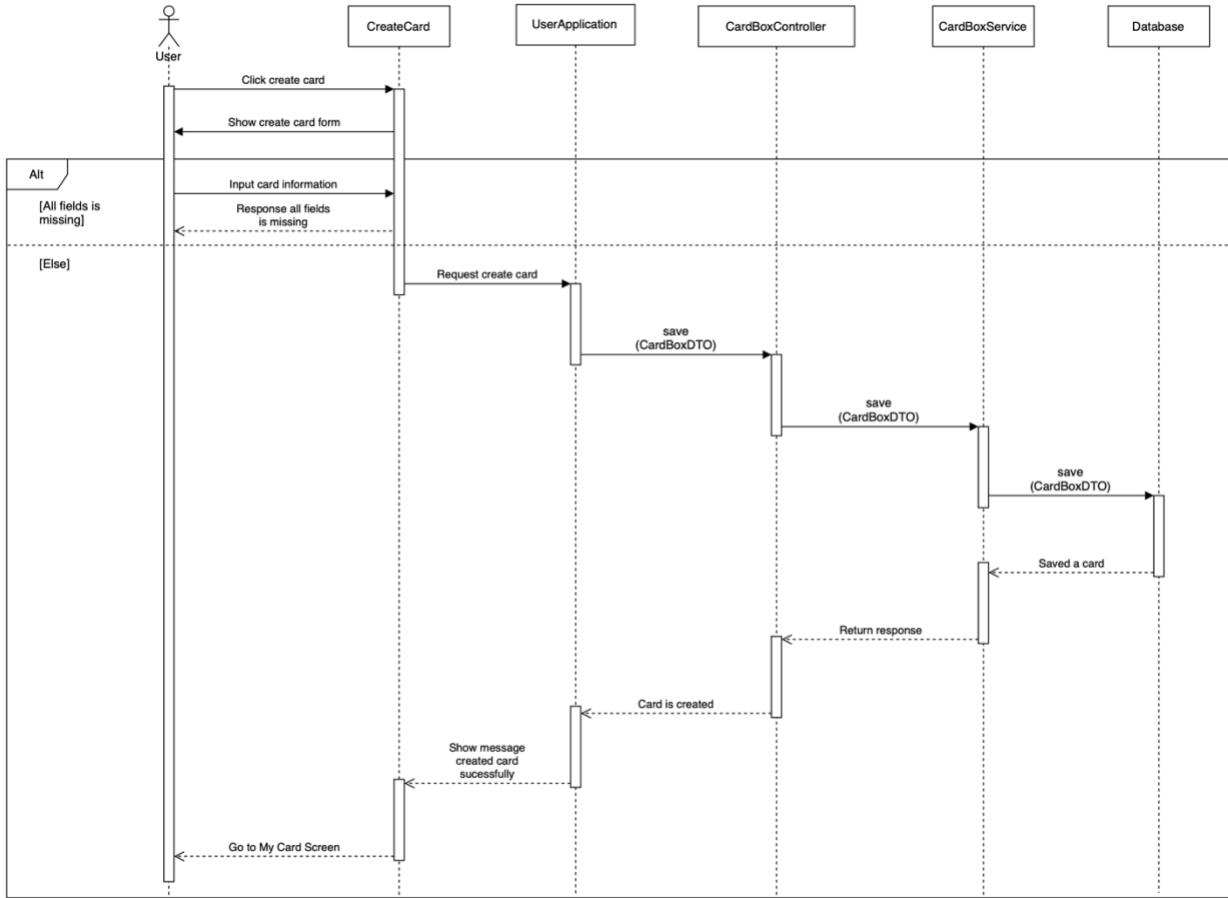


### 3.2.3 Create Card

a. Class Diagram

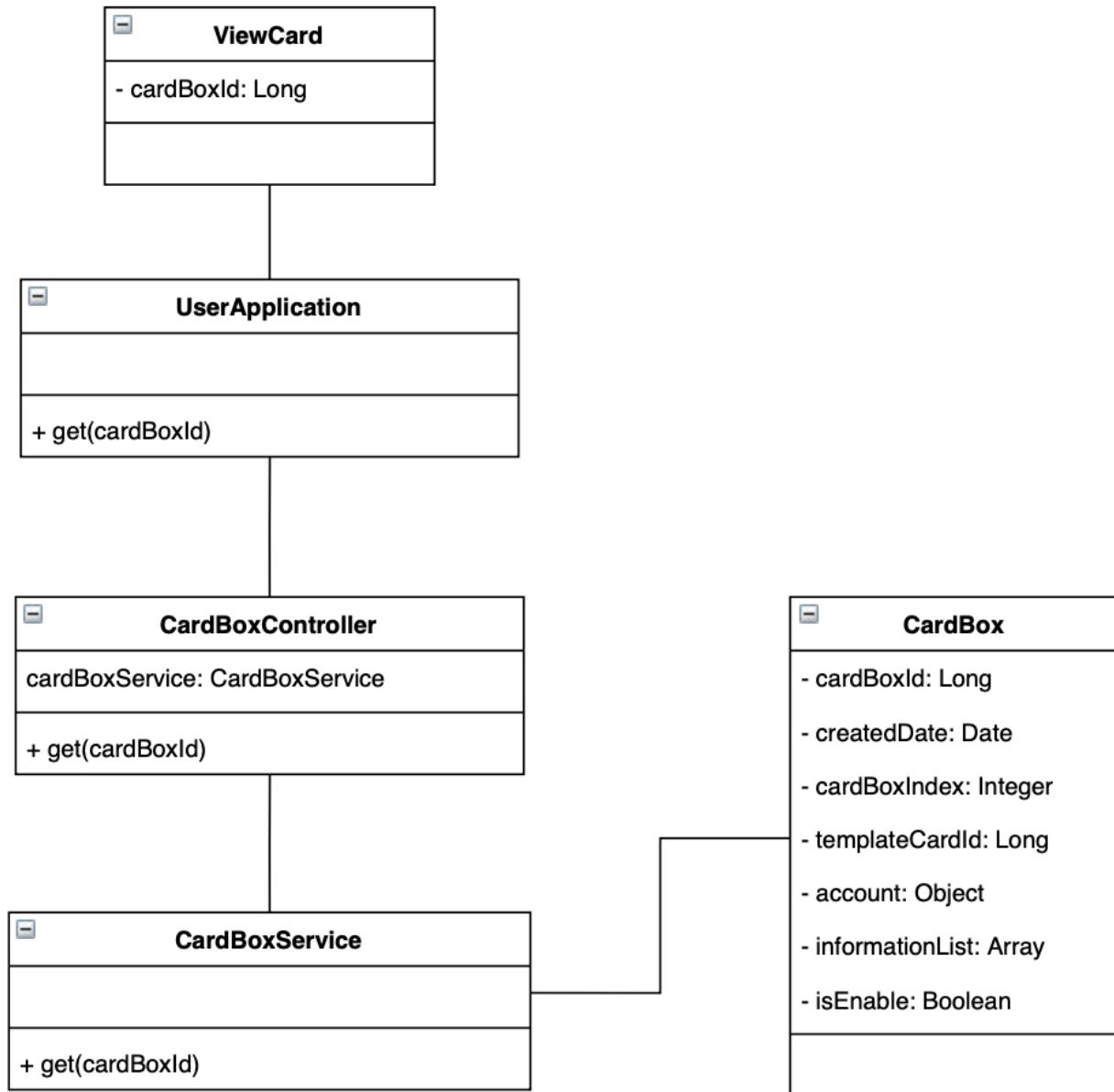


b. Sequence Diagram

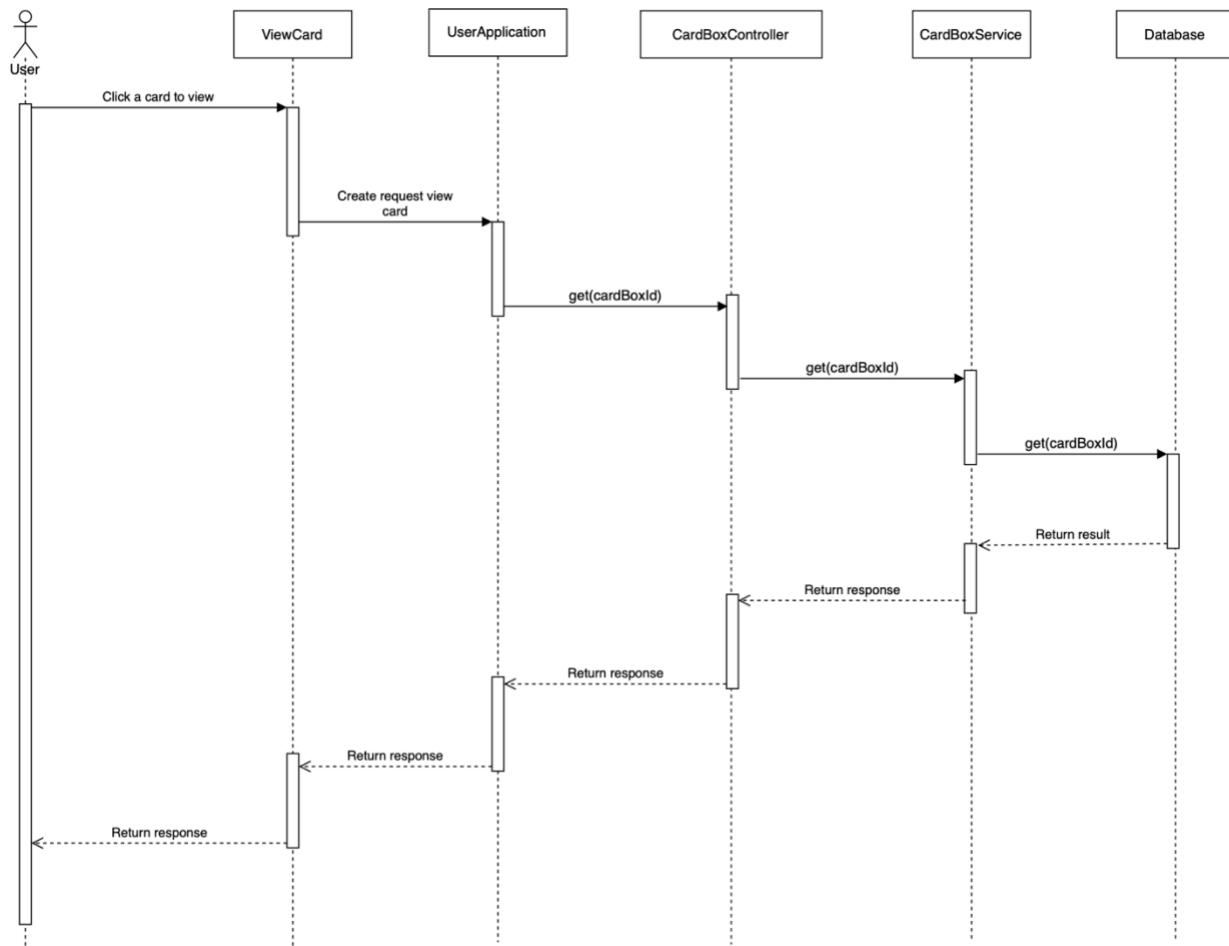


### 3.2.4 View Card

a. Class Diagram

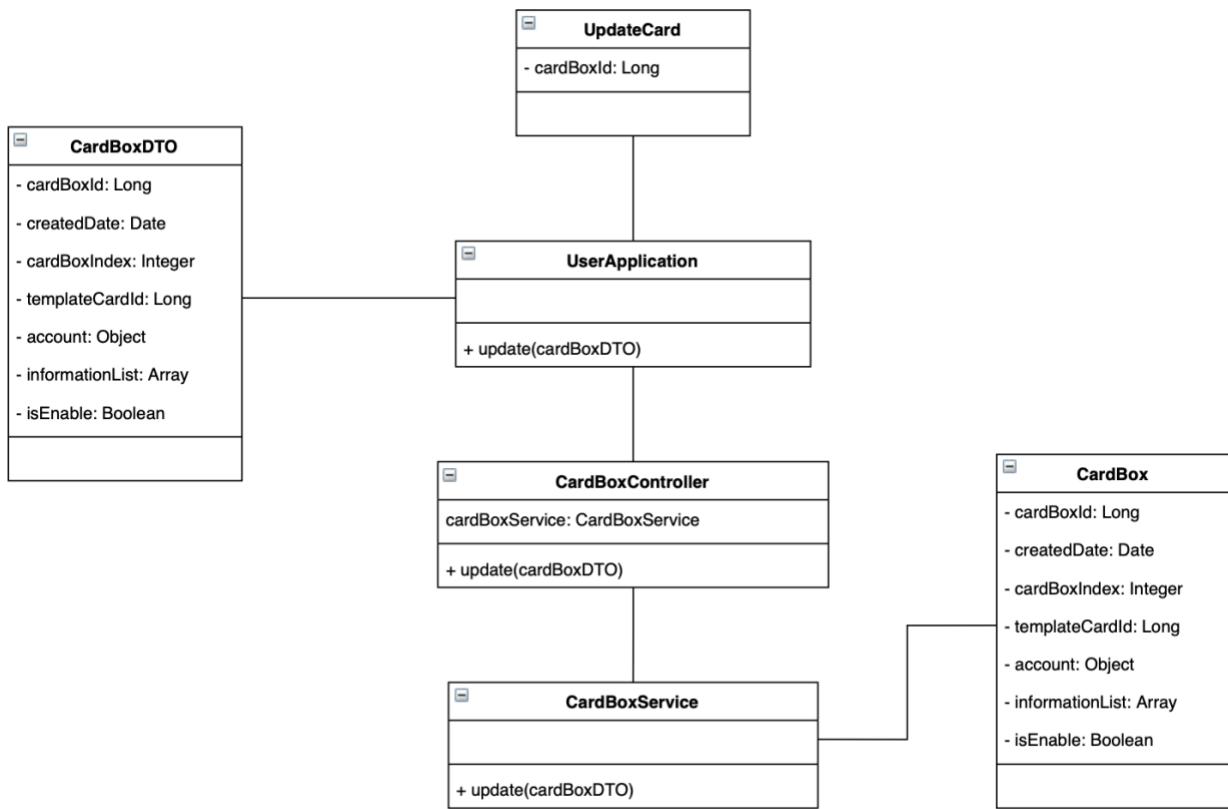


b. Sequence Diagram

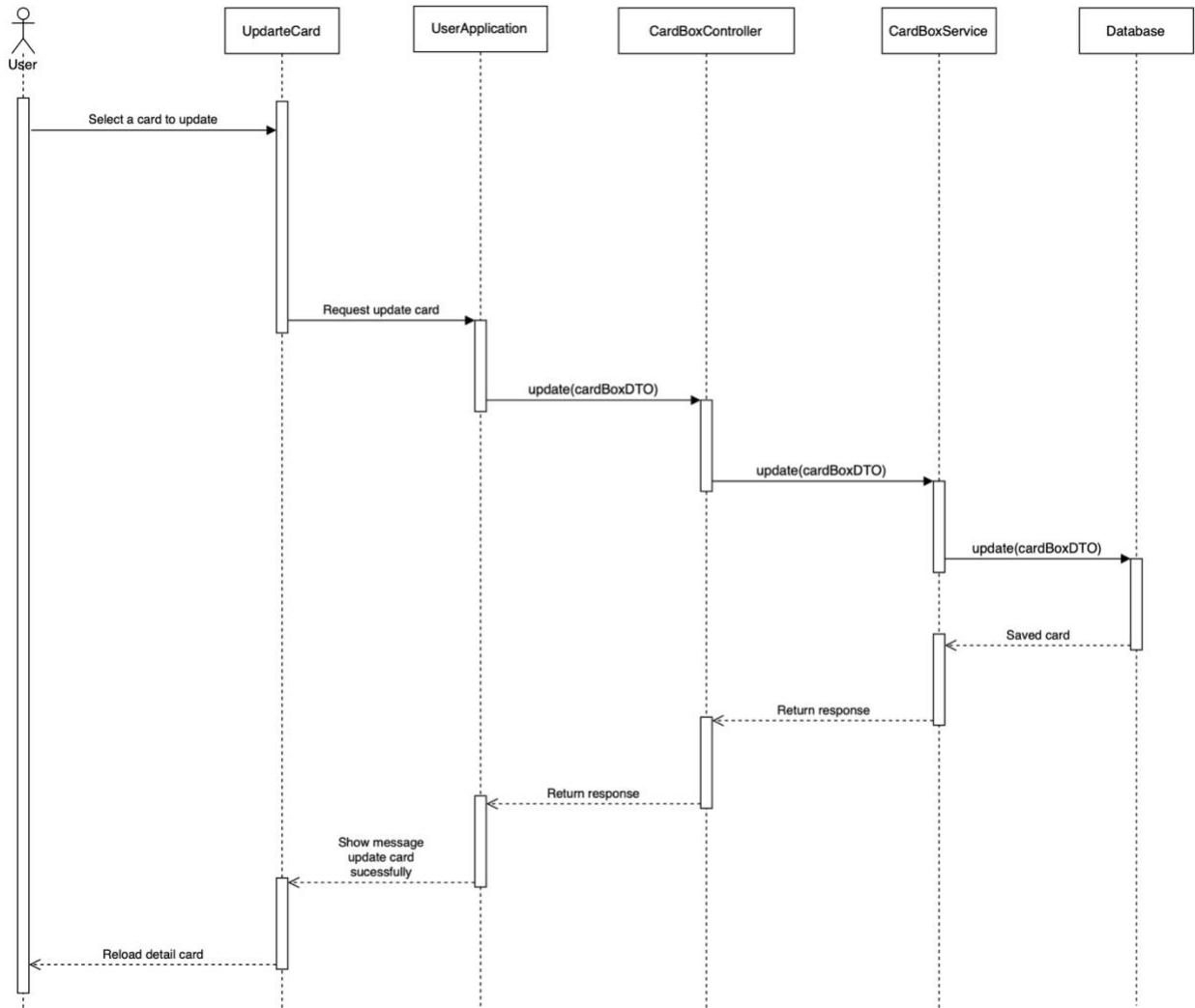


### 3.2.5 Update Card

a. Class Diagram

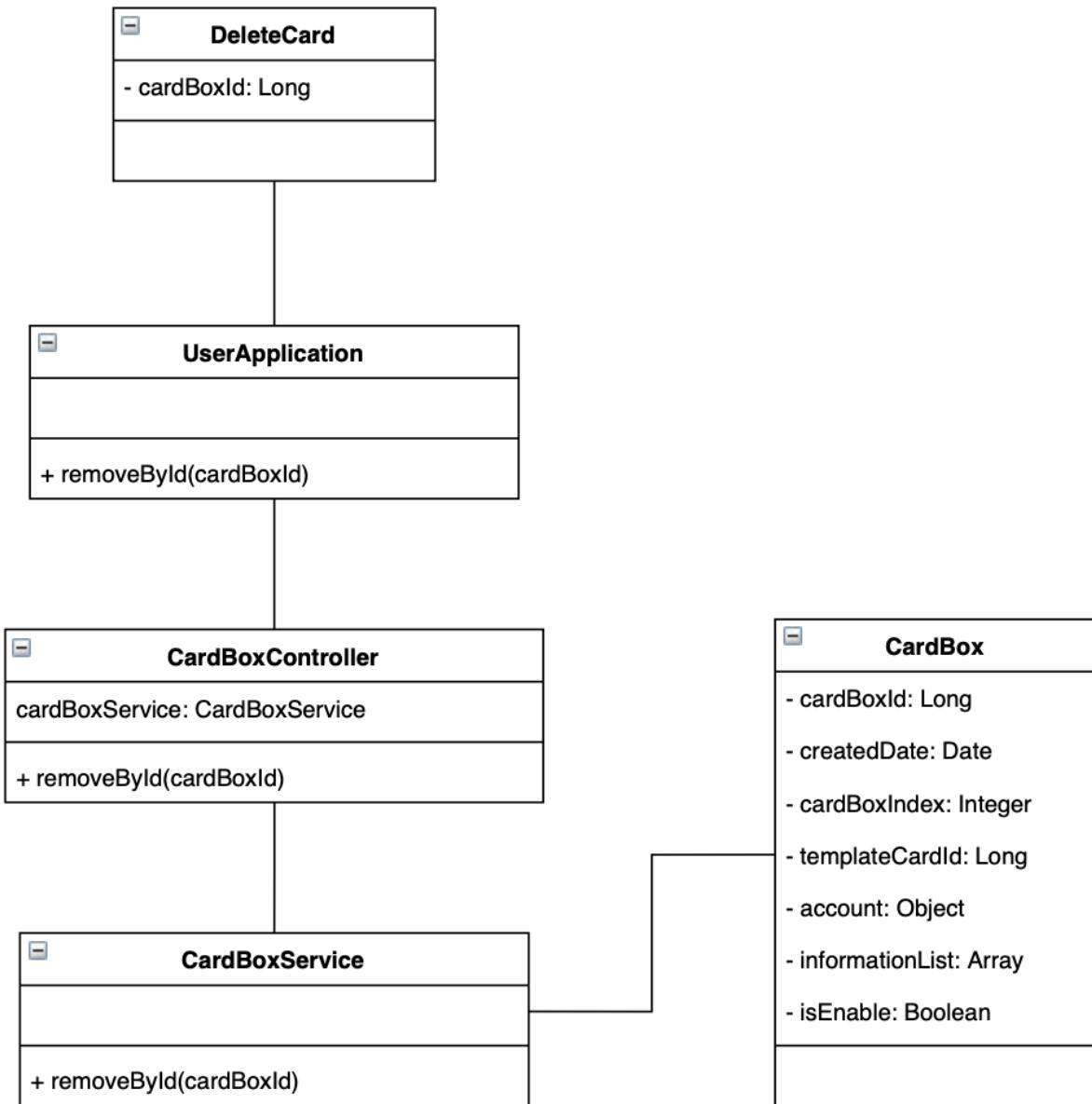


b. Sequence Diagram

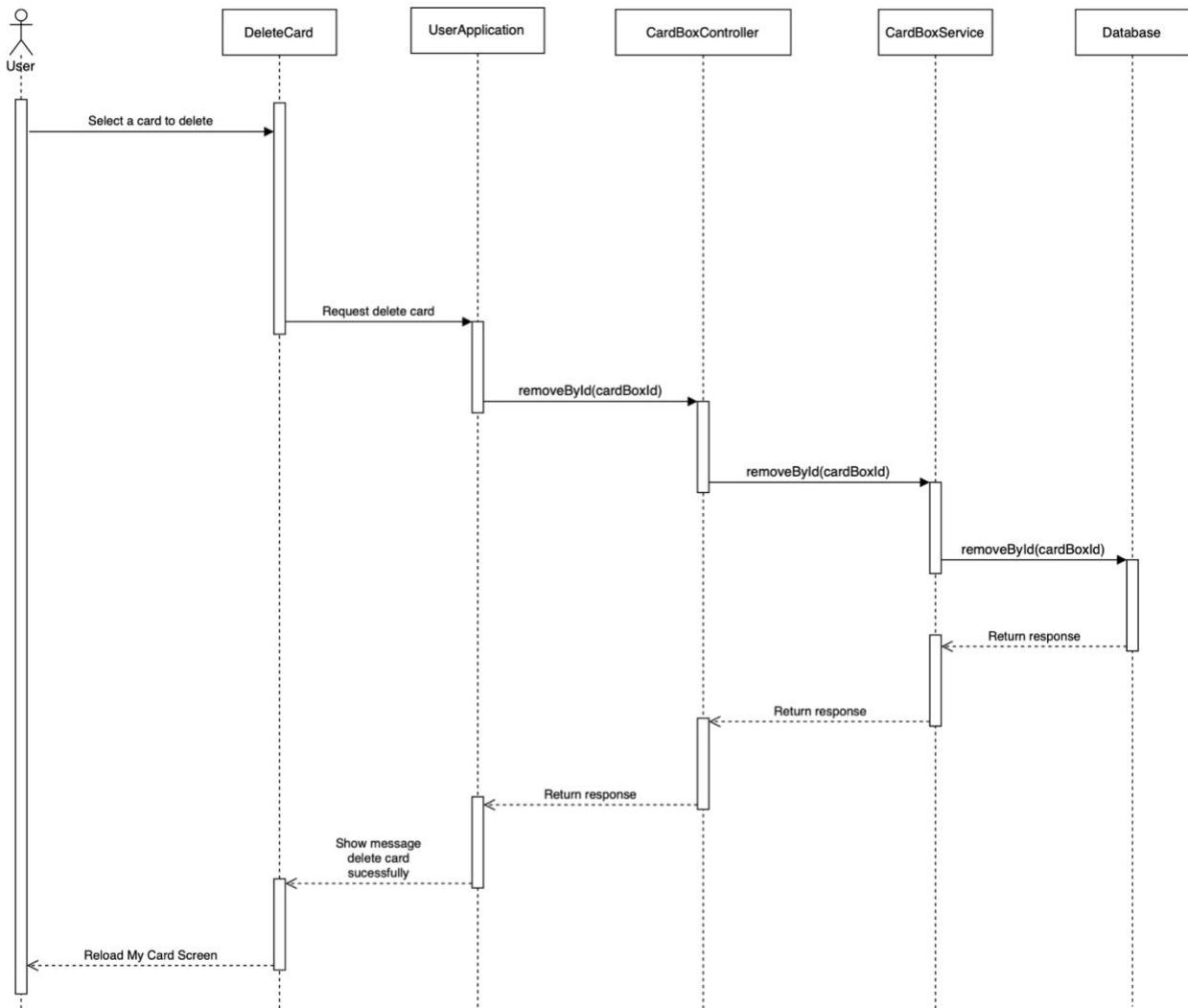


### 3.2.6 Delete Card

a. Class Diagram



b. Sequence Diagram



### 3.2.7 Scan QR Code

a. Class Diagram

b. Sequence Diagram

### 3.2.8 Enter Share Code

a. Class Diagram

b. Sequence Diagram

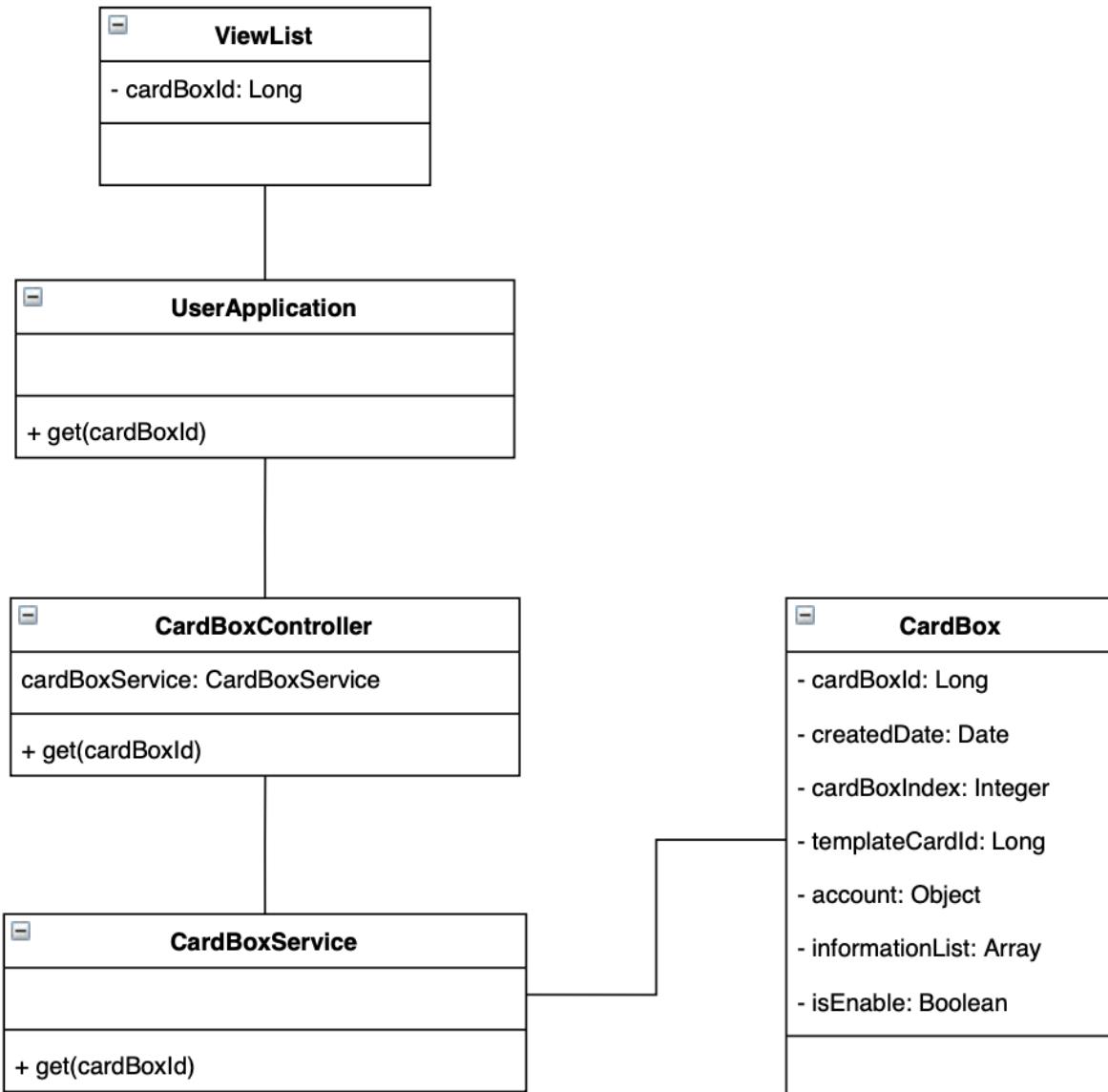
### 3.2.9 Share Card

a. Class Diagram

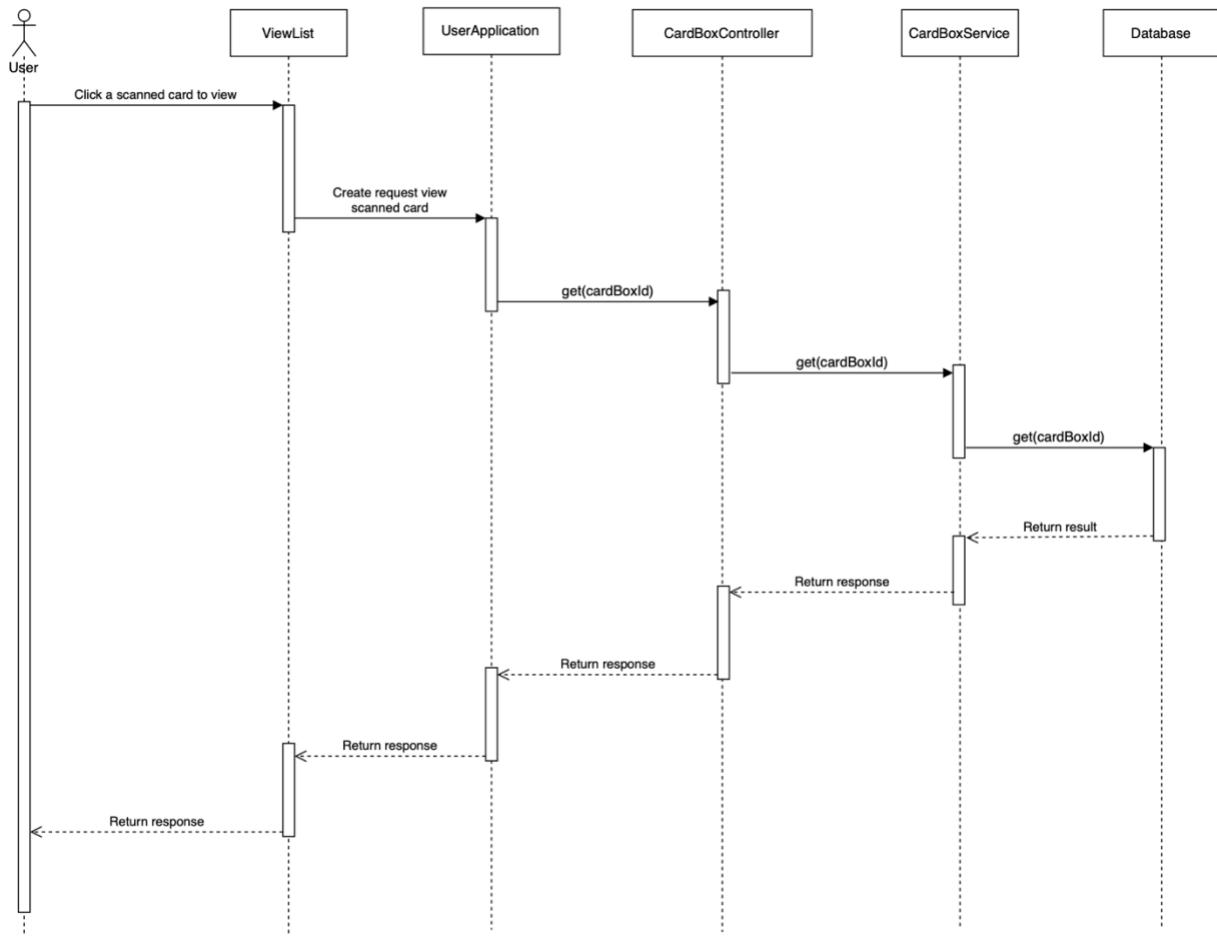
b. Sequence Diagram

### 3.2.10 View List

a. Class Diagram

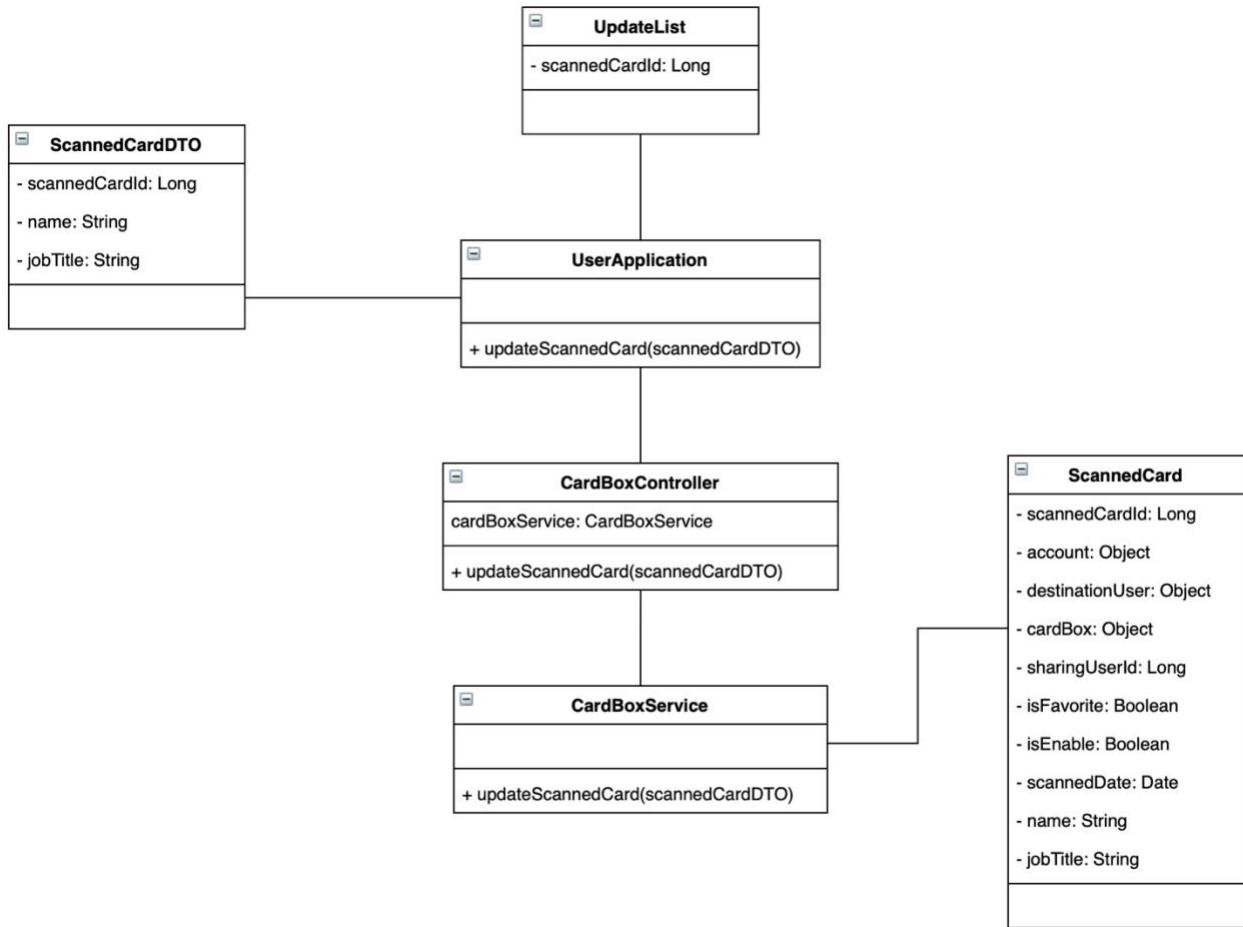


b. Sequence Diagram

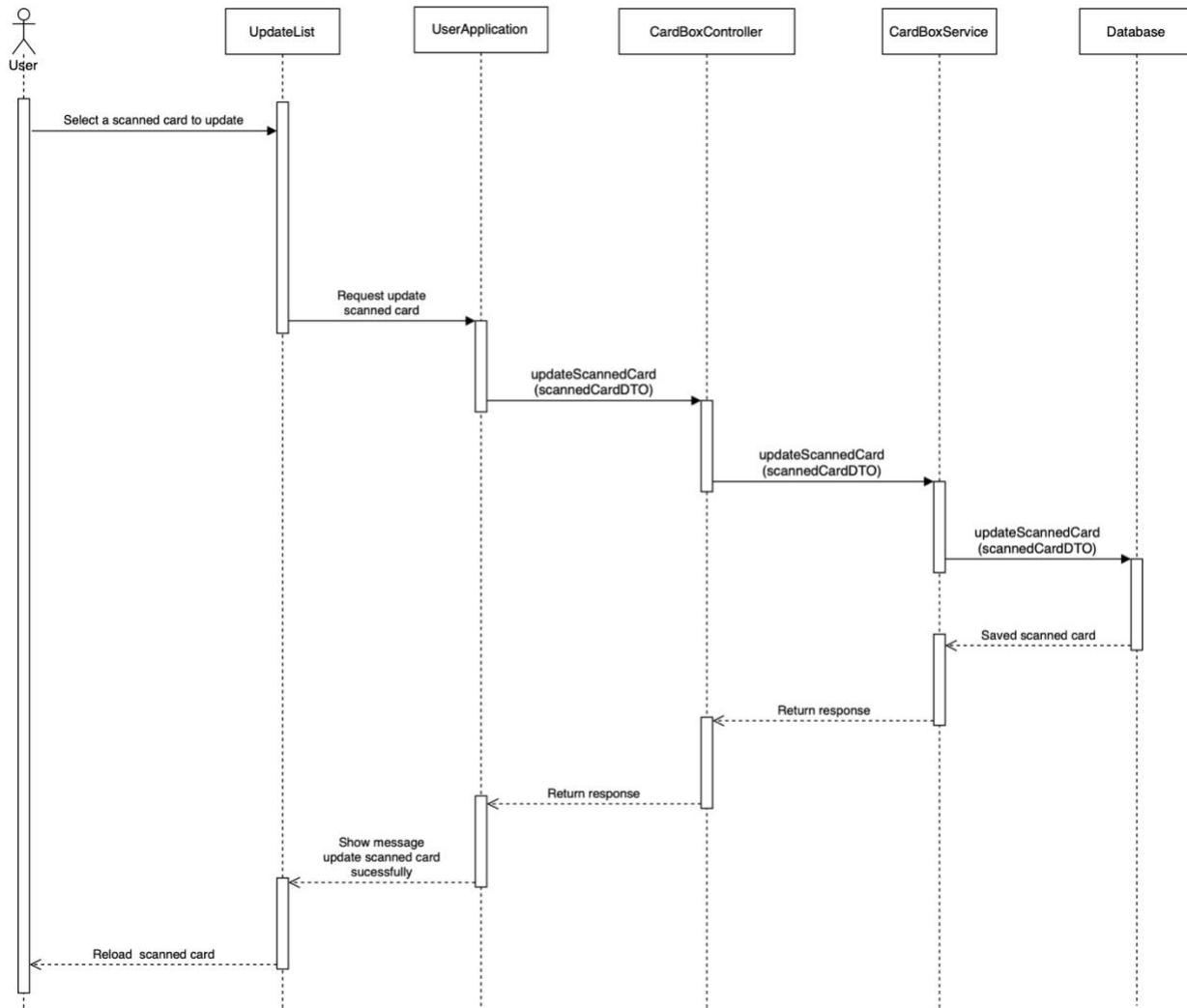


### 3.2.11 Update List

a. Class Diagram

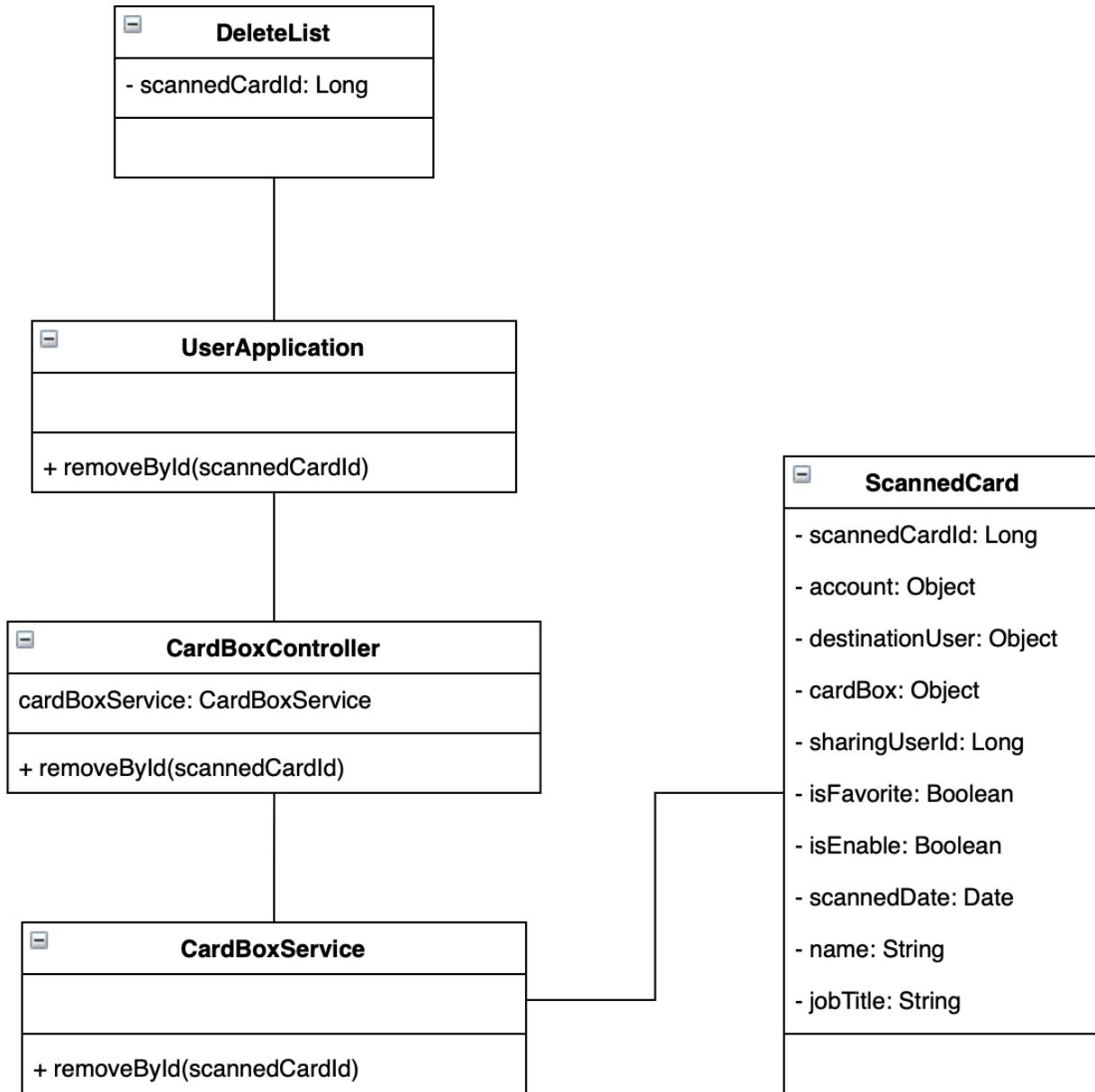


b. Sequence Diagram

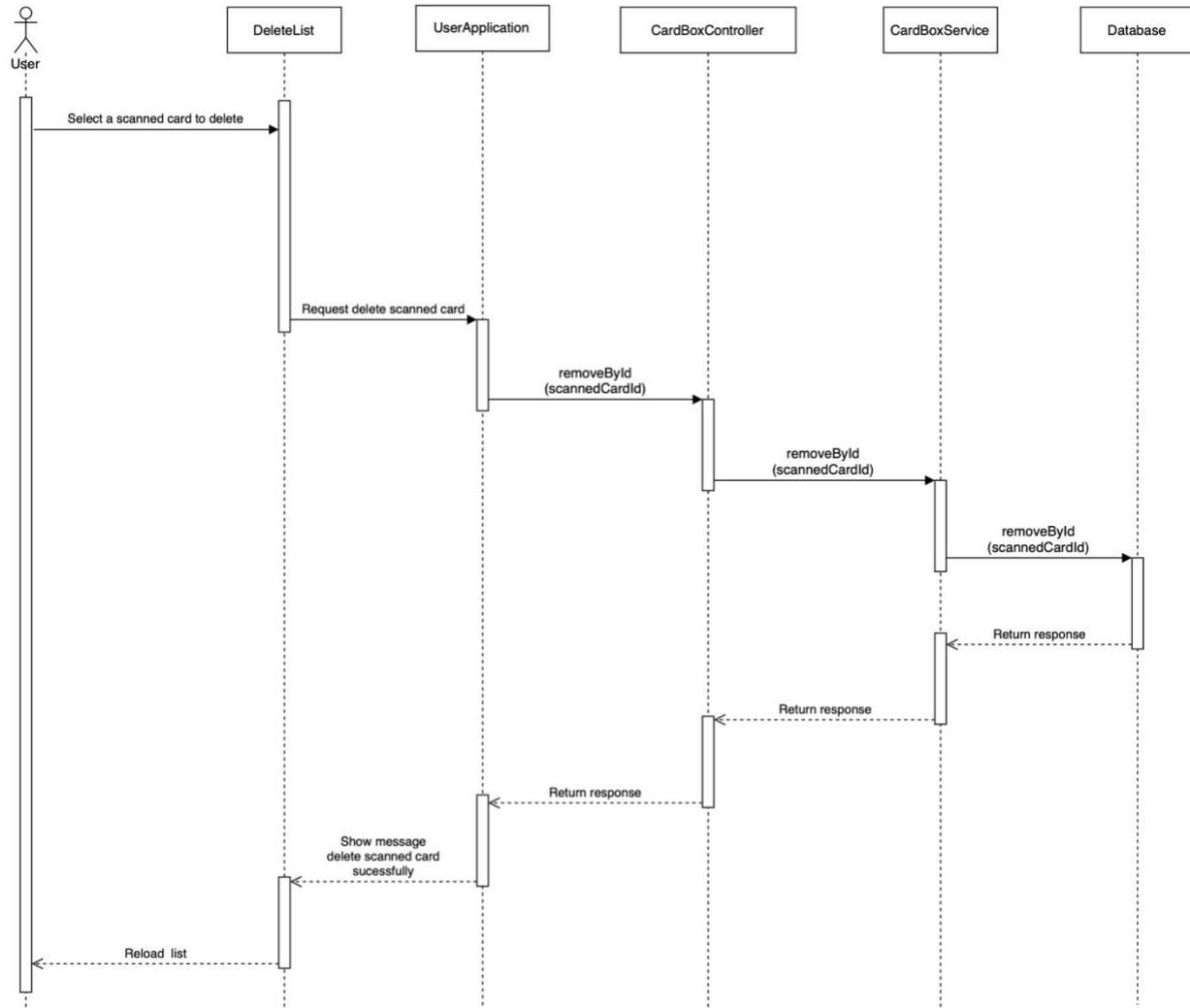


### 3.2.12 Delete List

a. Class Diagram

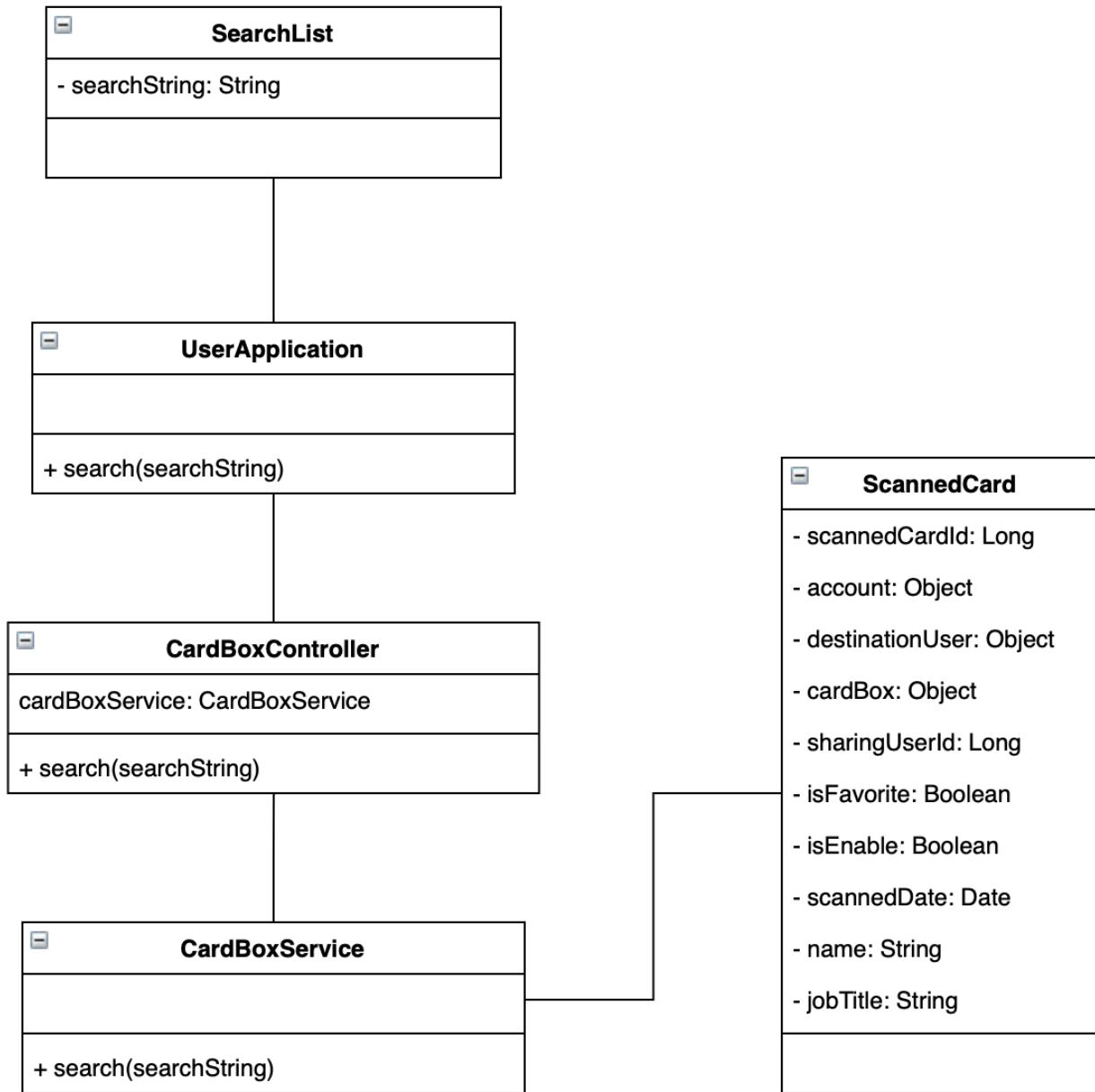


b. Sequence Diagram

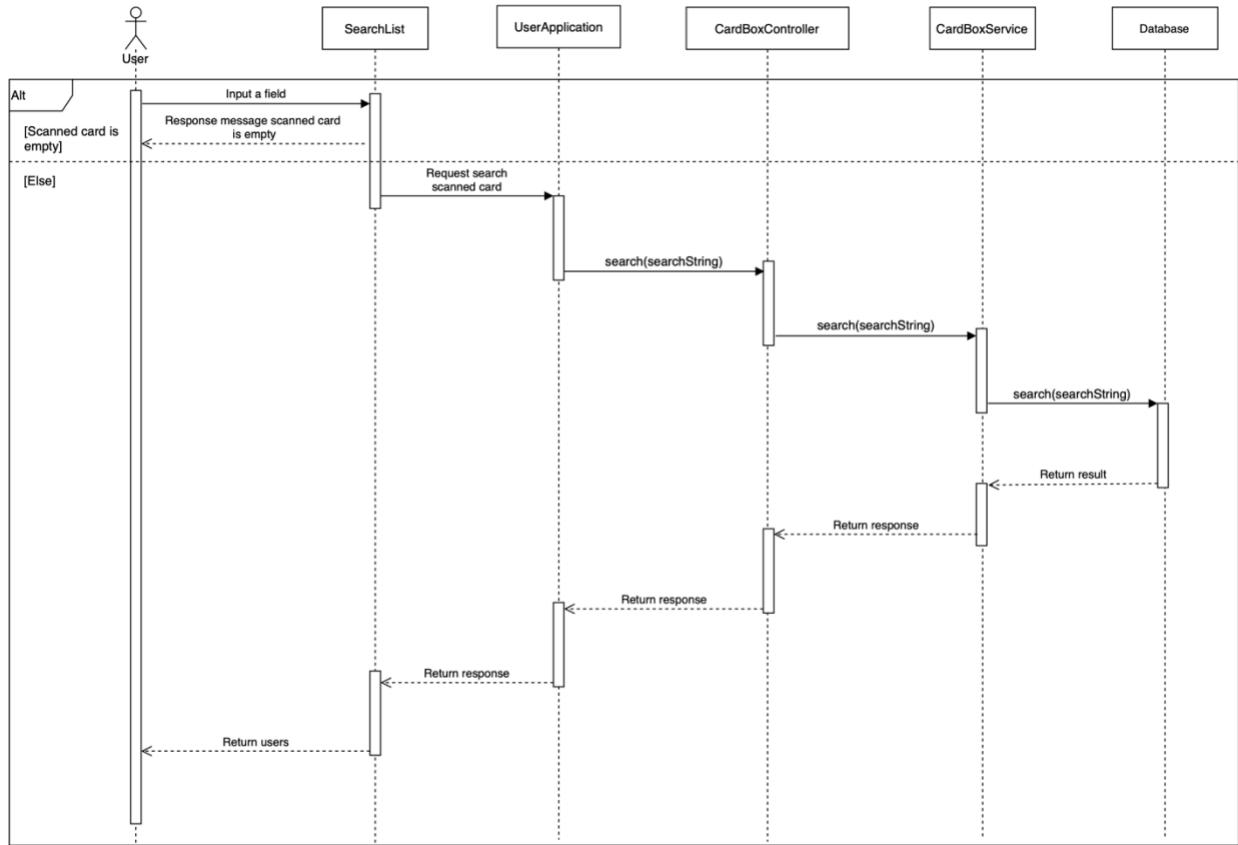


### 3.2.13 Search List

a. Class Diagram

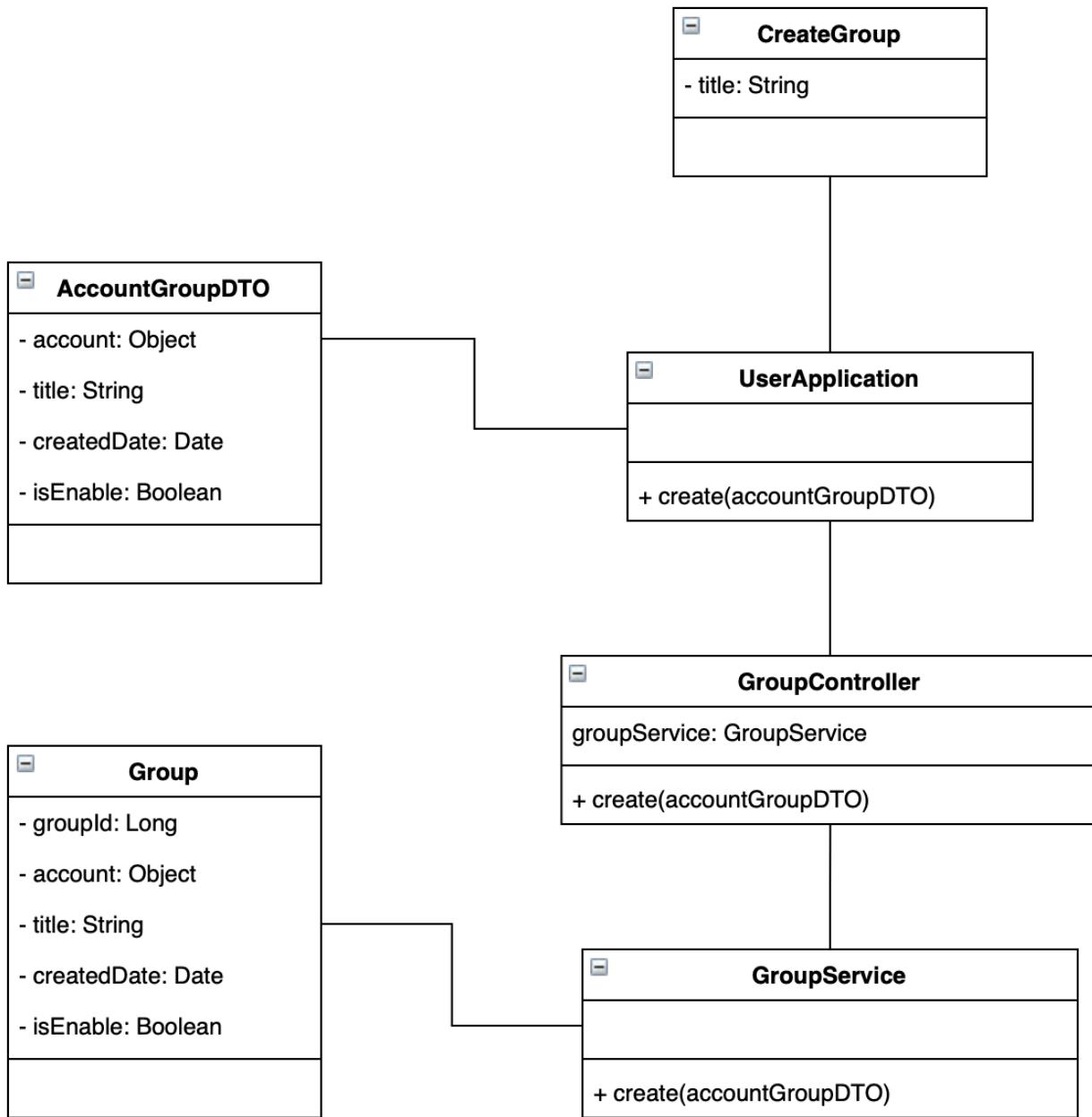


b. Sequence Diagram

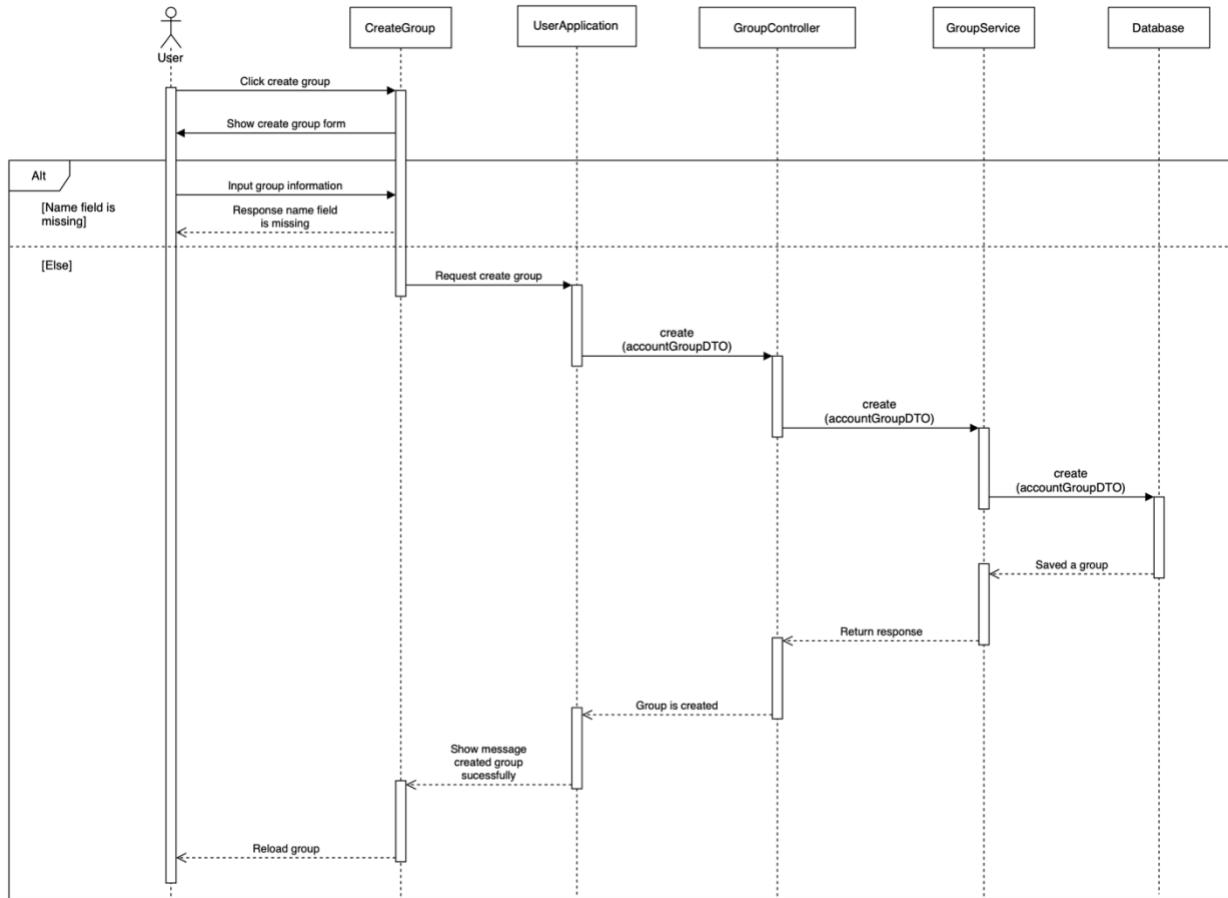


### 3.2.14 Create Group

a. Class Diagram

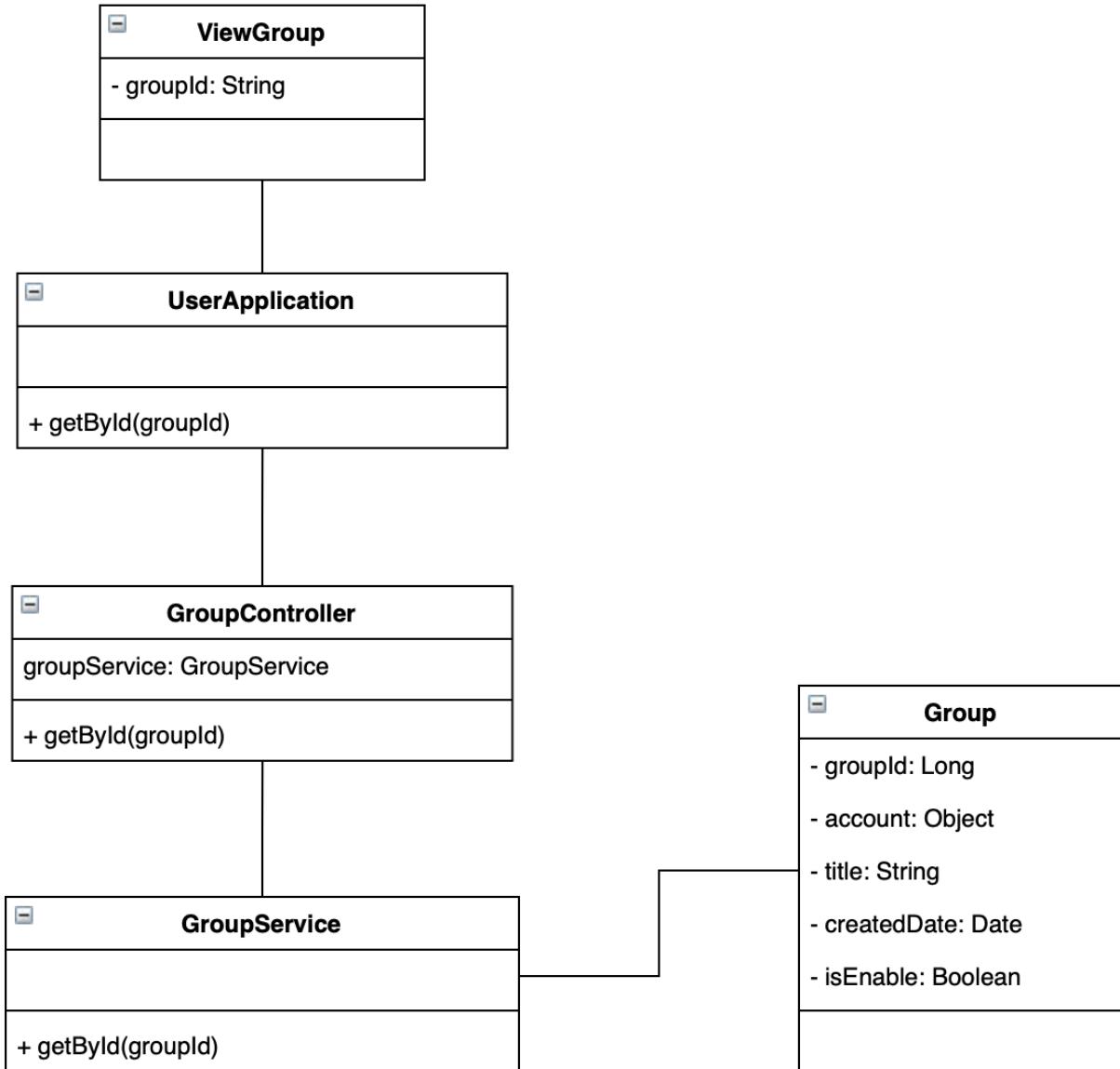


b. Sequence Diagram

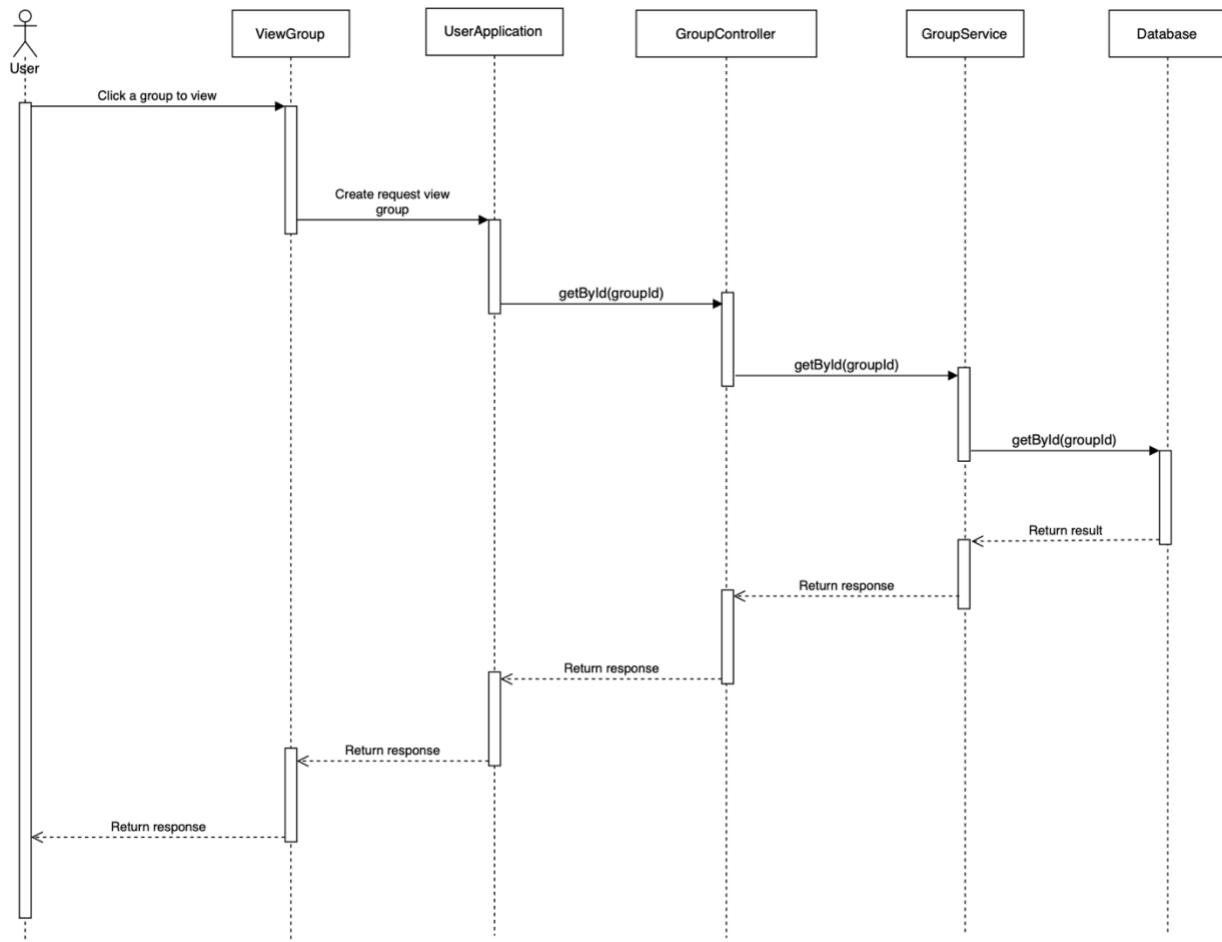


### 3.2.15 View Group

a. Class Diagram

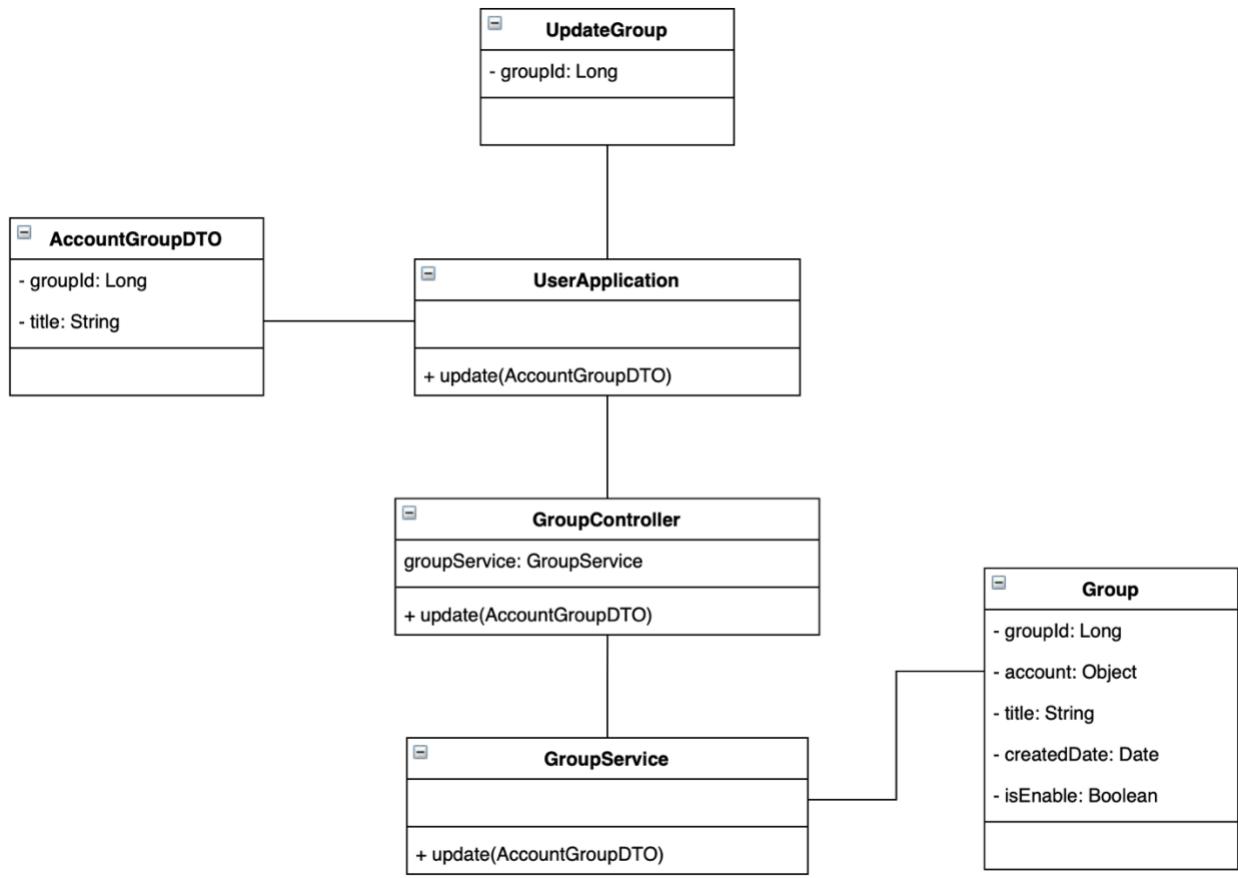


b. Sequence Diagram

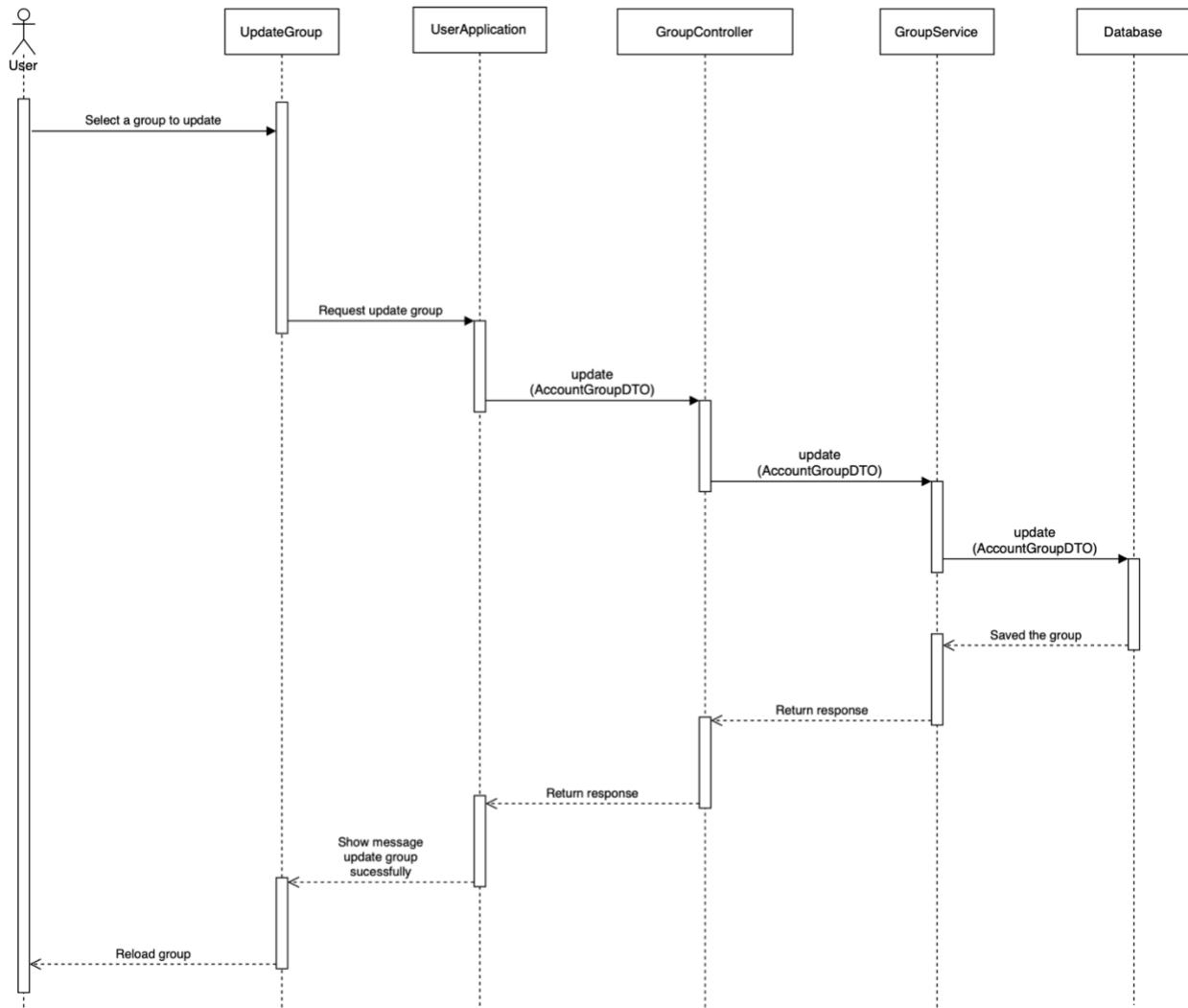


### 3.2.16 Update Group

a. Class Diagram

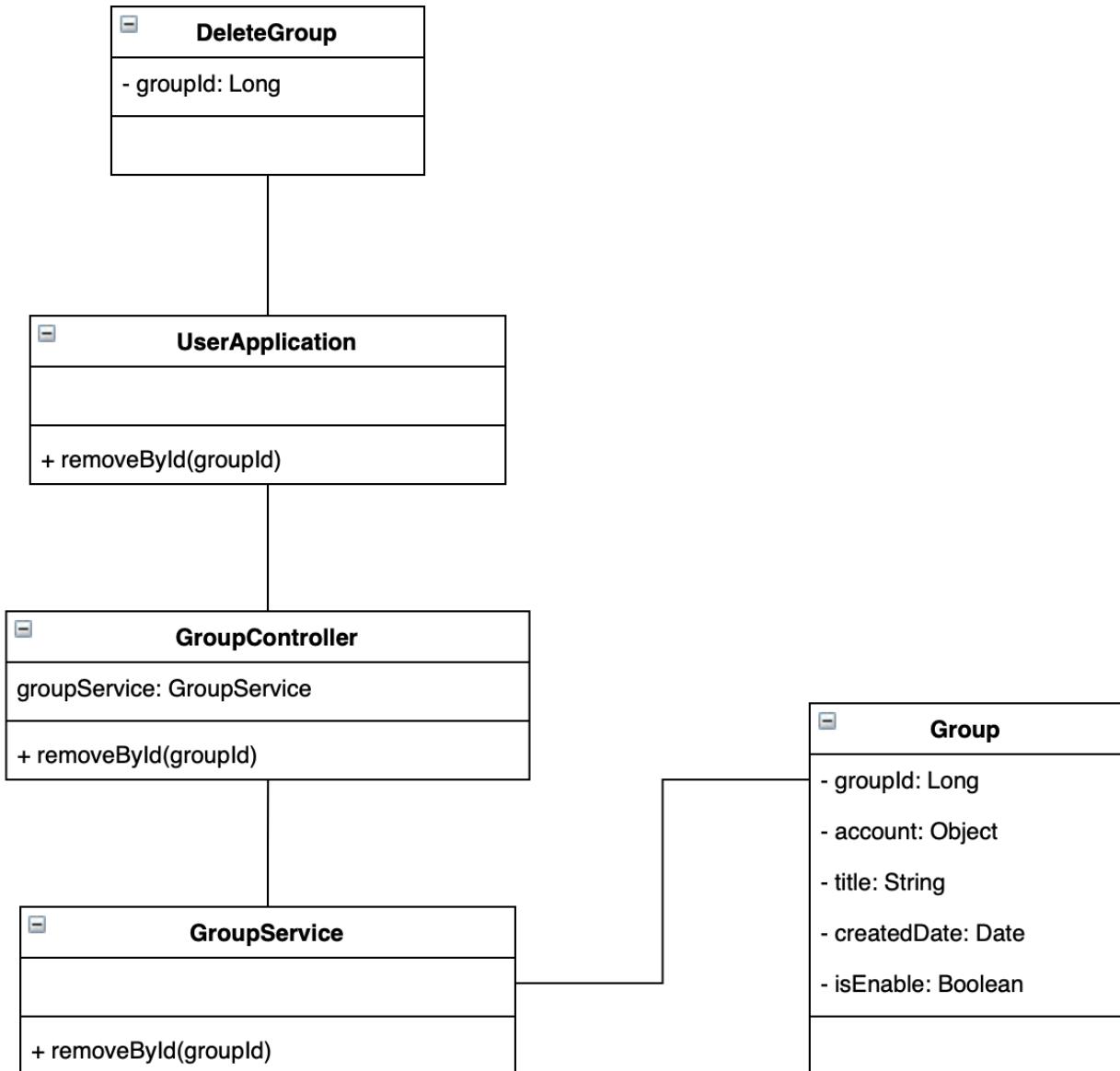


b. Sequence Diagram

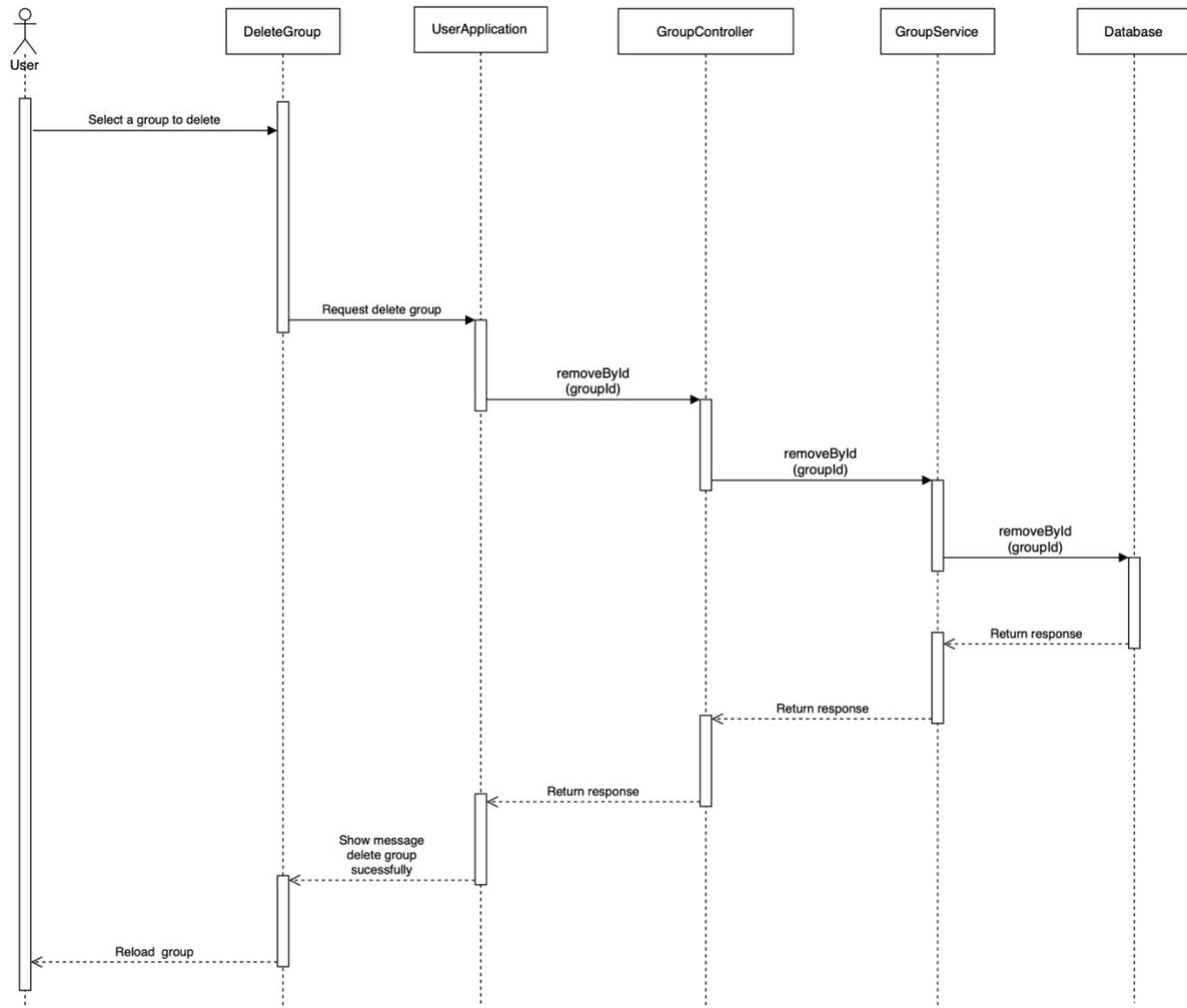


### 3.2.17 Delete Group

a. Class Diagram

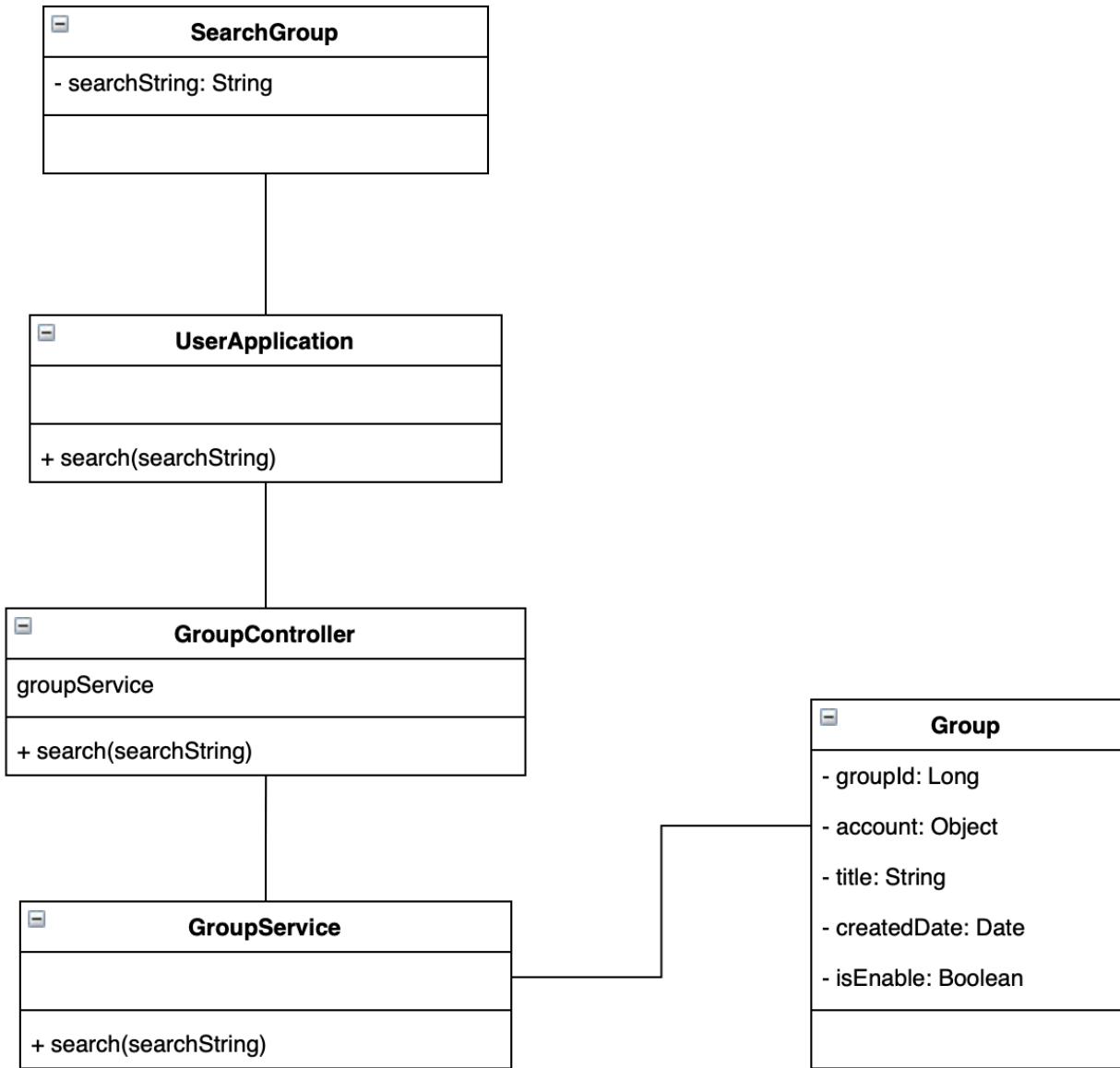


b. Sequence Diagram

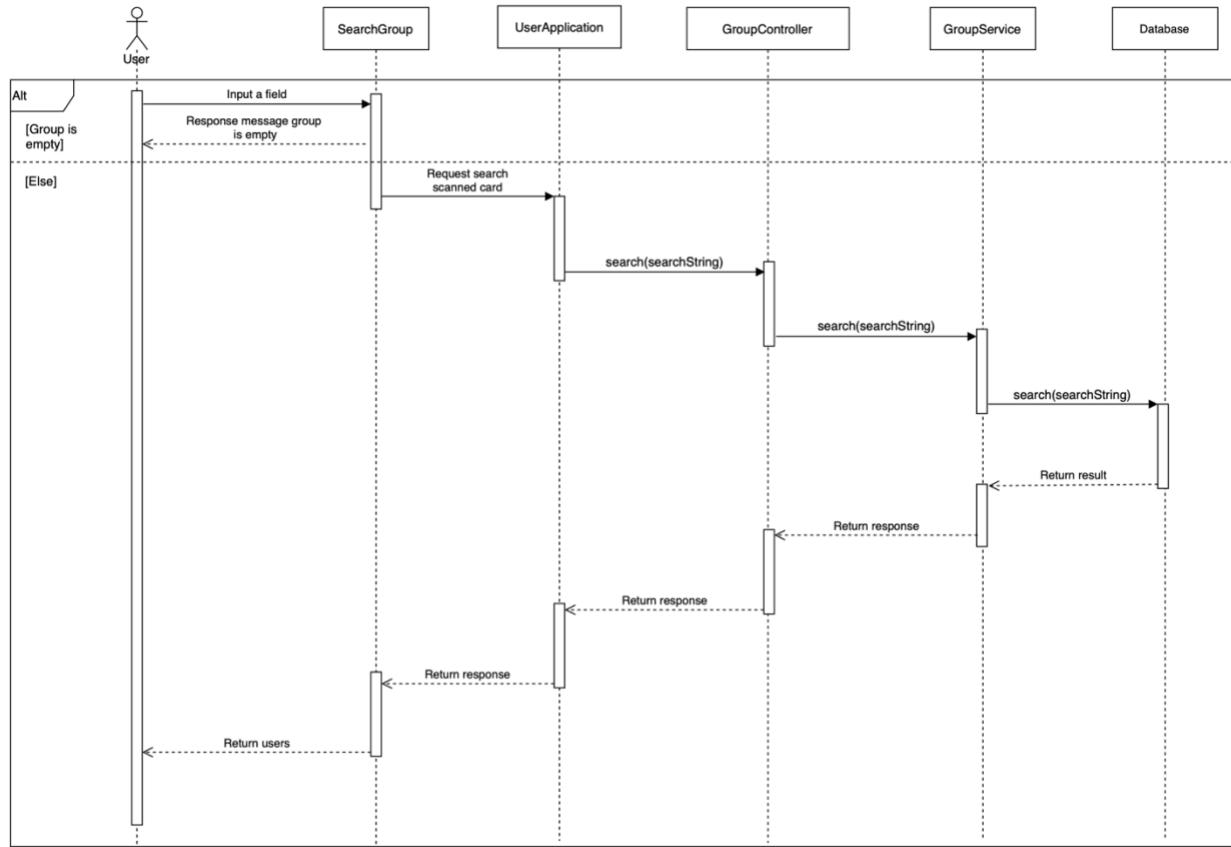


### 3.2.18 Search Group

a. Class Diagram

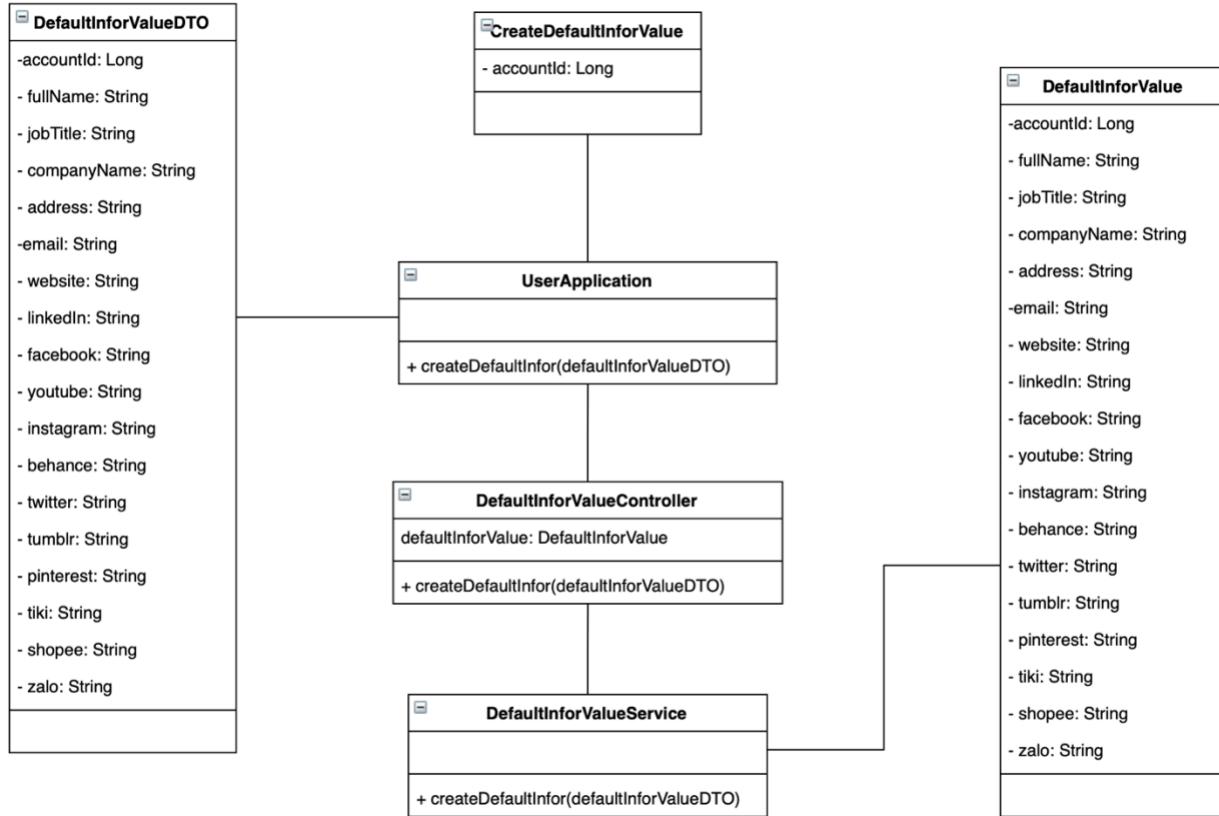


b. Sequence Diagram

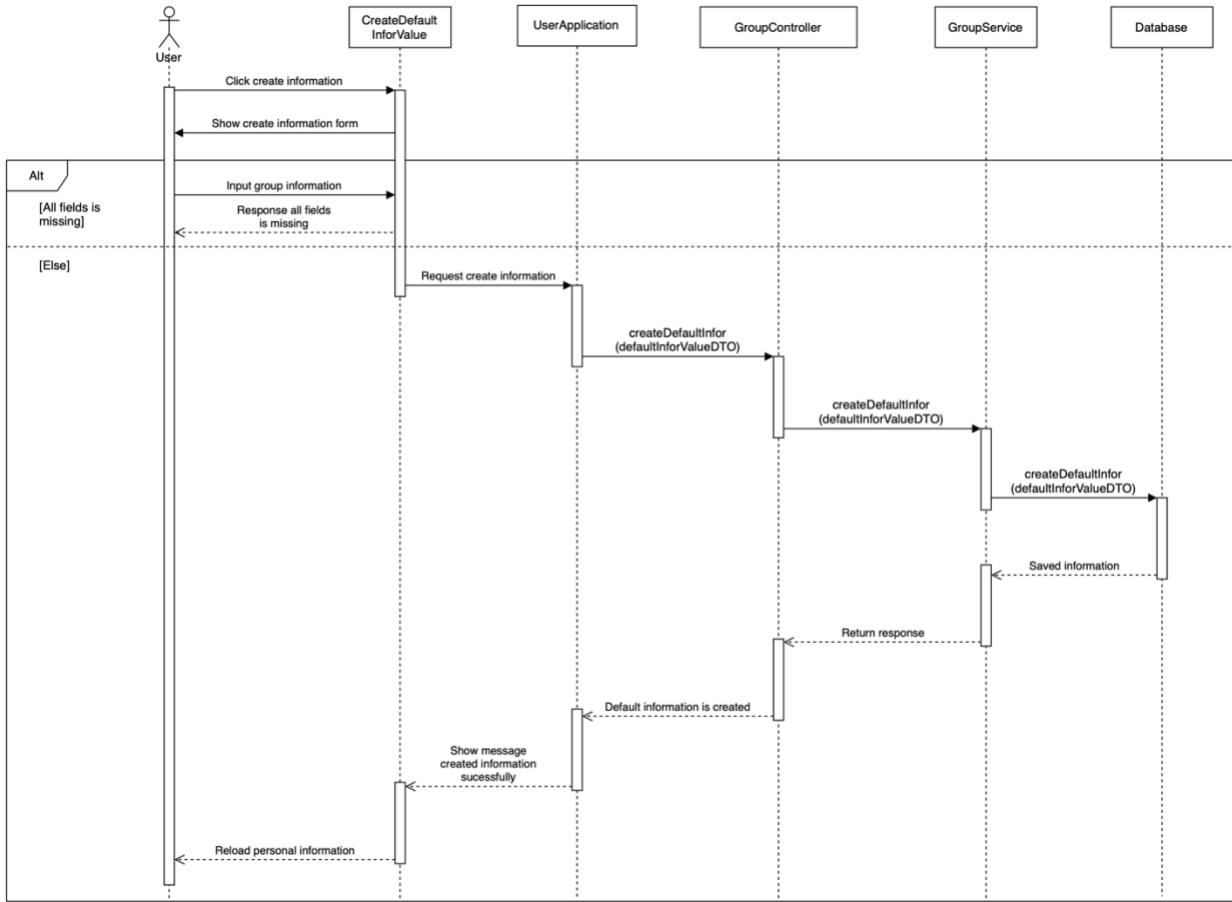


### 3.2.19 Create Information Personal

a. Class Diagram

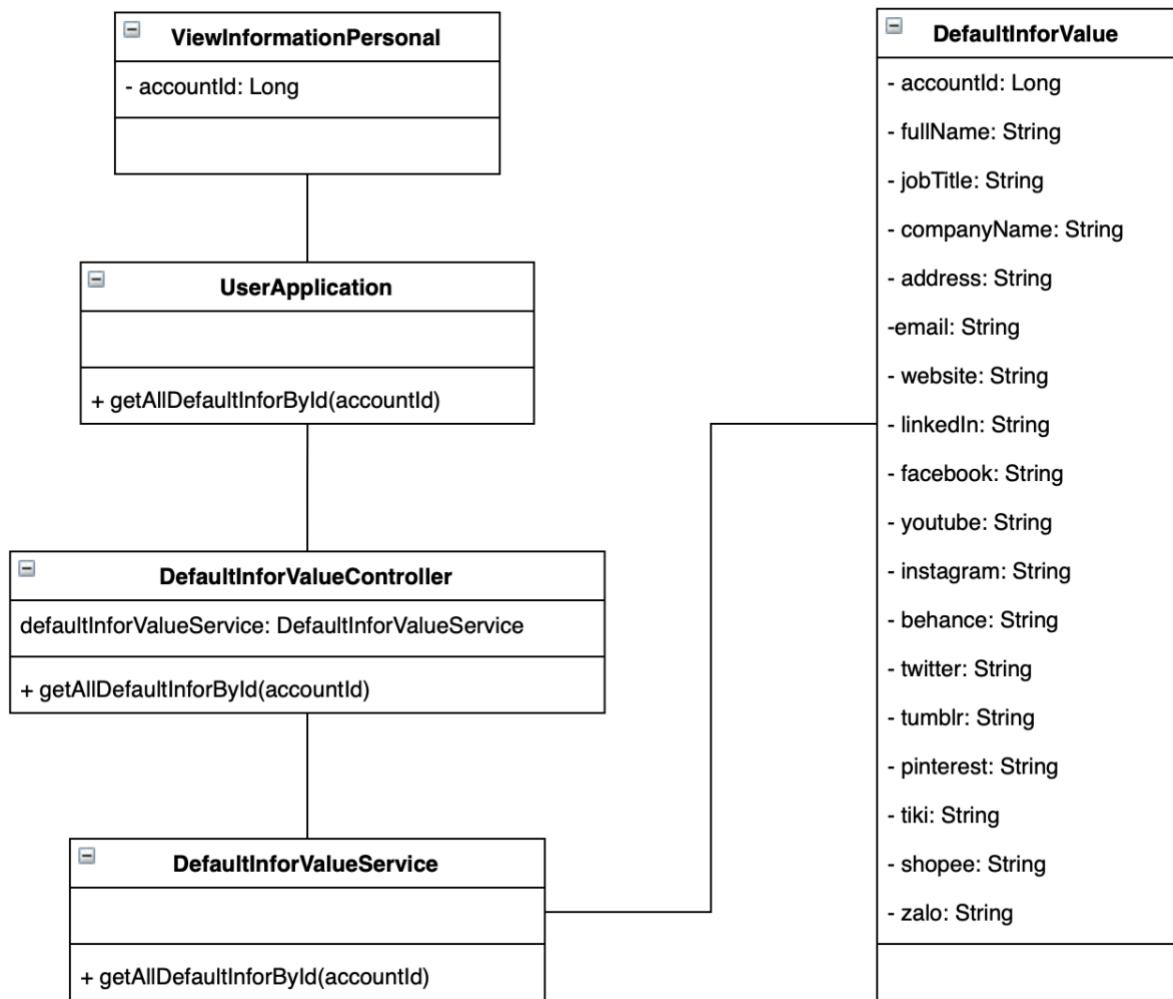


b. Sequence Diagram

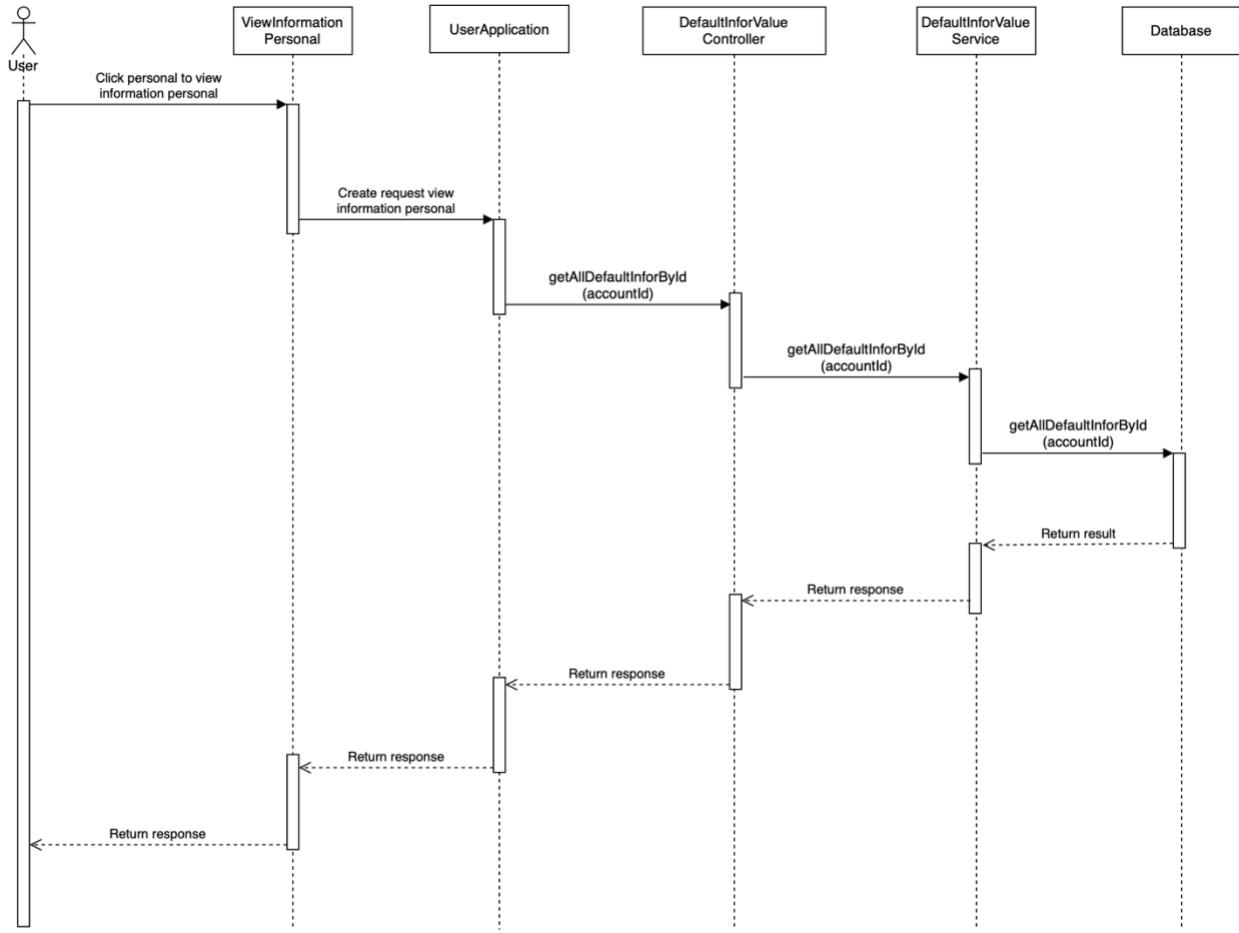


### 3.2.20 View Information Personal

a. Class Diagram

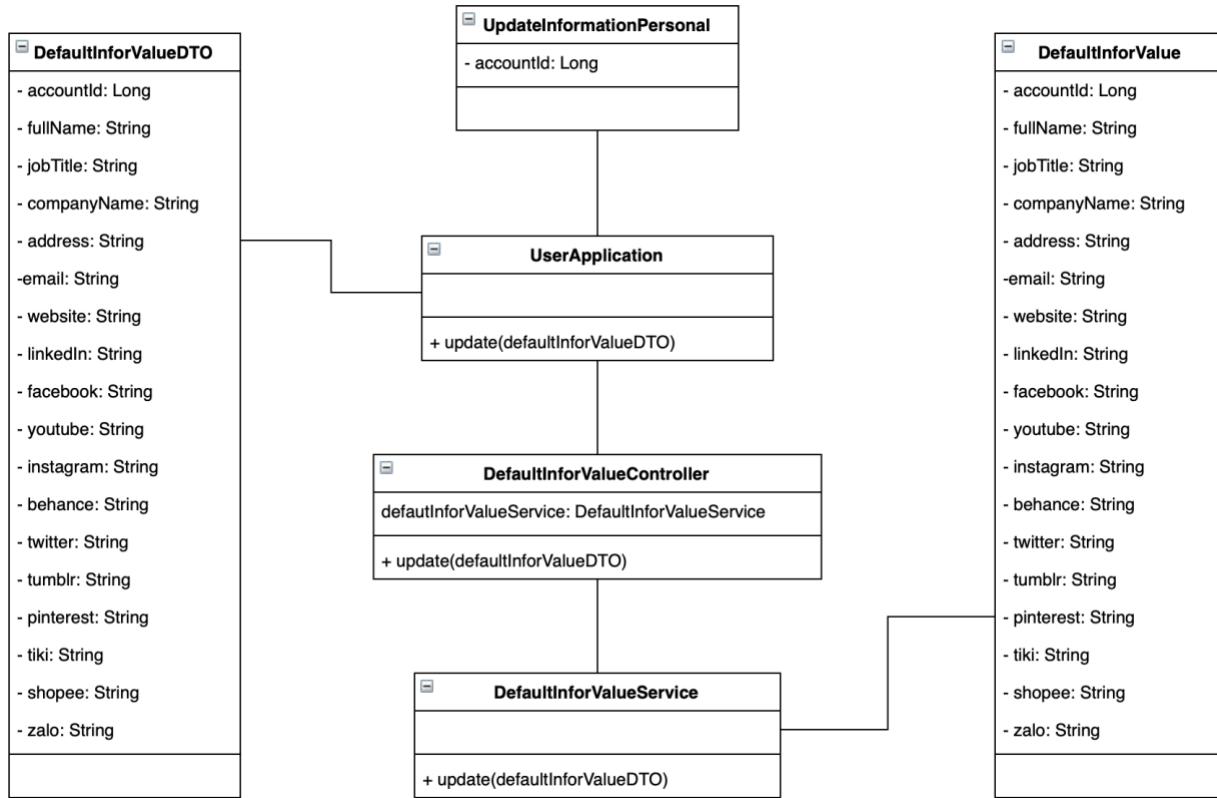


b. Sequence Diagram

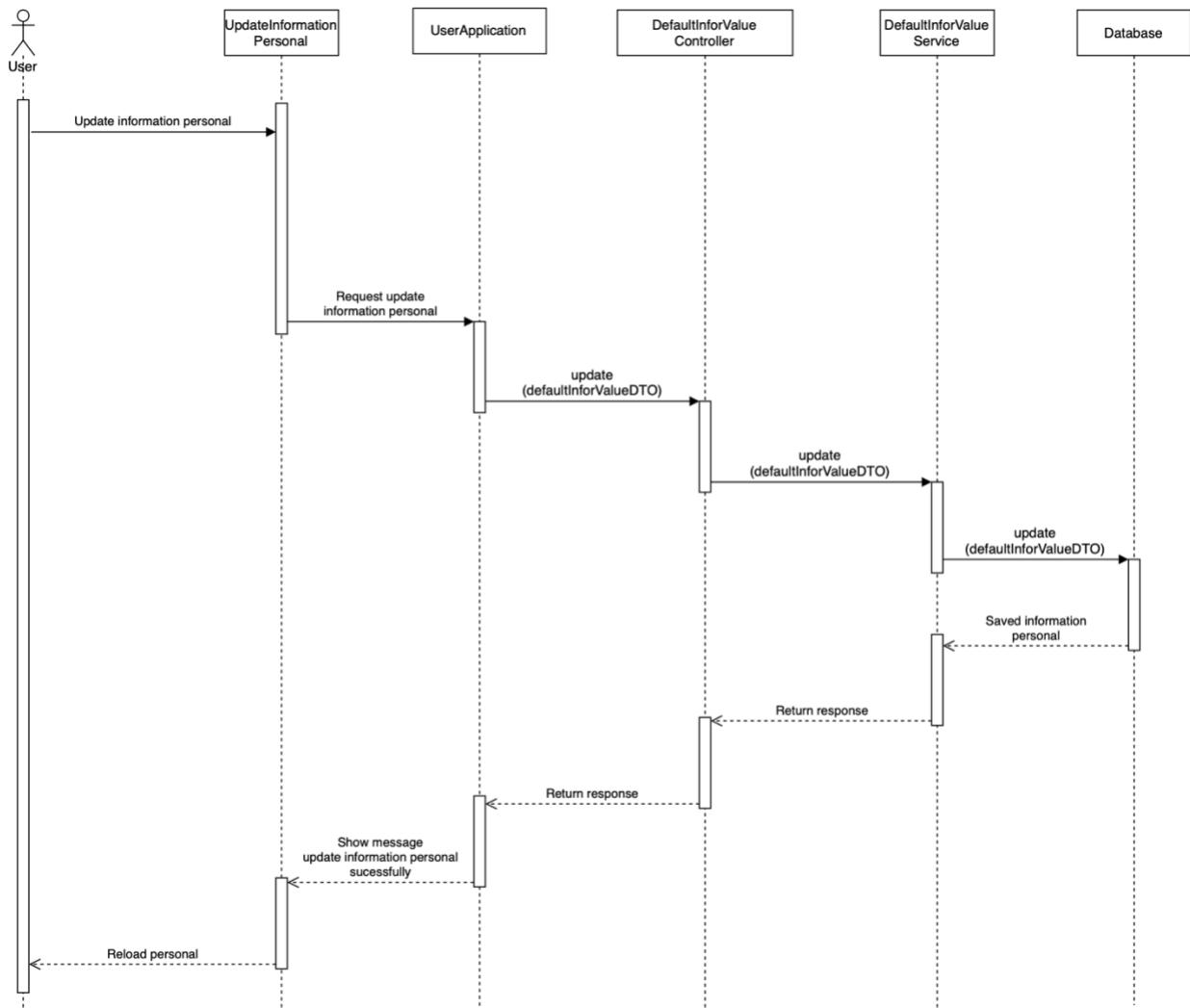


### 3.2.21 Update Information Personal

a. Class Diagram

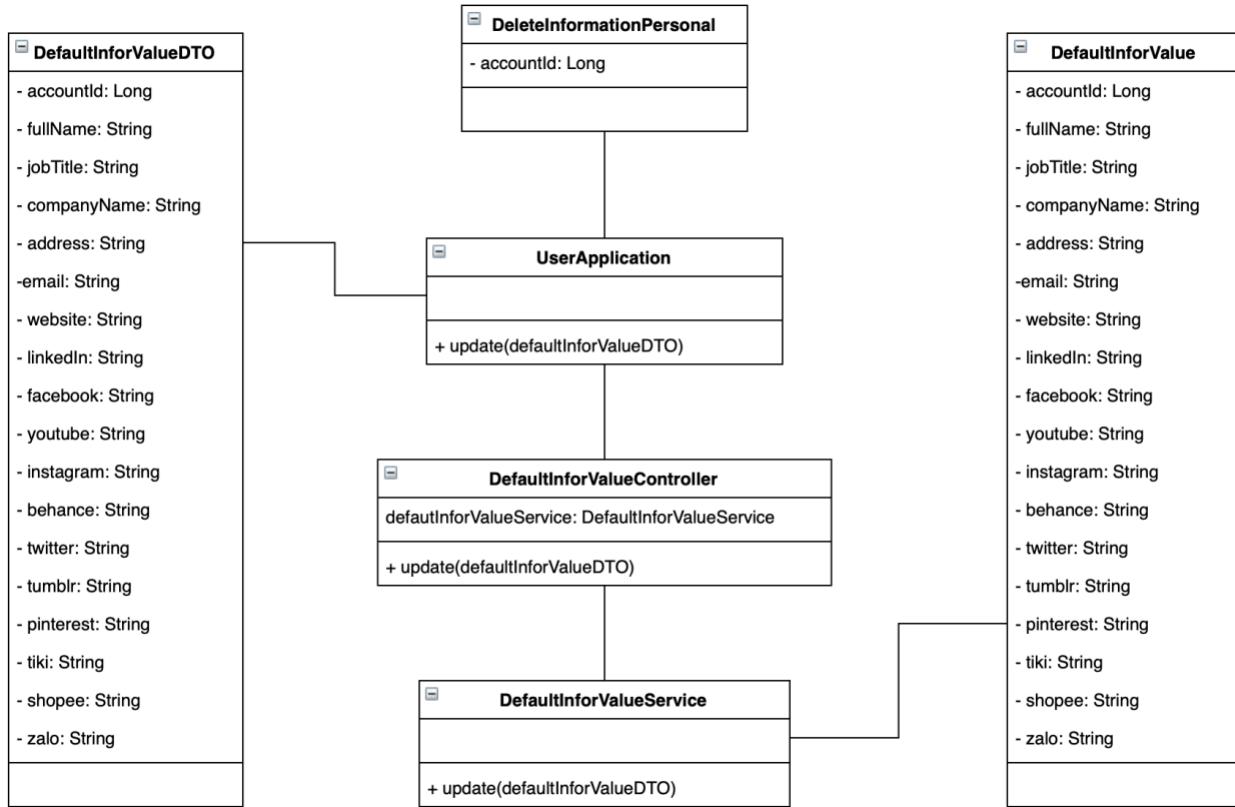


b. Sequence Diagram

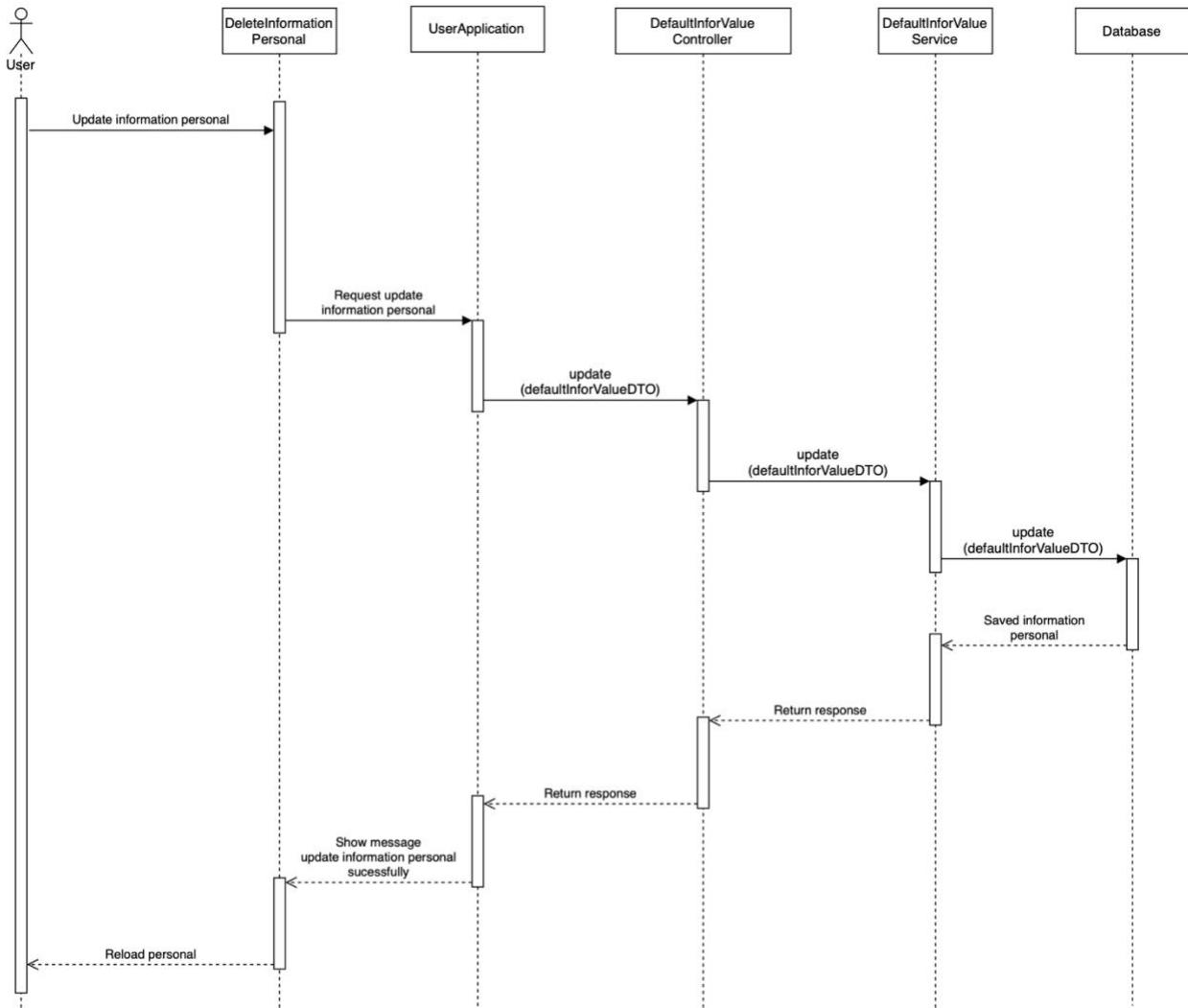


### 3.2.22 Delete Information Personal

a. Class Diagram



b. Sequence Diagram



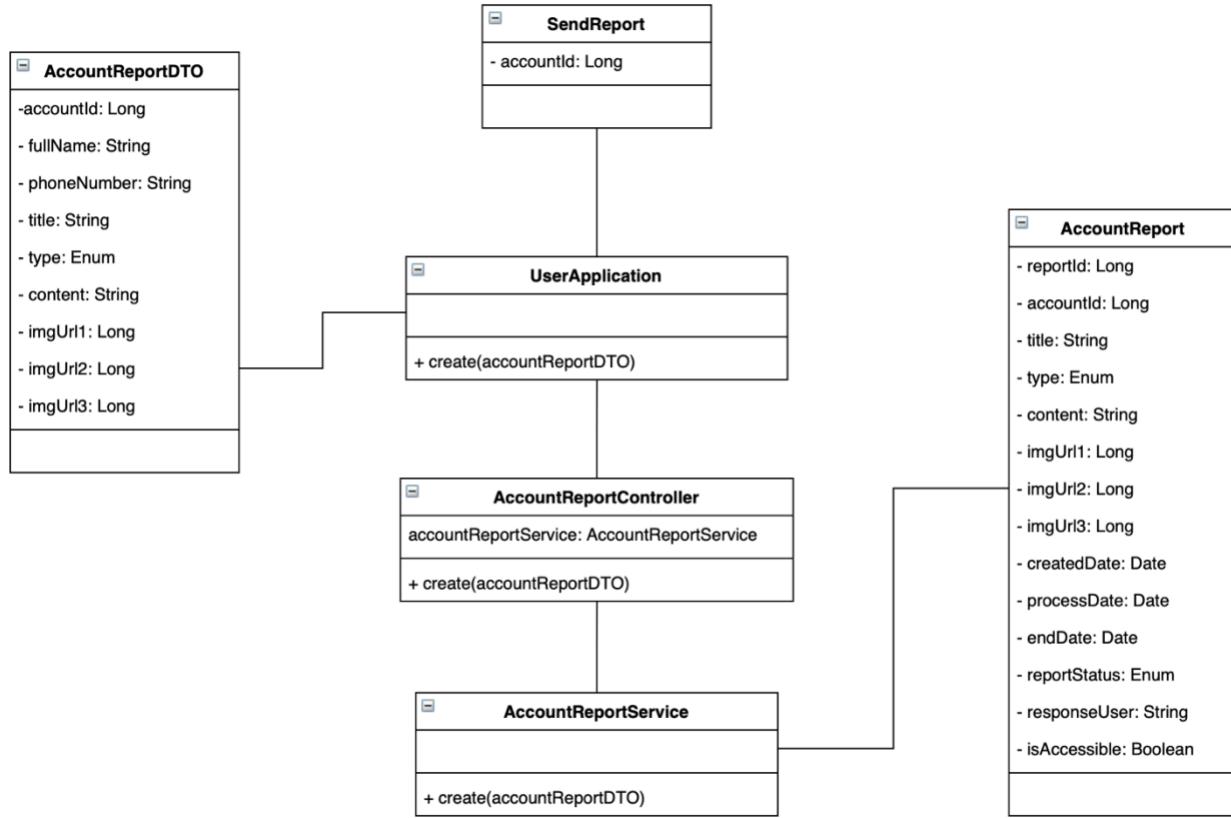
### 3.2.23 View History

a. Class Diagram

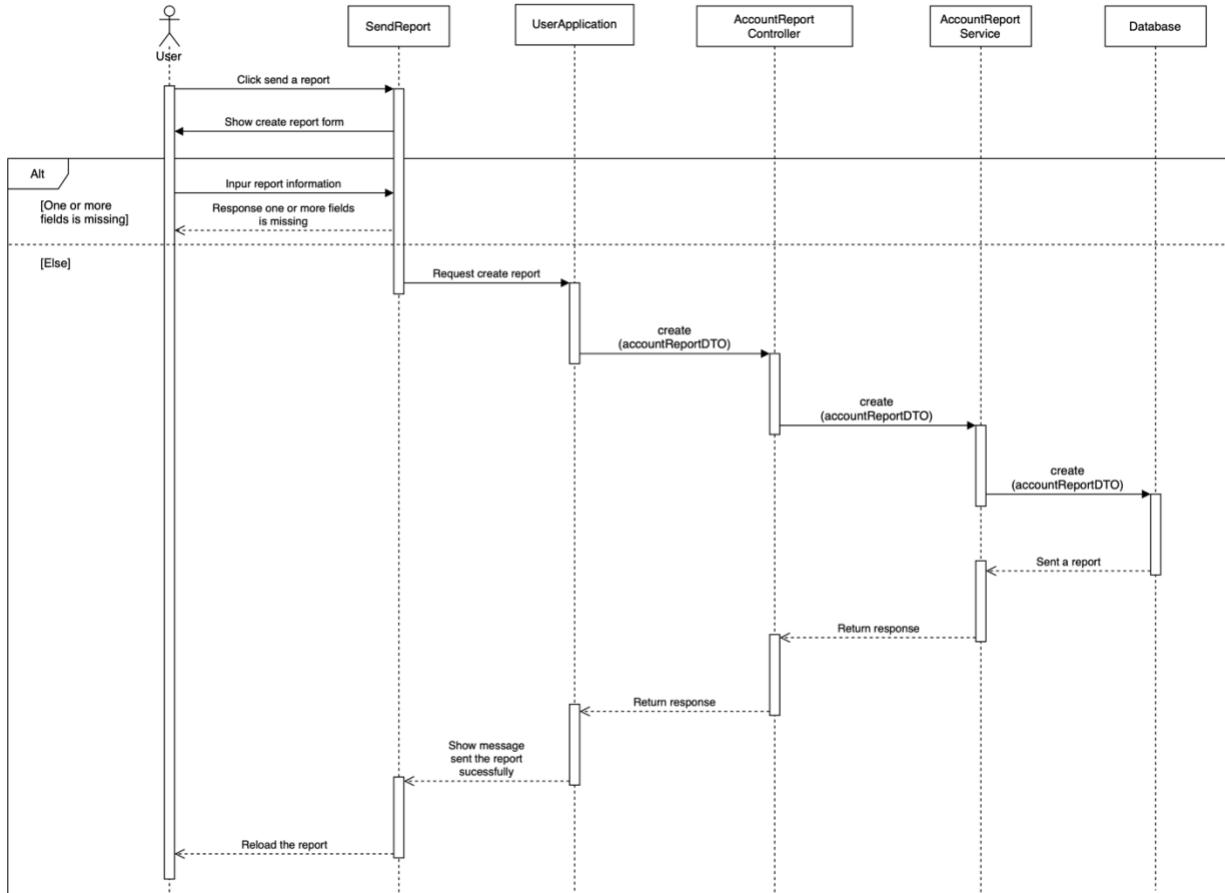
b. Sequence Diagram

### 3.2.24 Send Report

a. Class Diagram

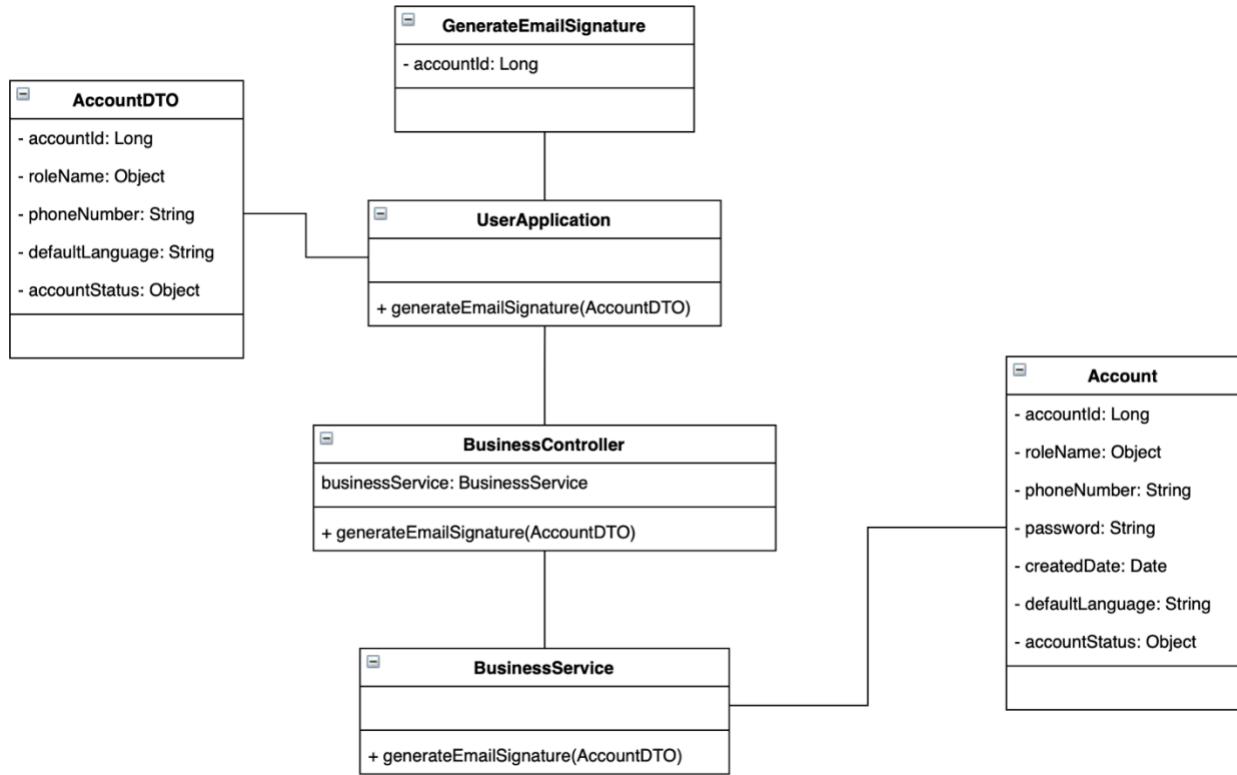


b. Sequence Diagram

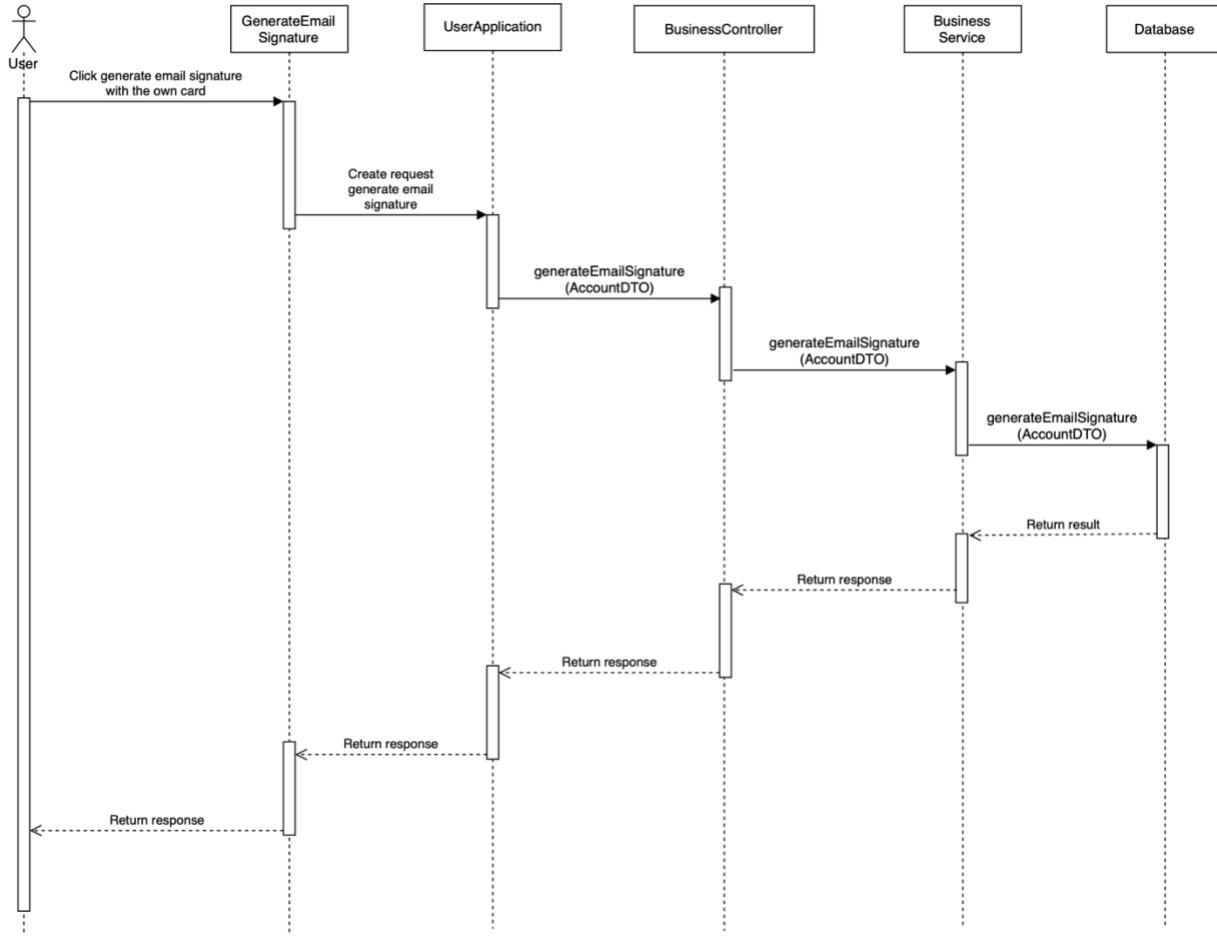


### 3.2.25 Generate Email Signature

a. Class Diagram

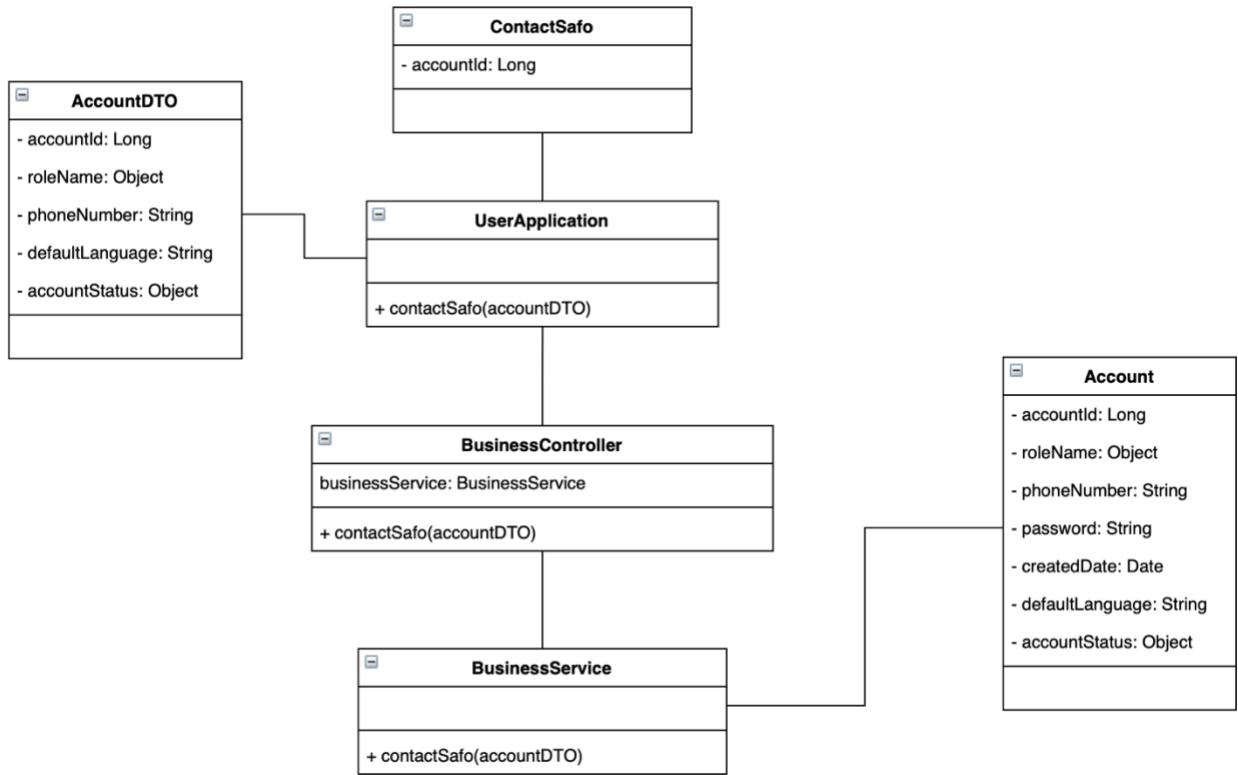


b. Sequence Diagram

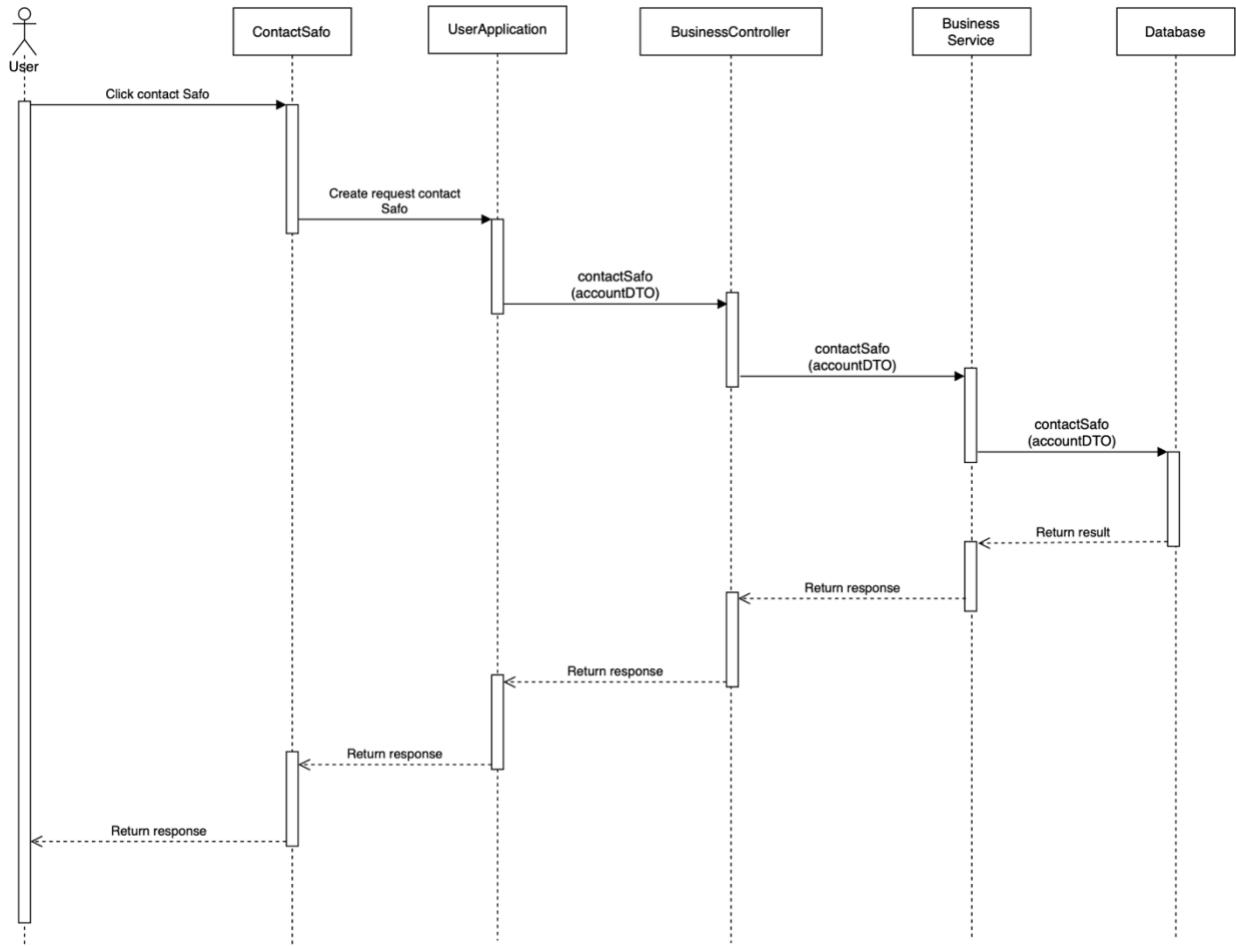


### 3.2.26 Contact Safo

a. Class Diagram

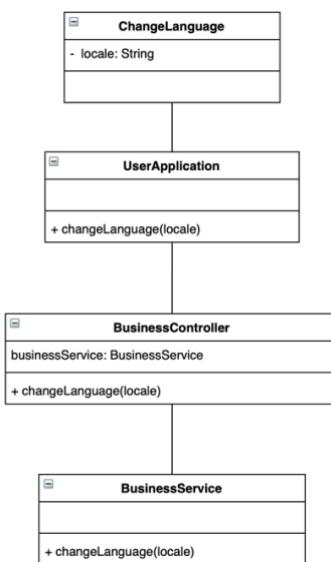


b. Sequence Diagram

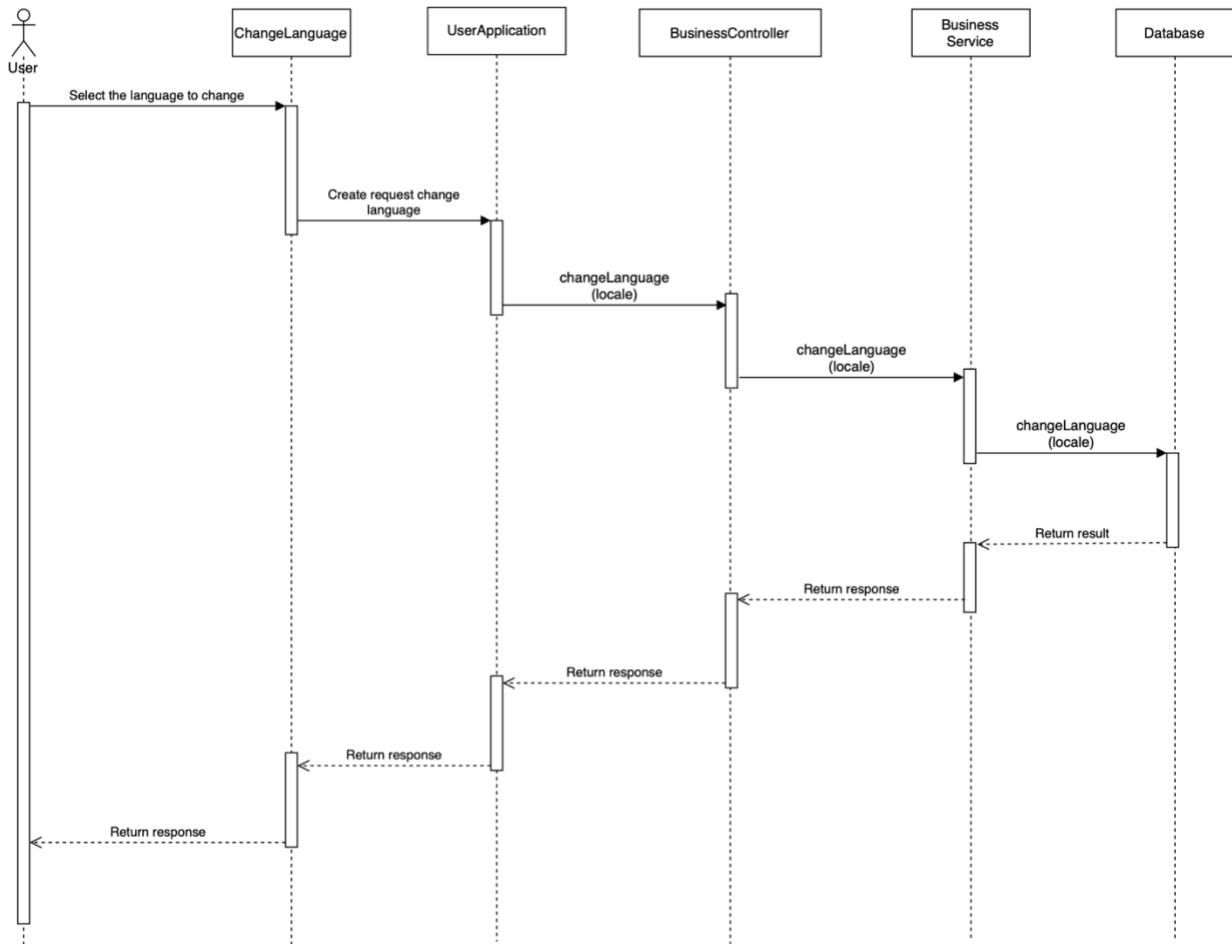


### 3.2.27 Change Language

a. Class Diagram

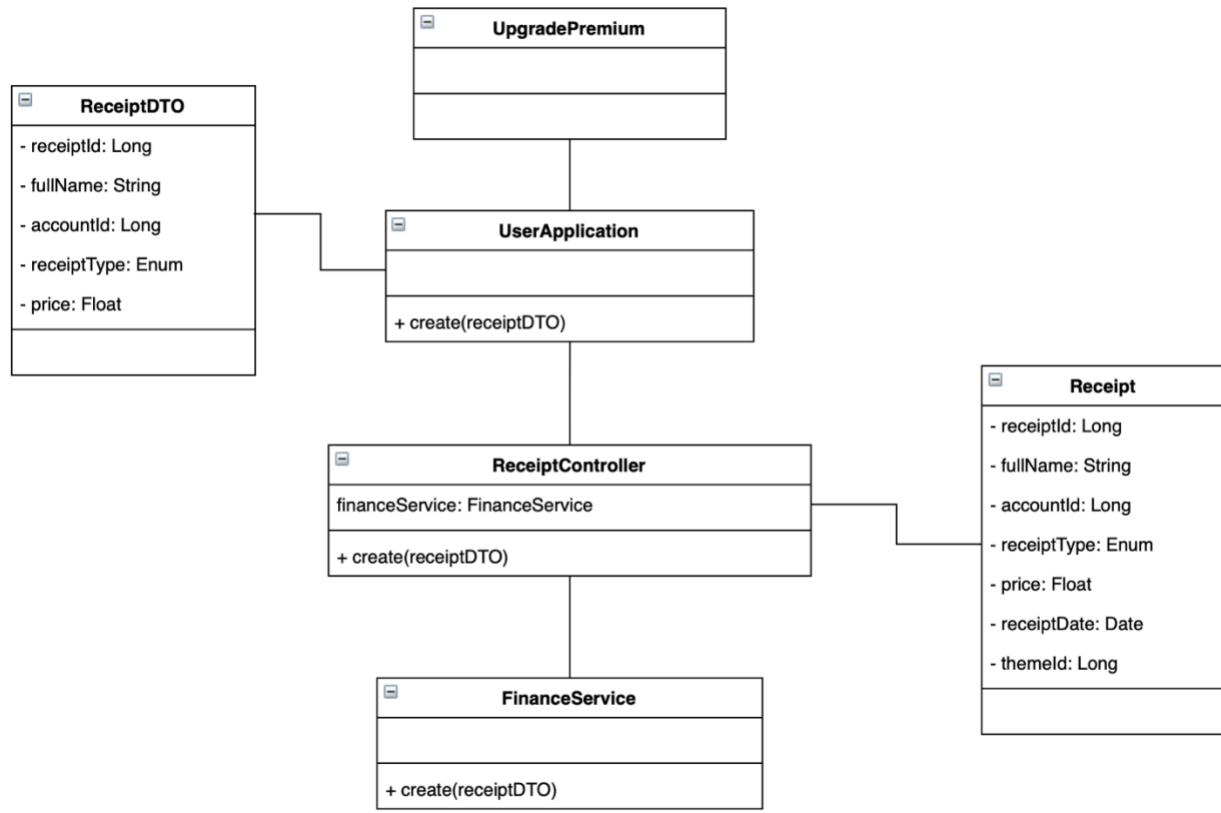


### b. Sequence Diagram

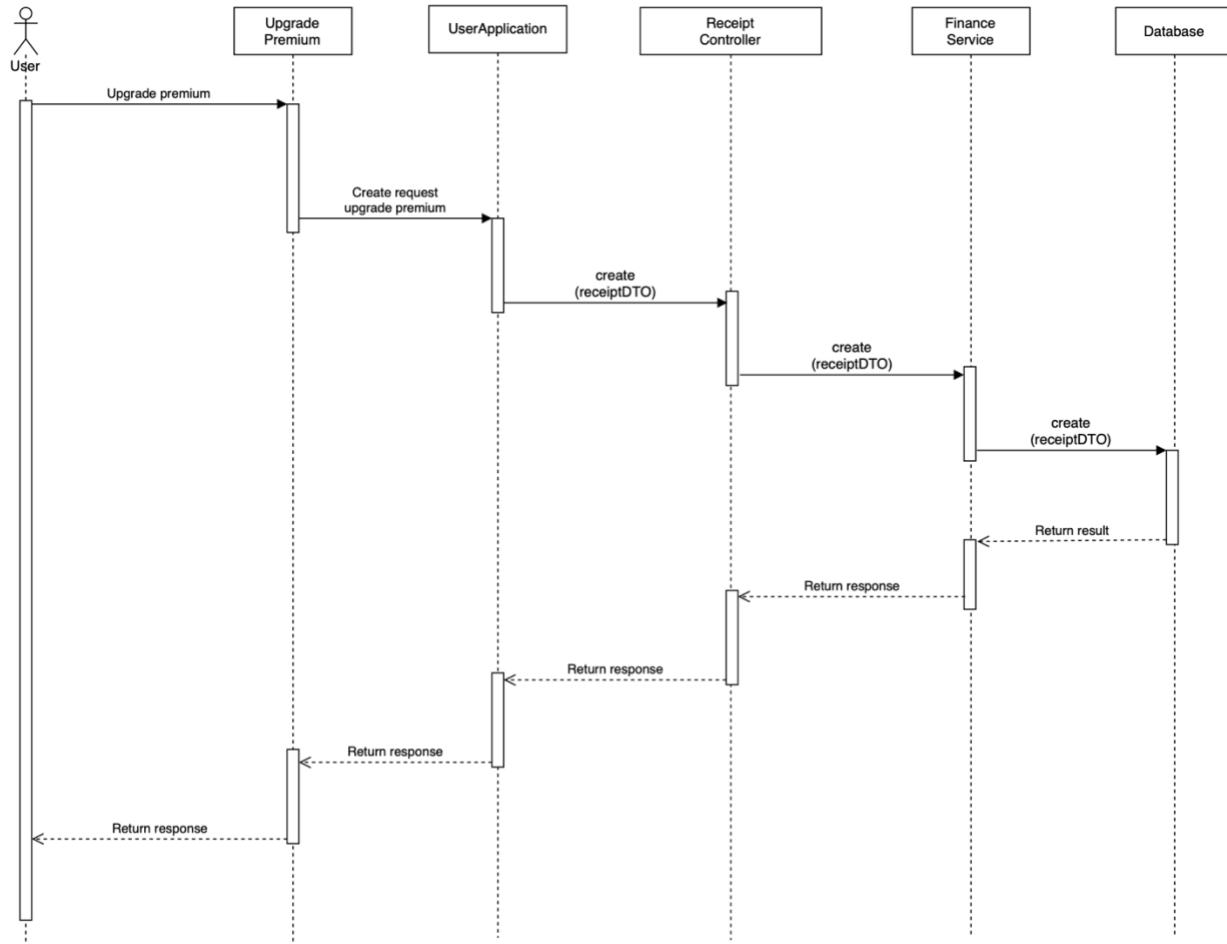


### 3.2.28 Upgrade Premium

#### a. Class Diagram

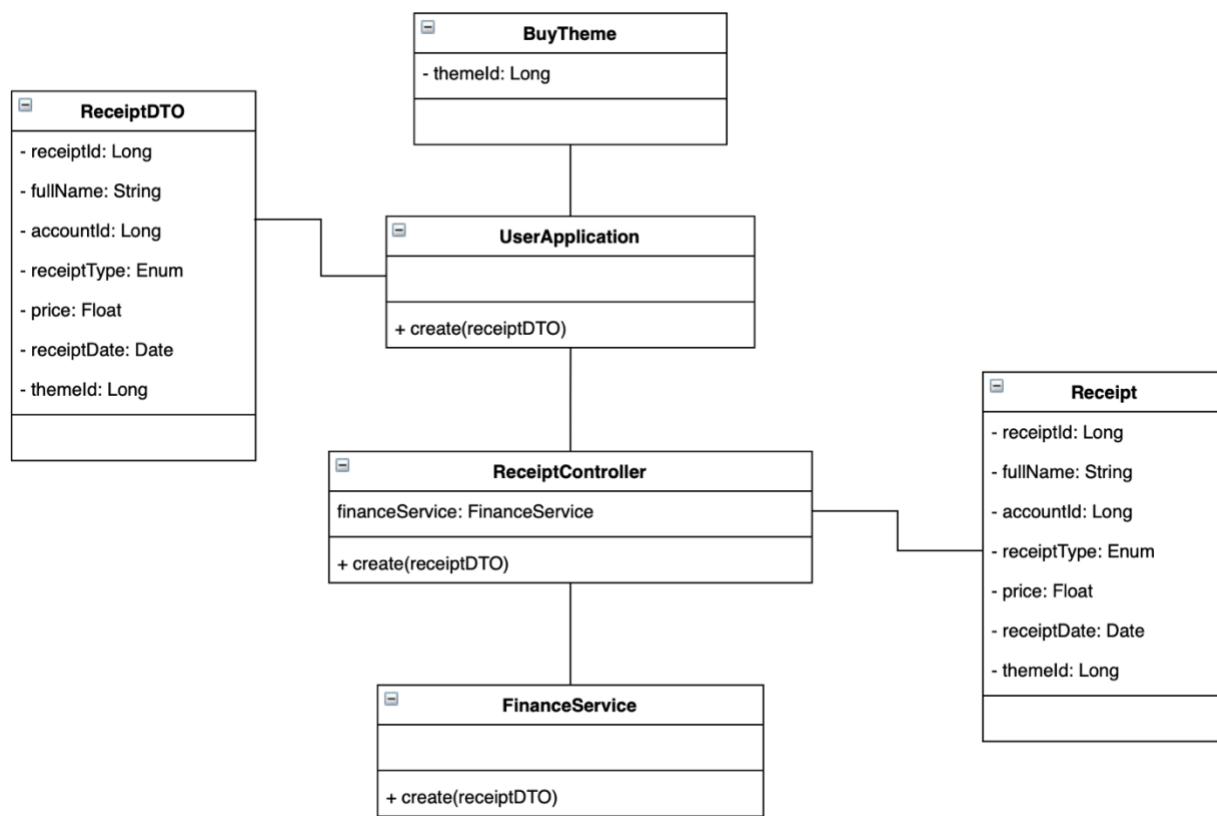


b. Sequence Diagram

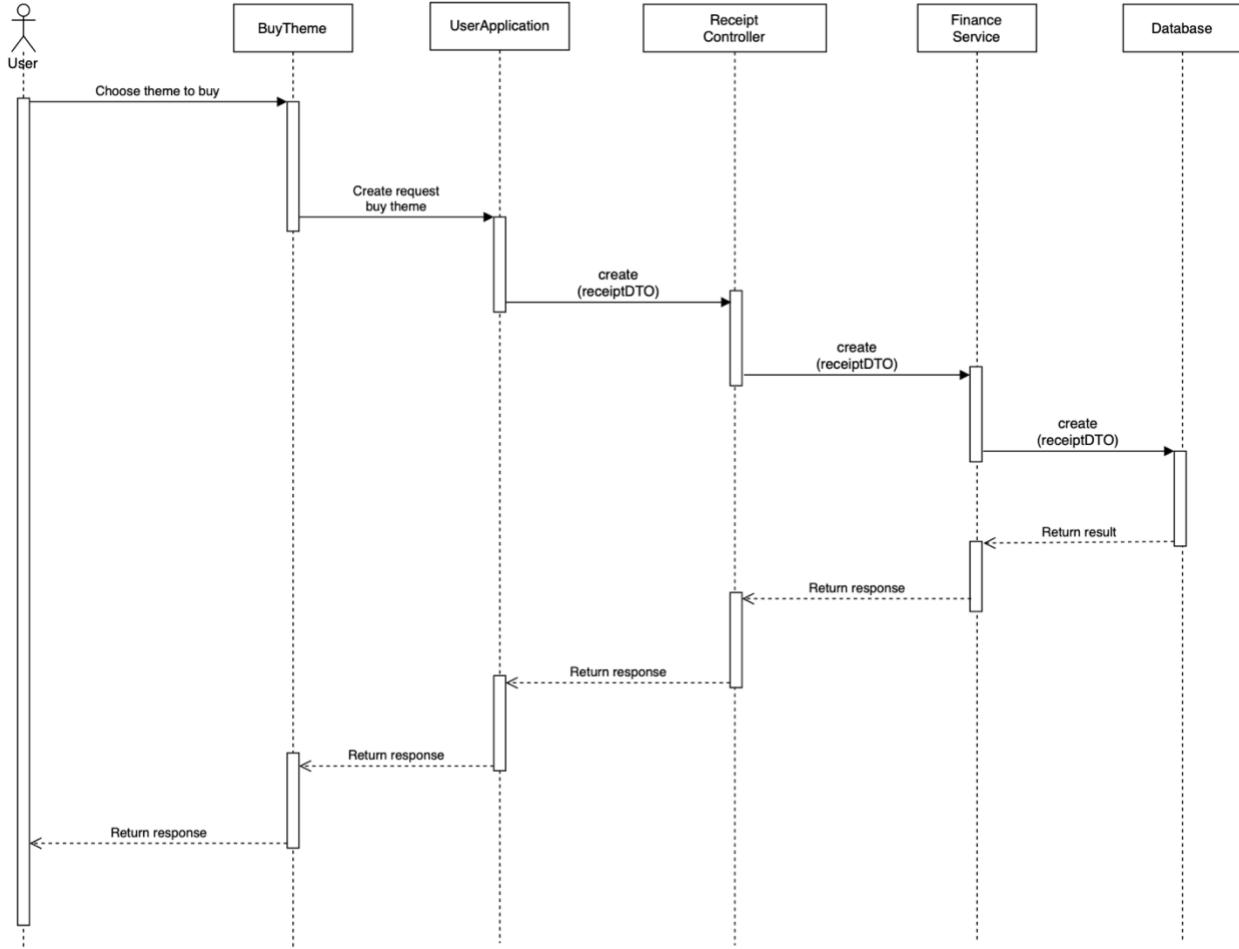


### 3.2.29 Buy Theme

a. Class Diagram



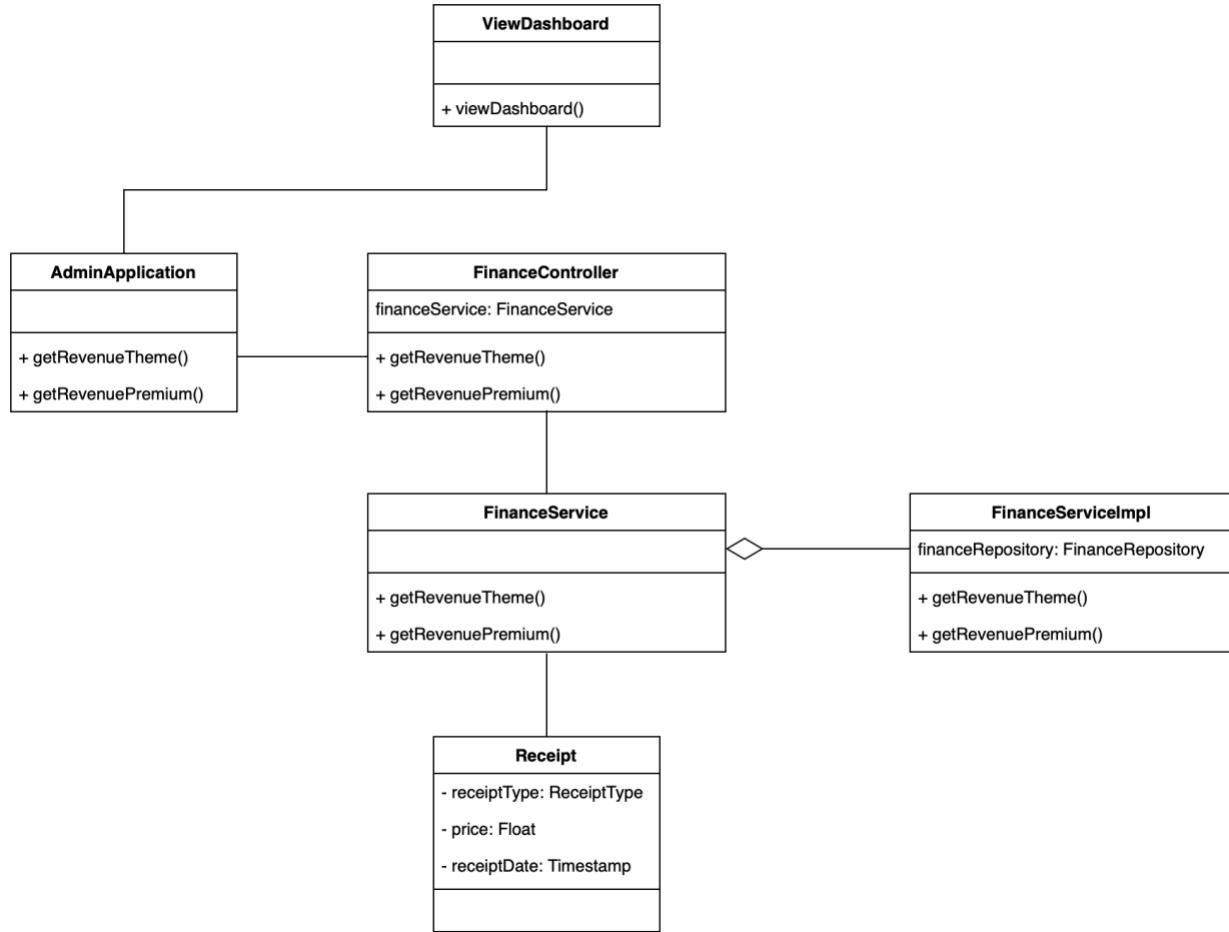
b. Sequence Diagram



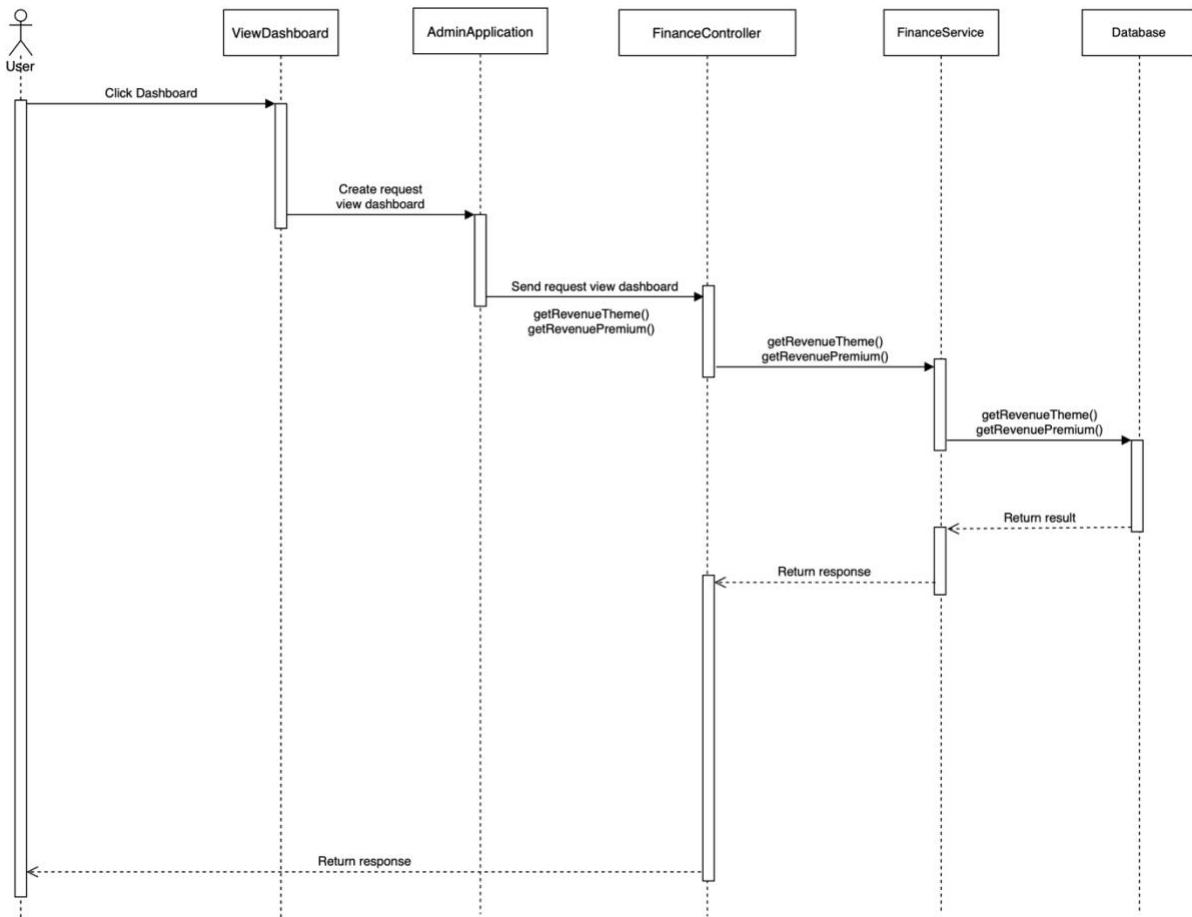
### 3.3 Admin Management

#### 3.3.1 View Dashboard

a. Class diagram

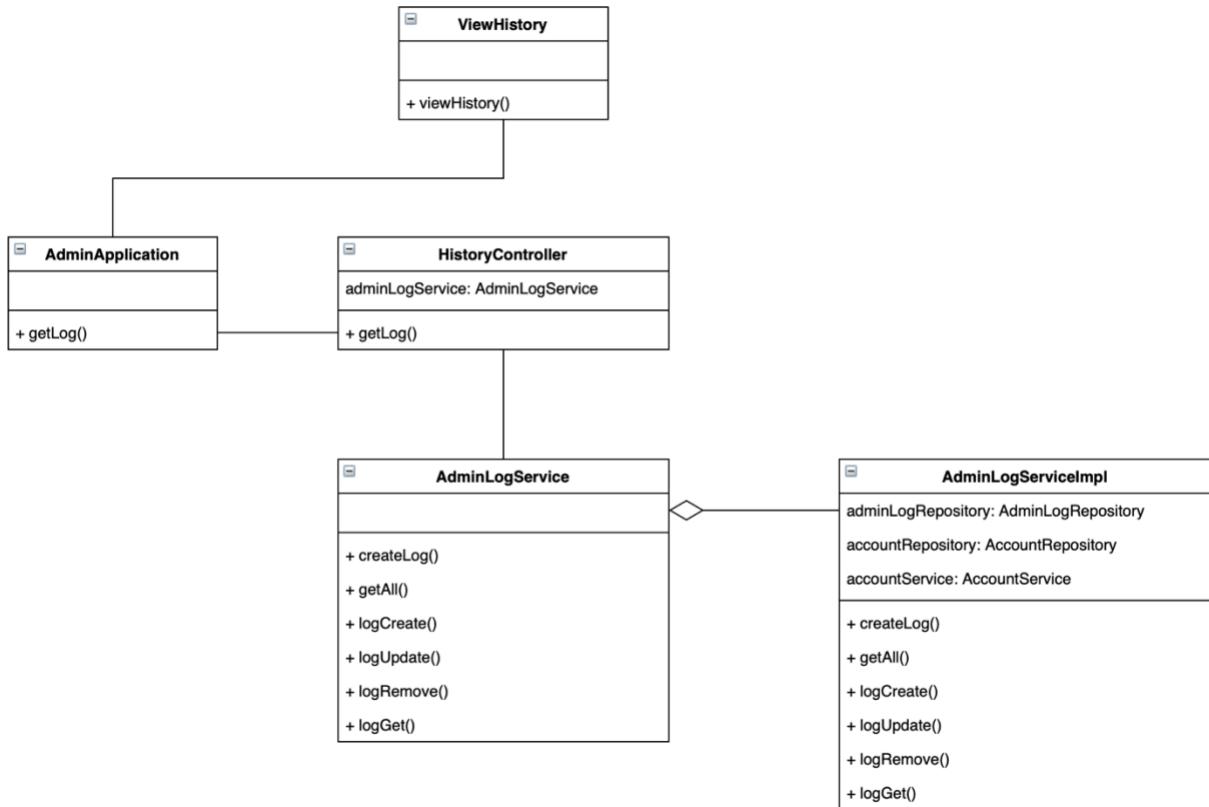


b. Sequence diagram

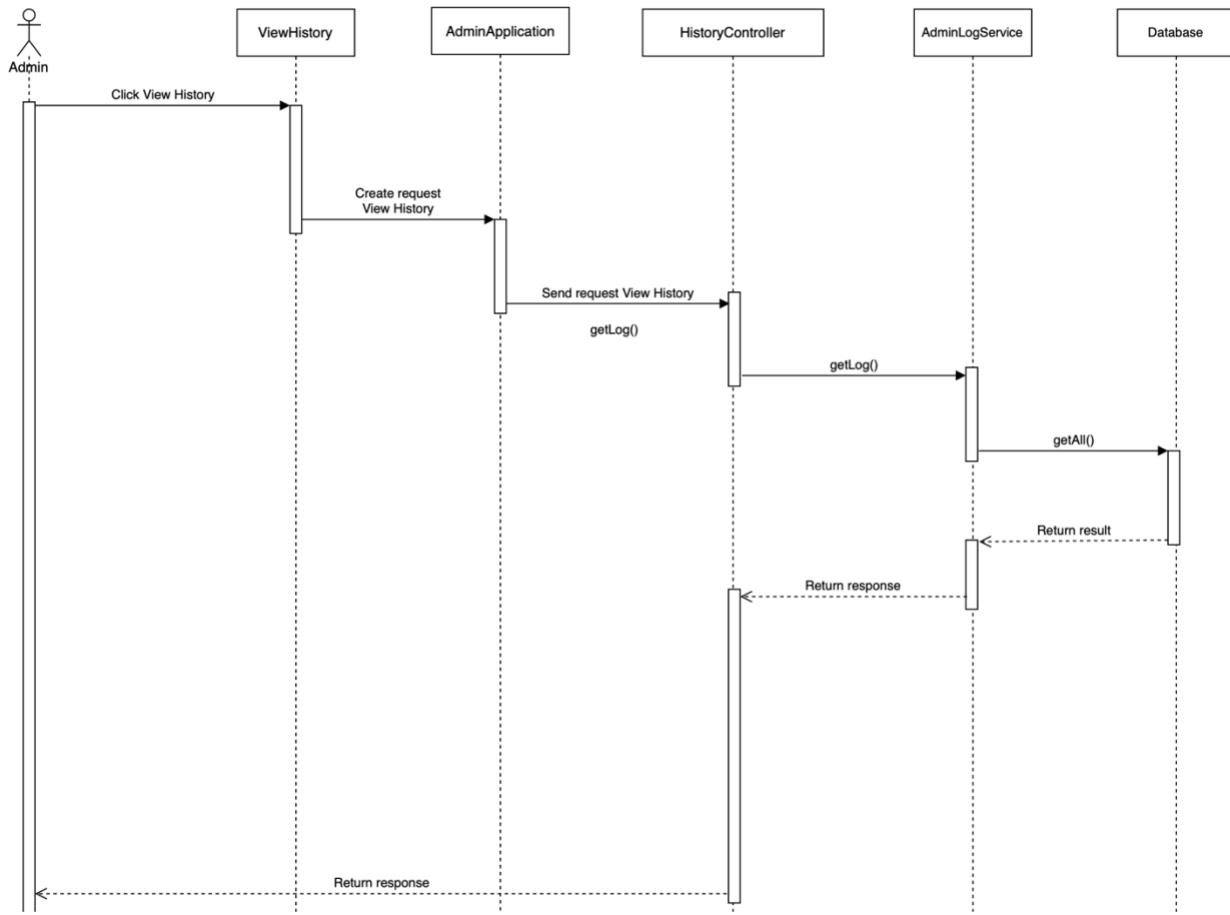


### 3.3.2 View History

a. Class Diagram

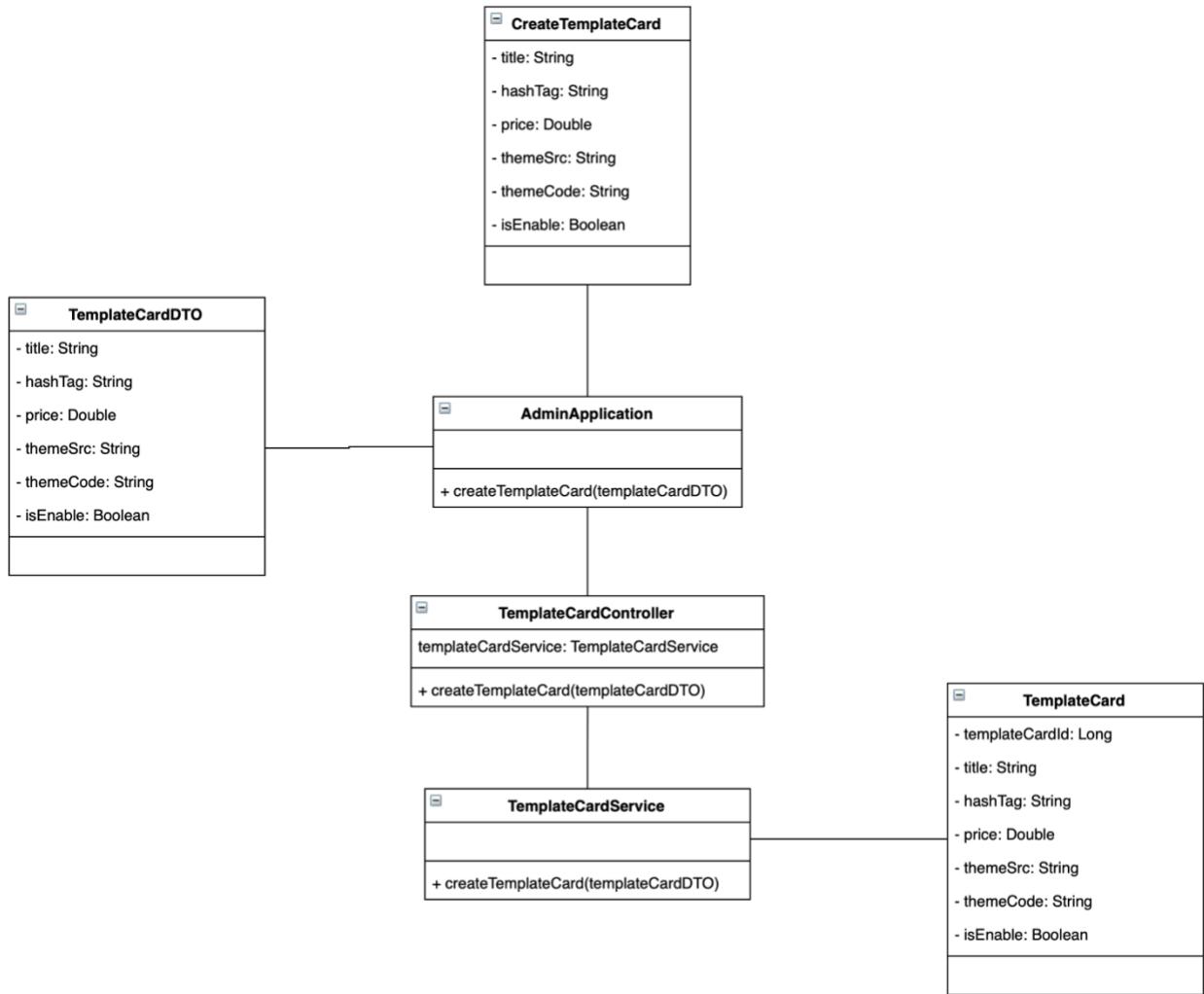


b. Sequence Diagram

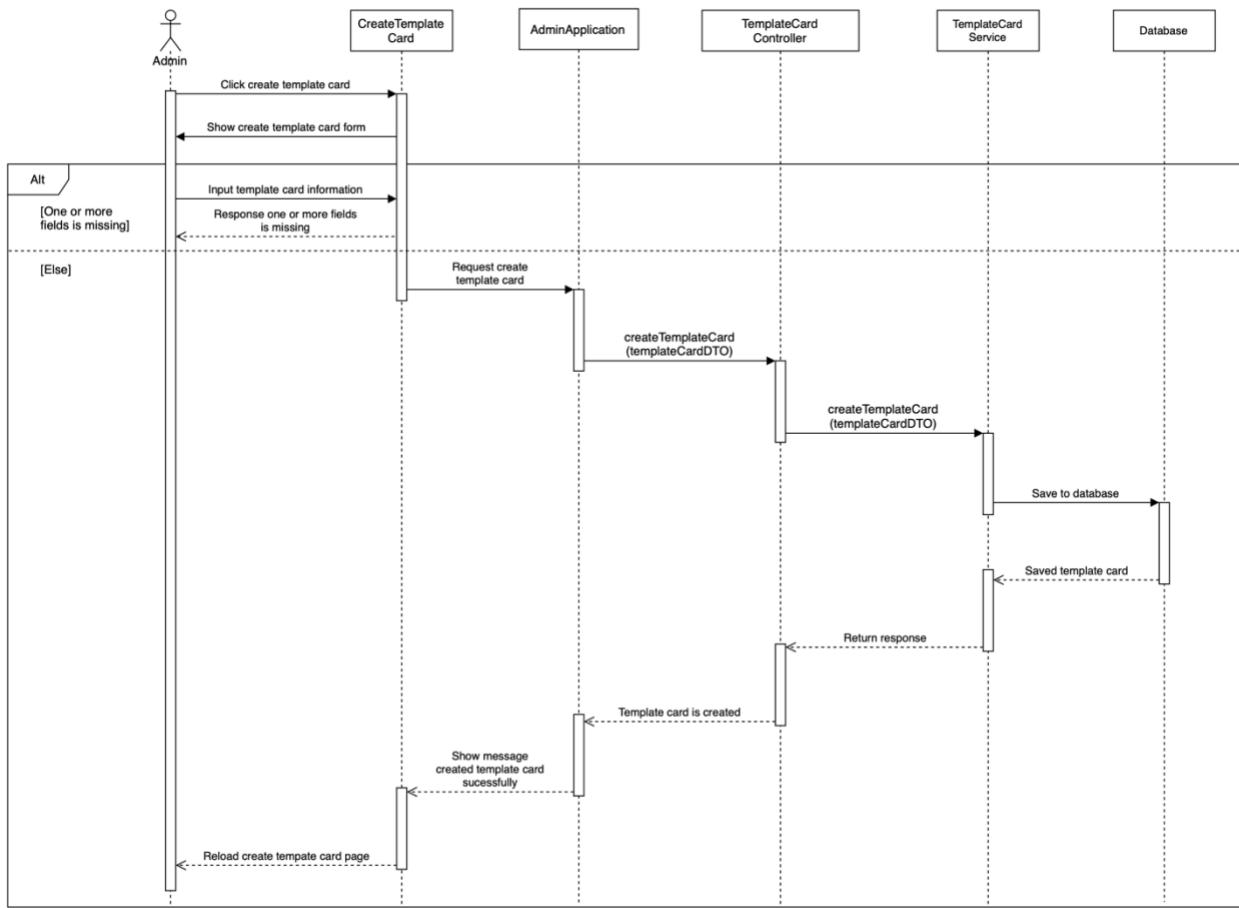


### 3.3.3 Create Template Card

a. Class Diagram

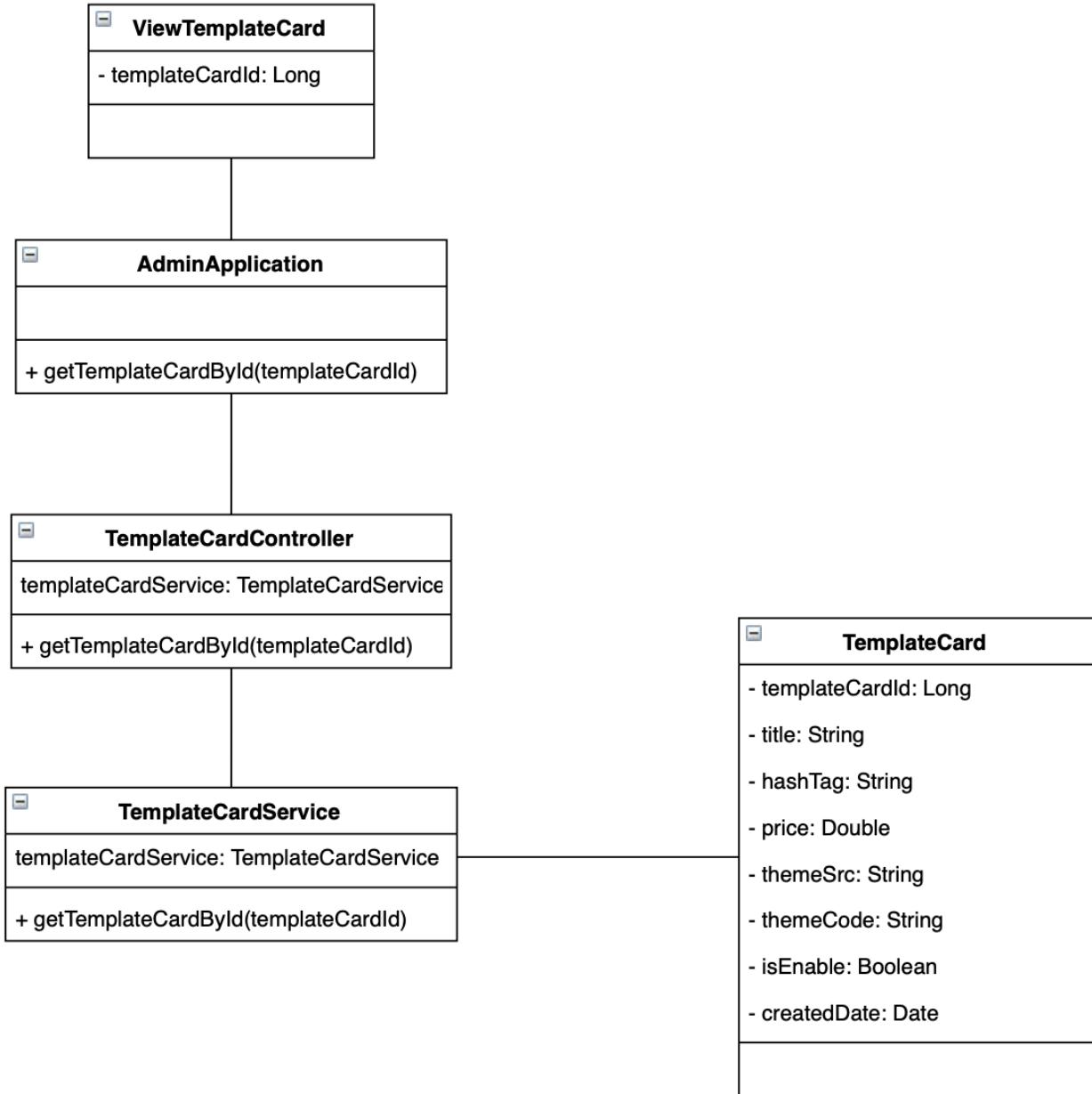


b. Sequence Diagram

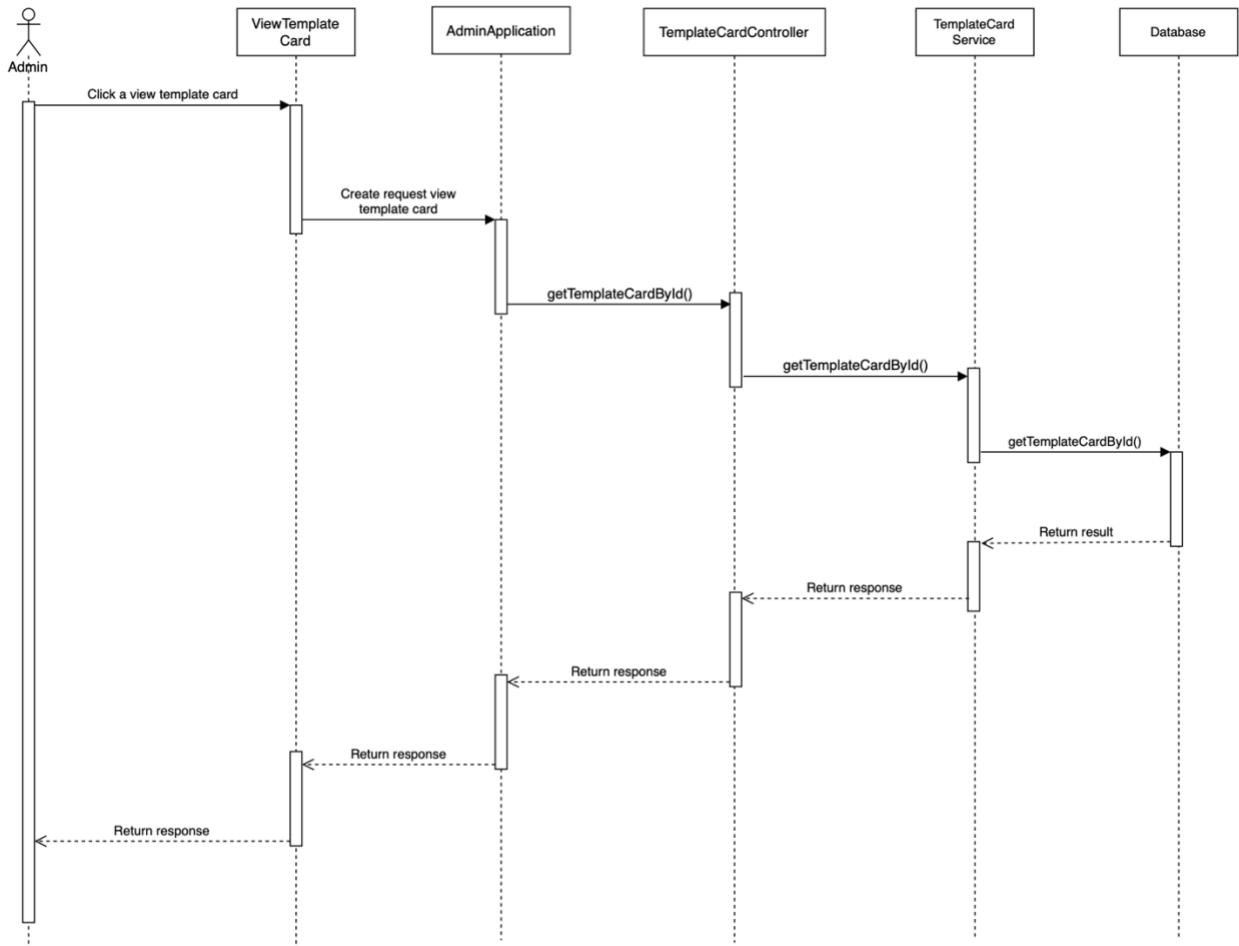


### 3.3.4 View Template Card

a. Class Diagram

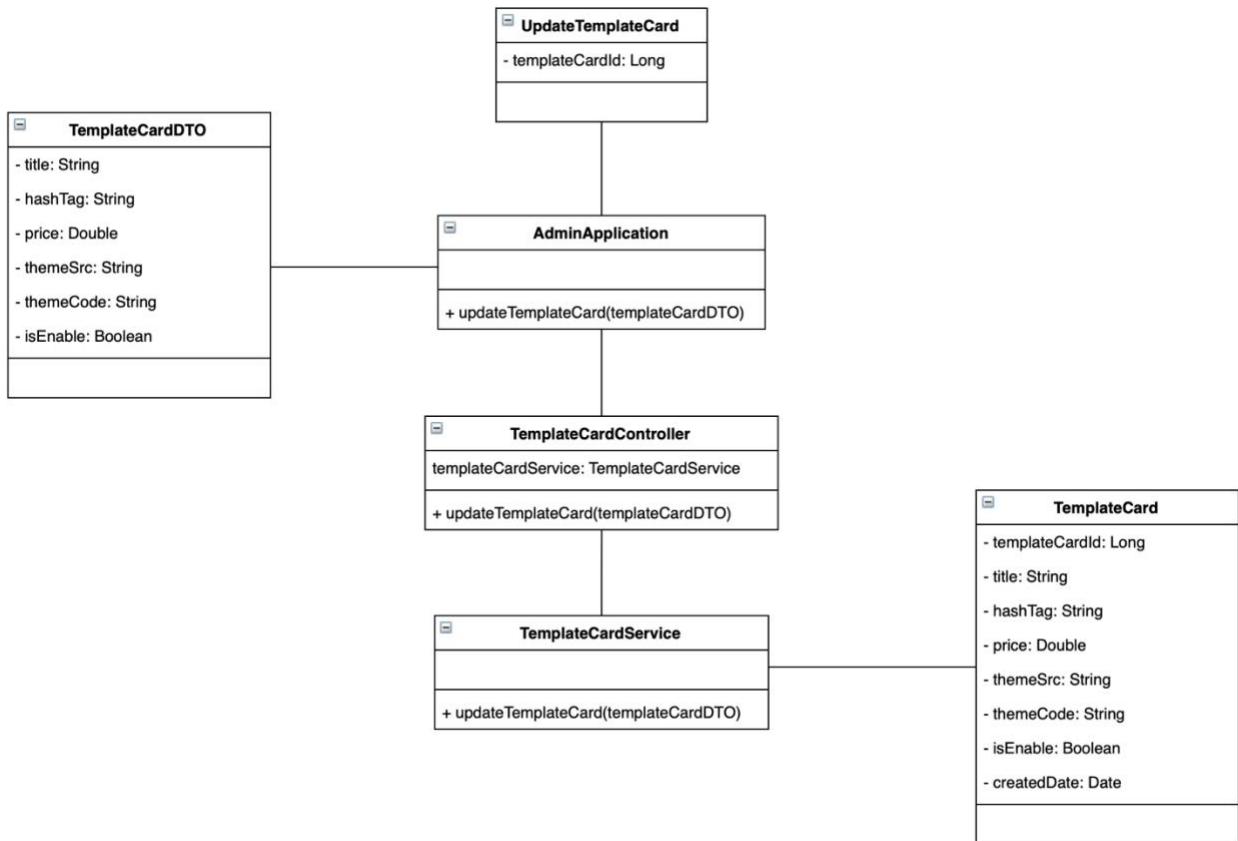


b. Sequence Diagram

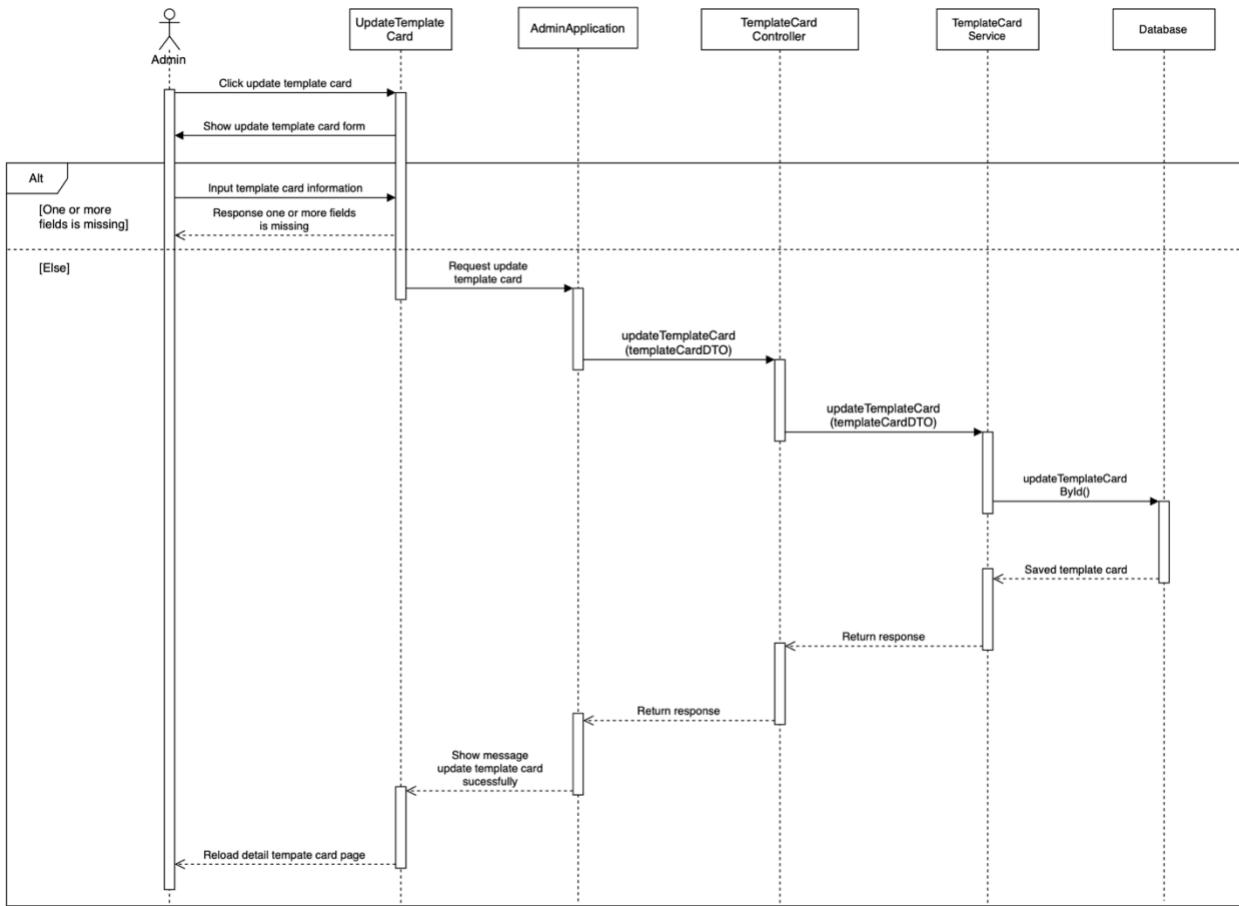


### 3.3.5 Update Template Card

a. Class Diagram

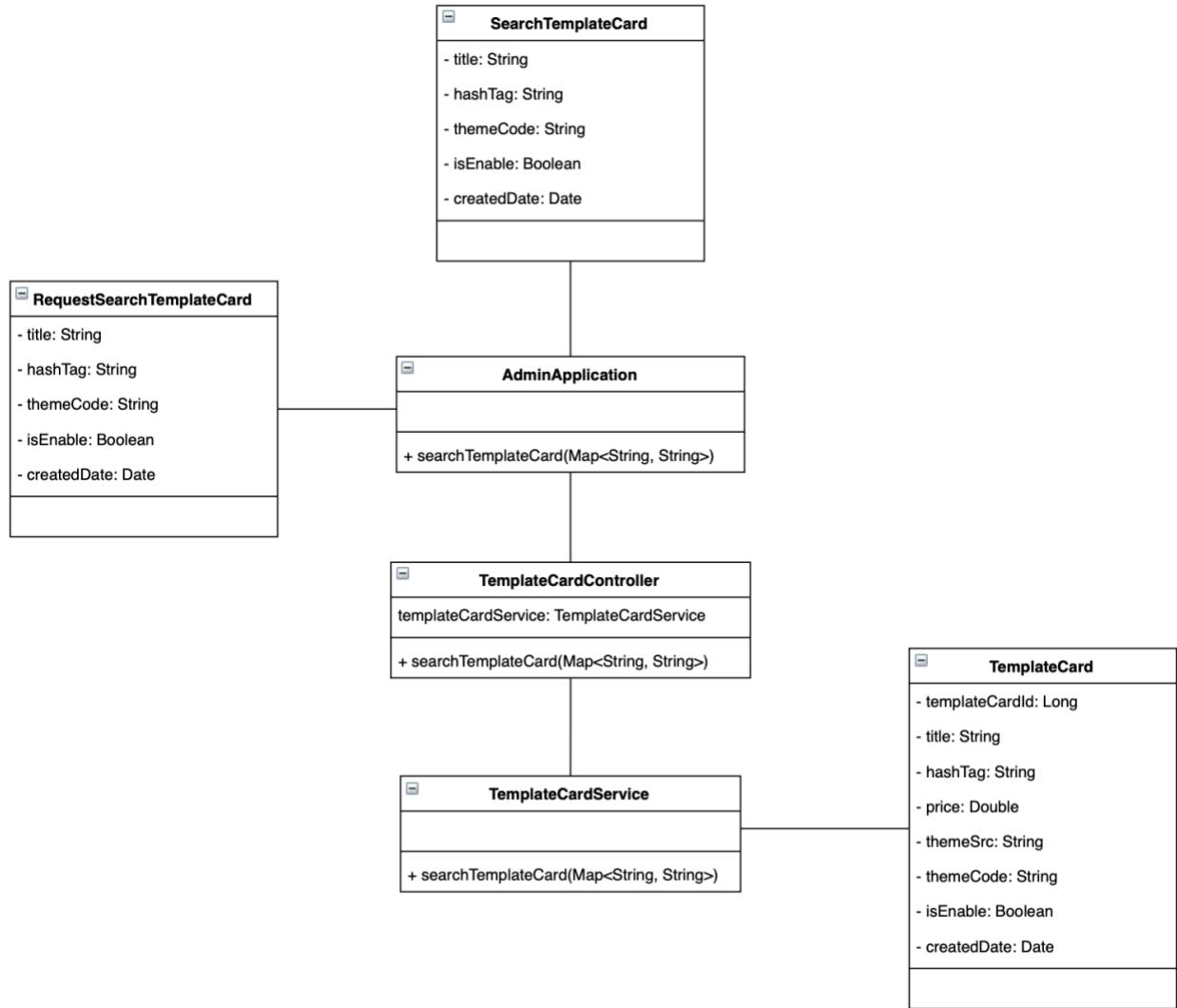


b. Sequence Diagram

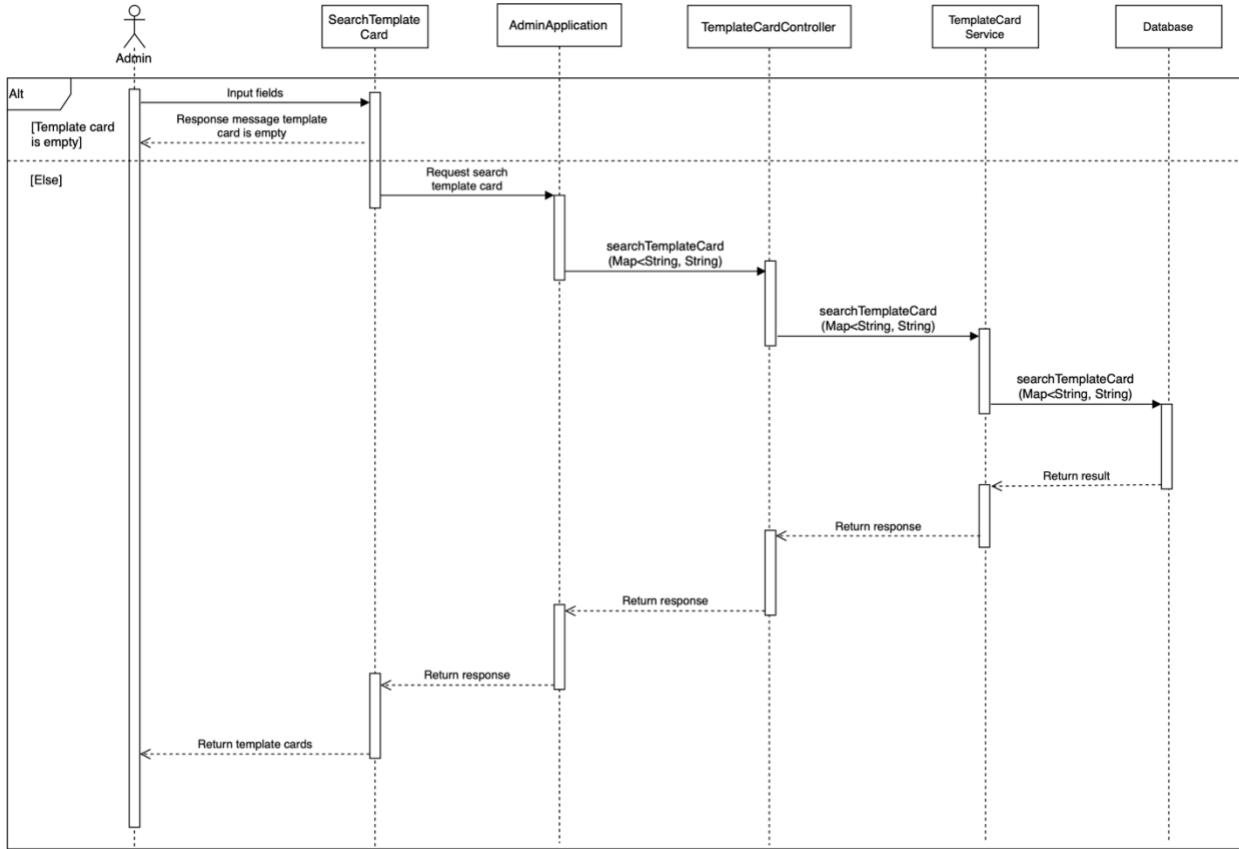


### 3.3.6 Search Template Card

a. Class Diagram

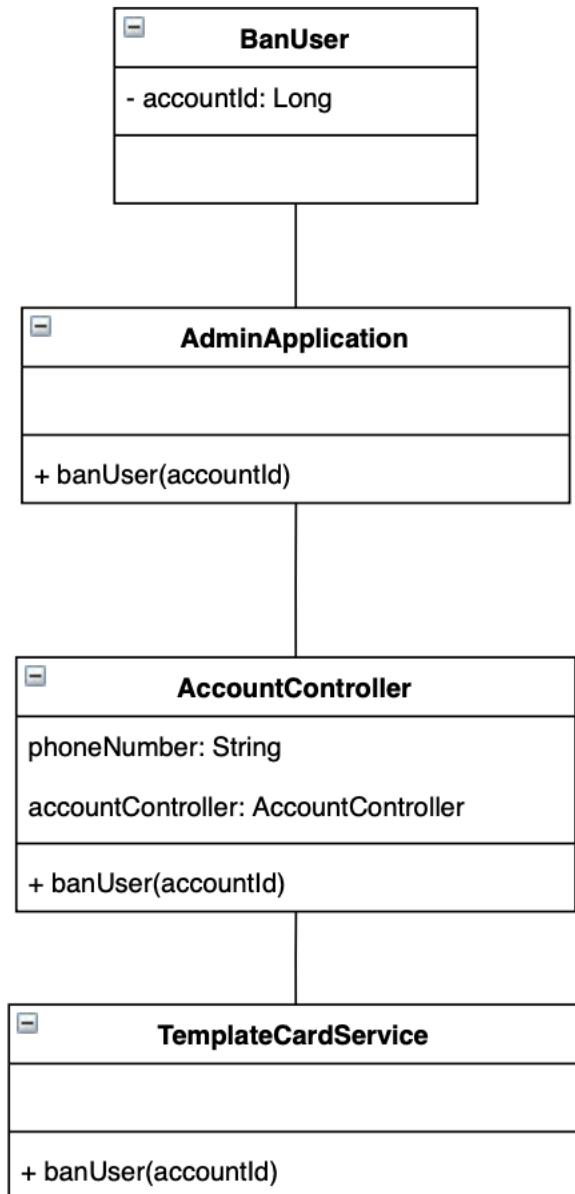


b. Sequence Diagram

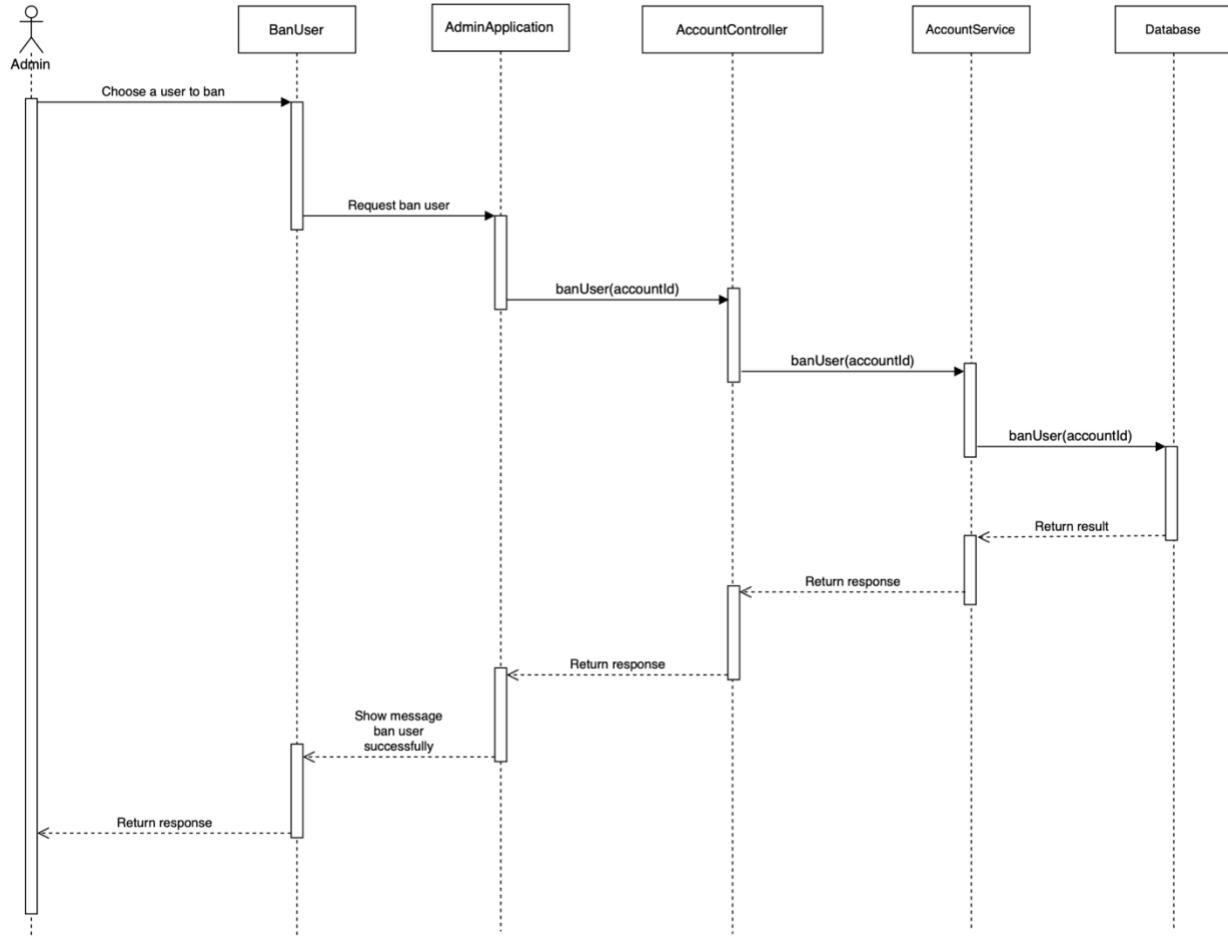


### 3.3.7 Ban User

a. Class Diagram

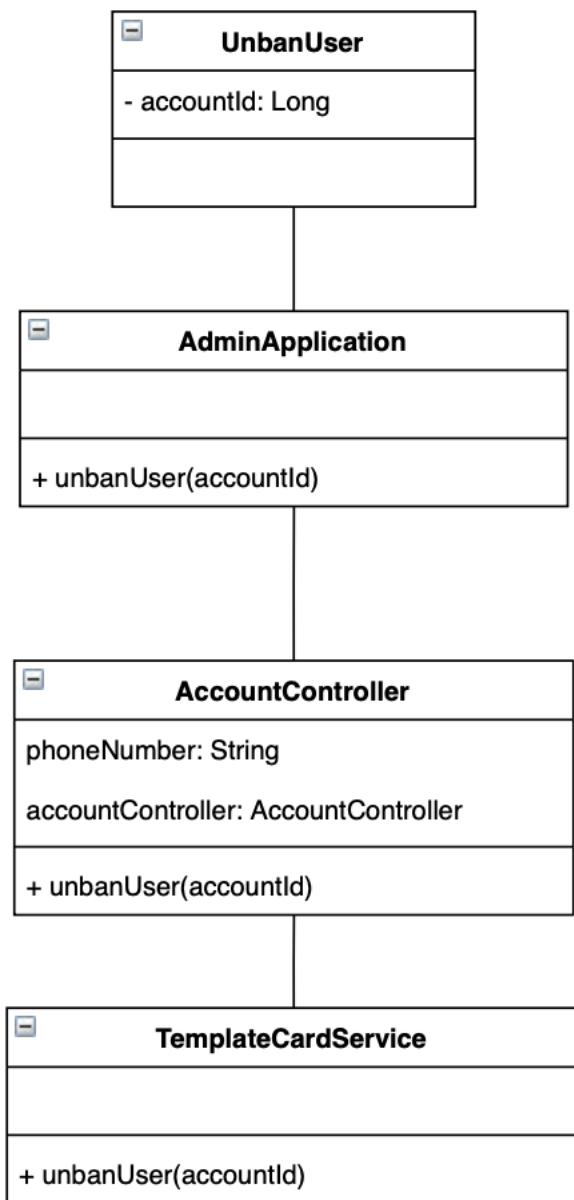


b. Sequence Diagram

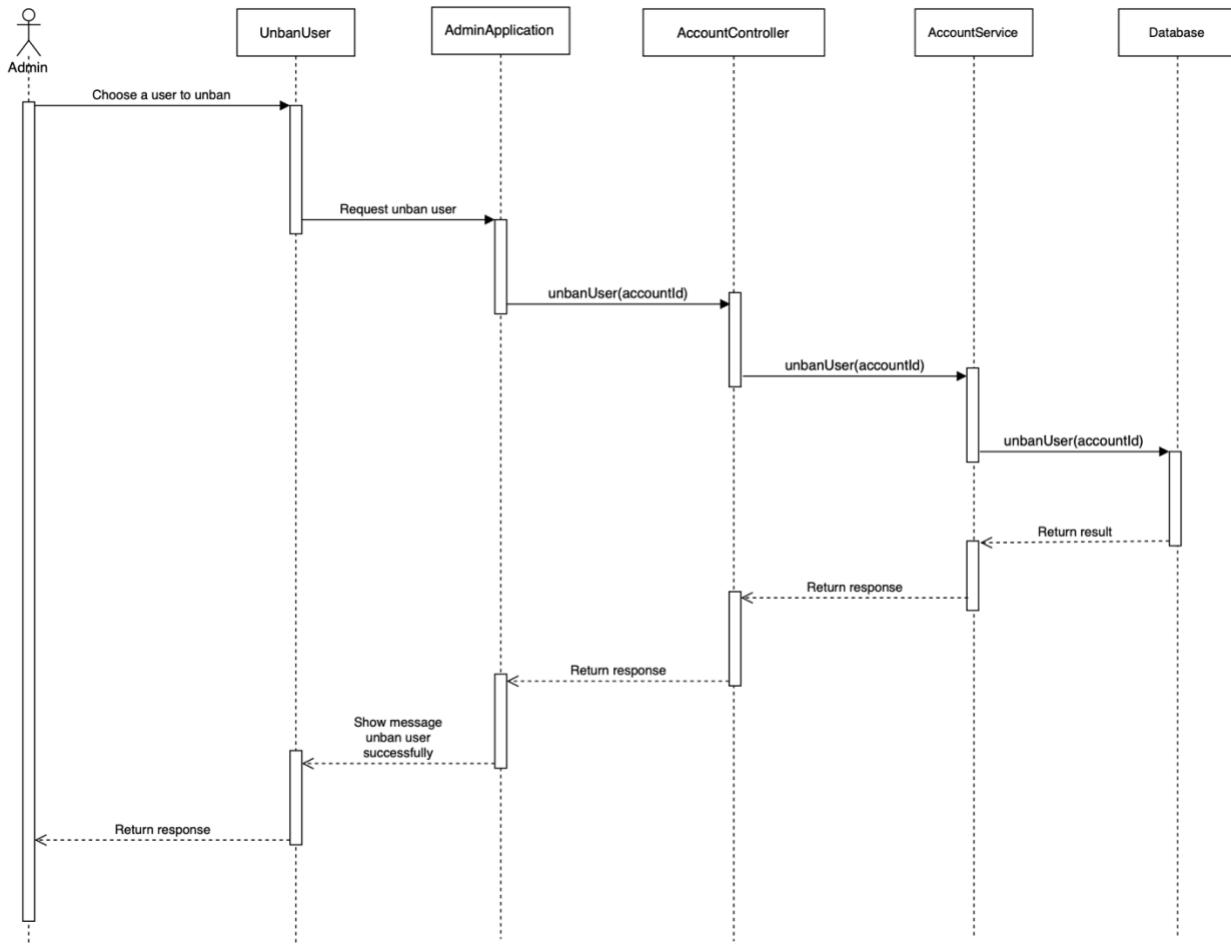


### 3.3.8 Unban User

a. Class Diagram

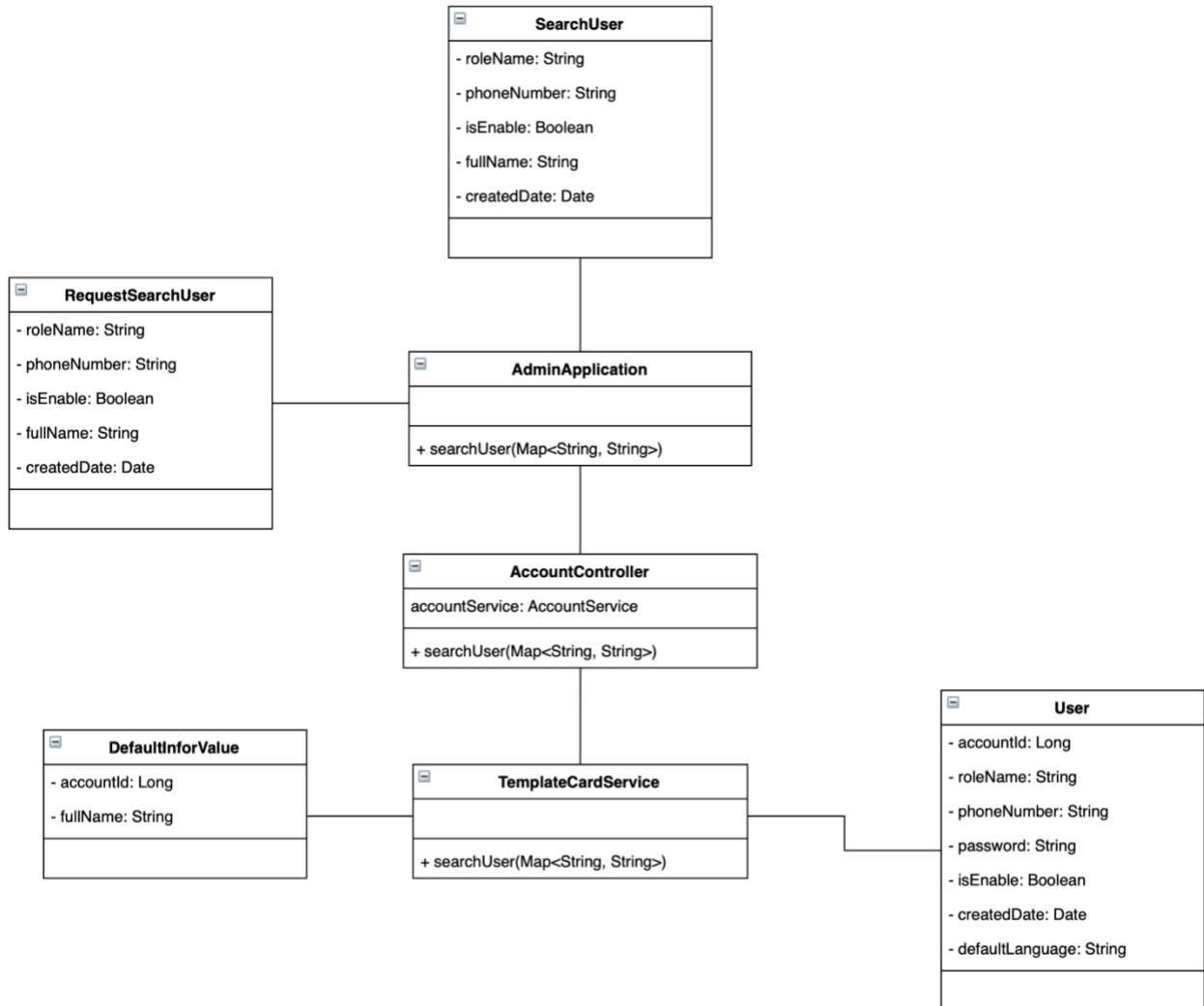


b. Sequence Diagram

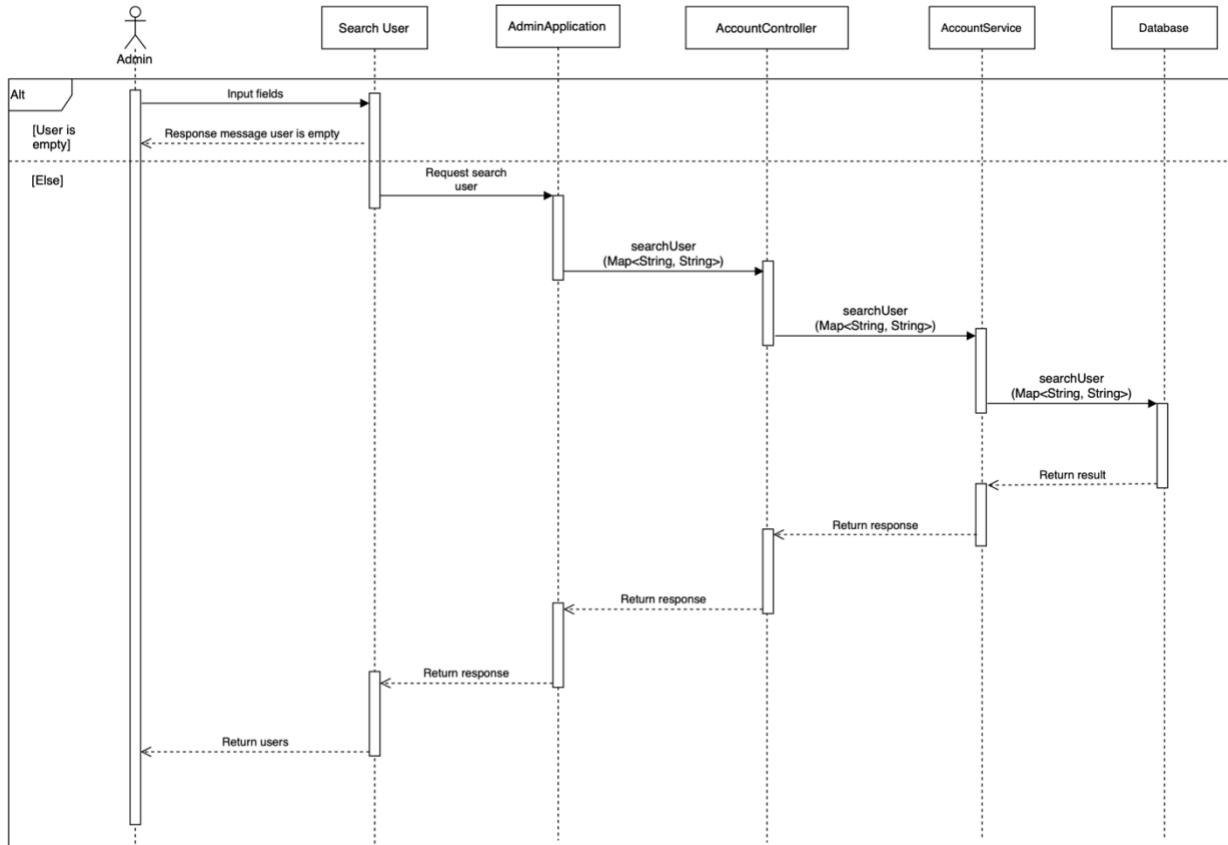


### 3.3.9 Search User

a. Class Diagram

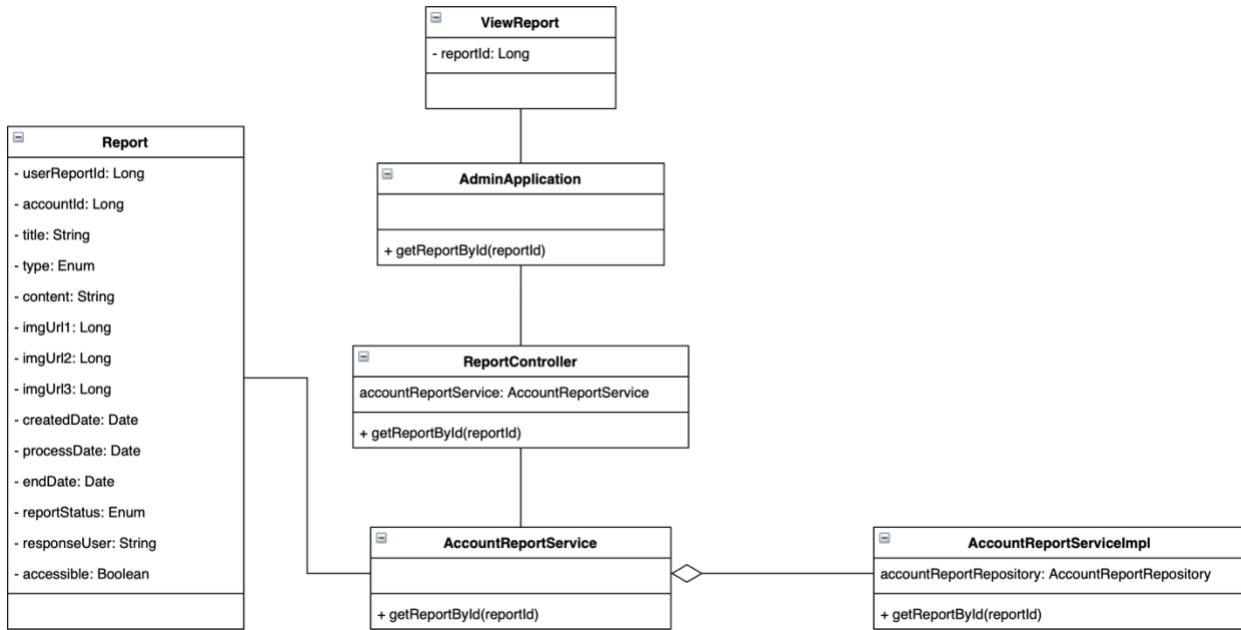


b. Sequence Diagram

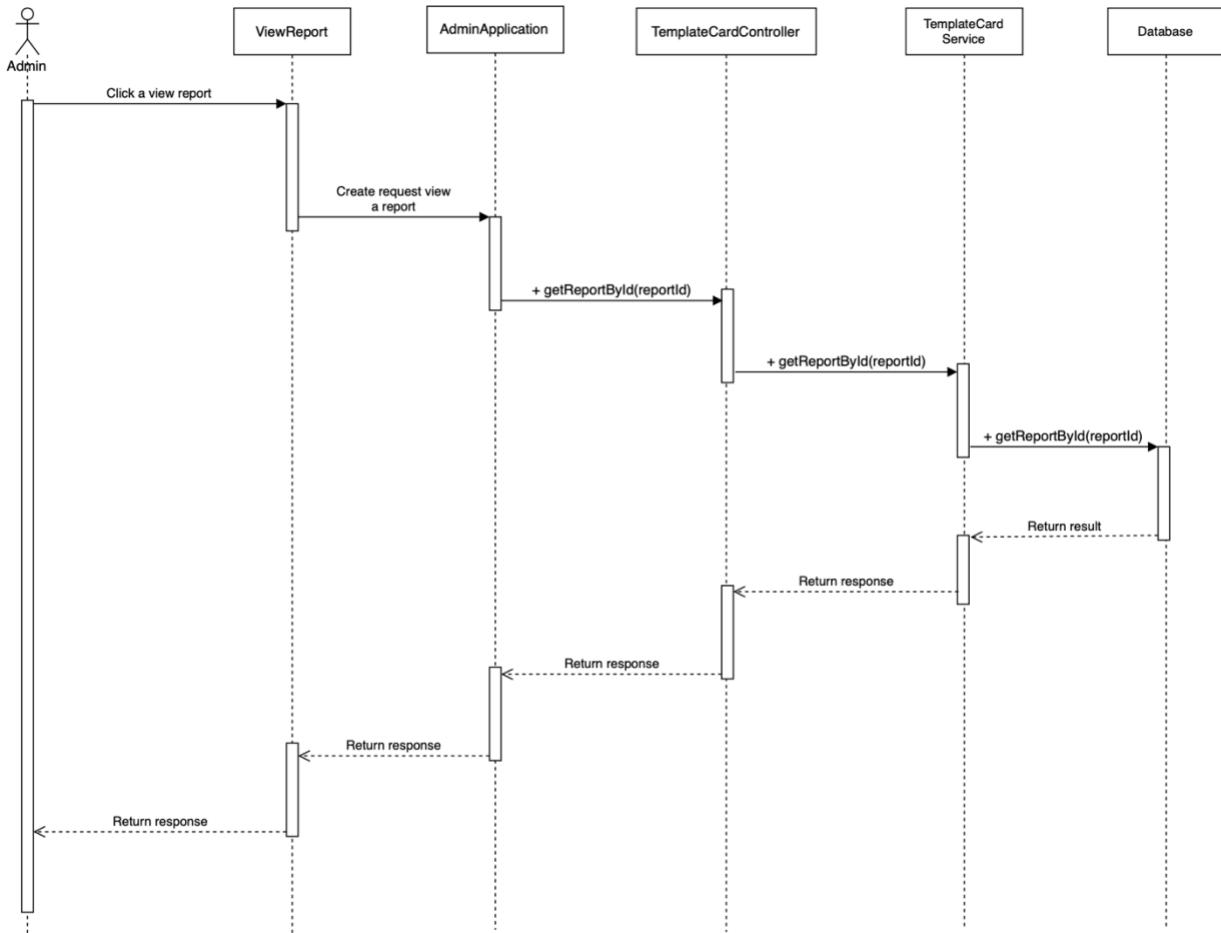


### 3.3.10 View Report

a. Class Diagram

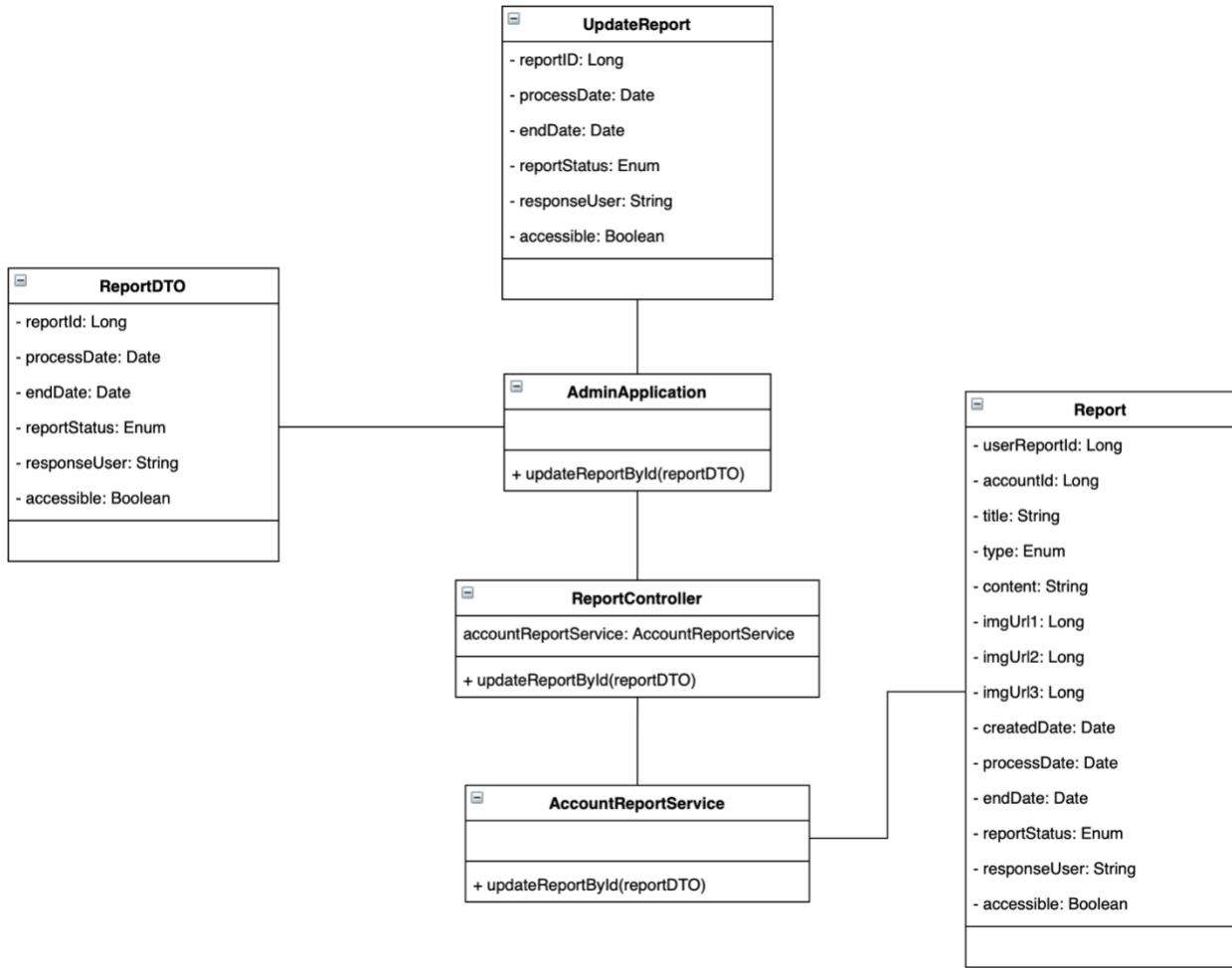


### b. Sequence Diagram

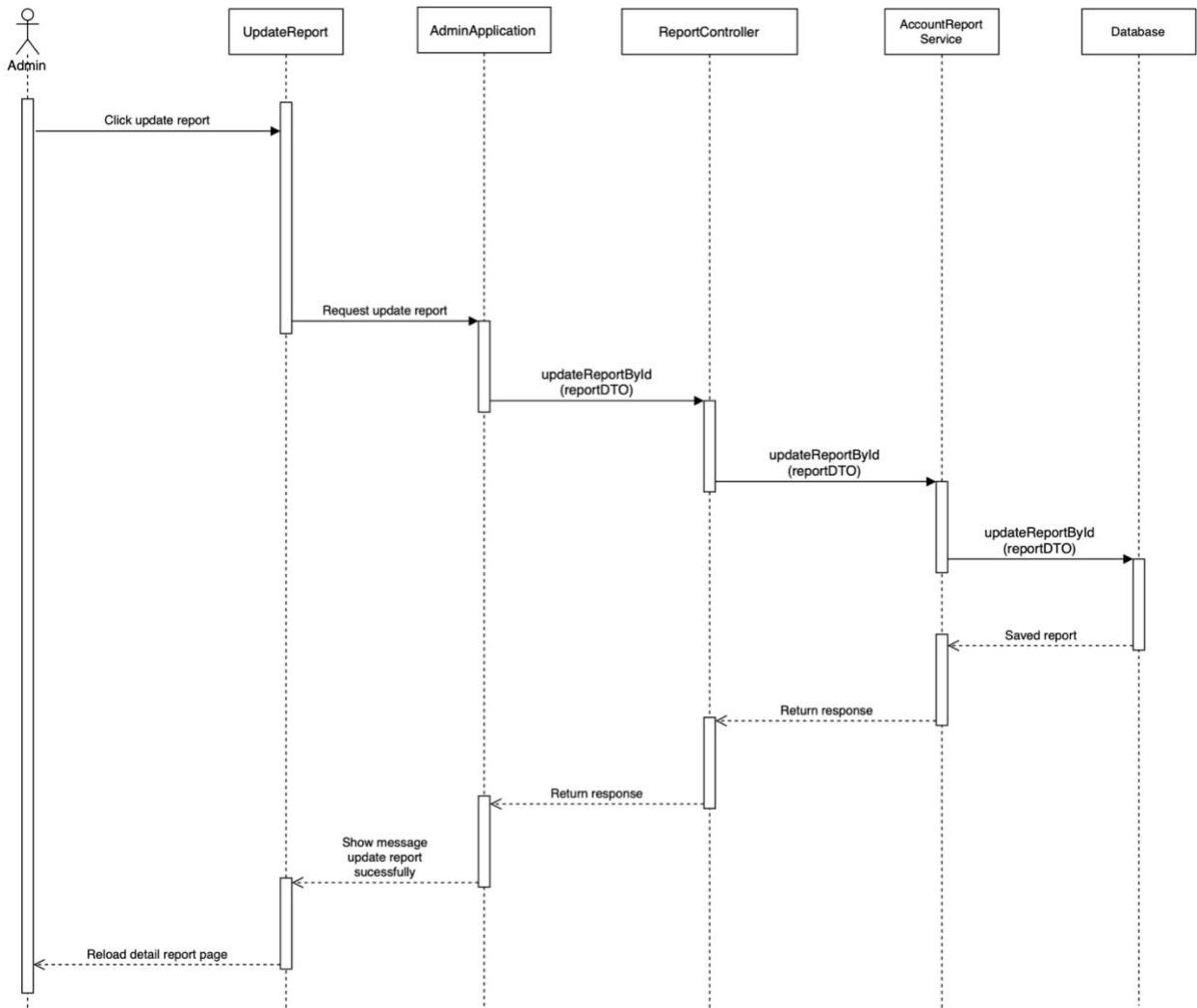


### 3.3.11 Update Report

#### a. Class Diagram

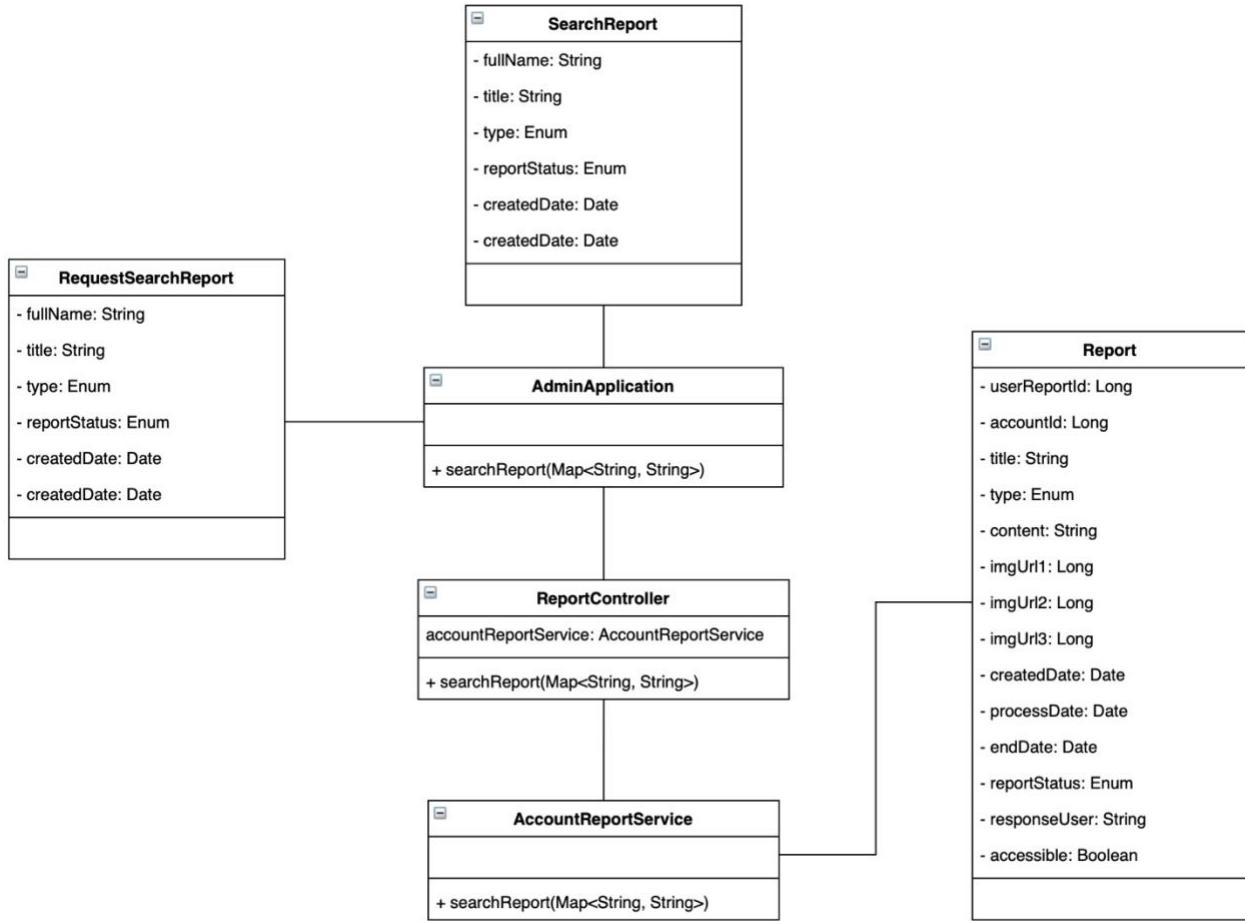


b. Sequence Diagram

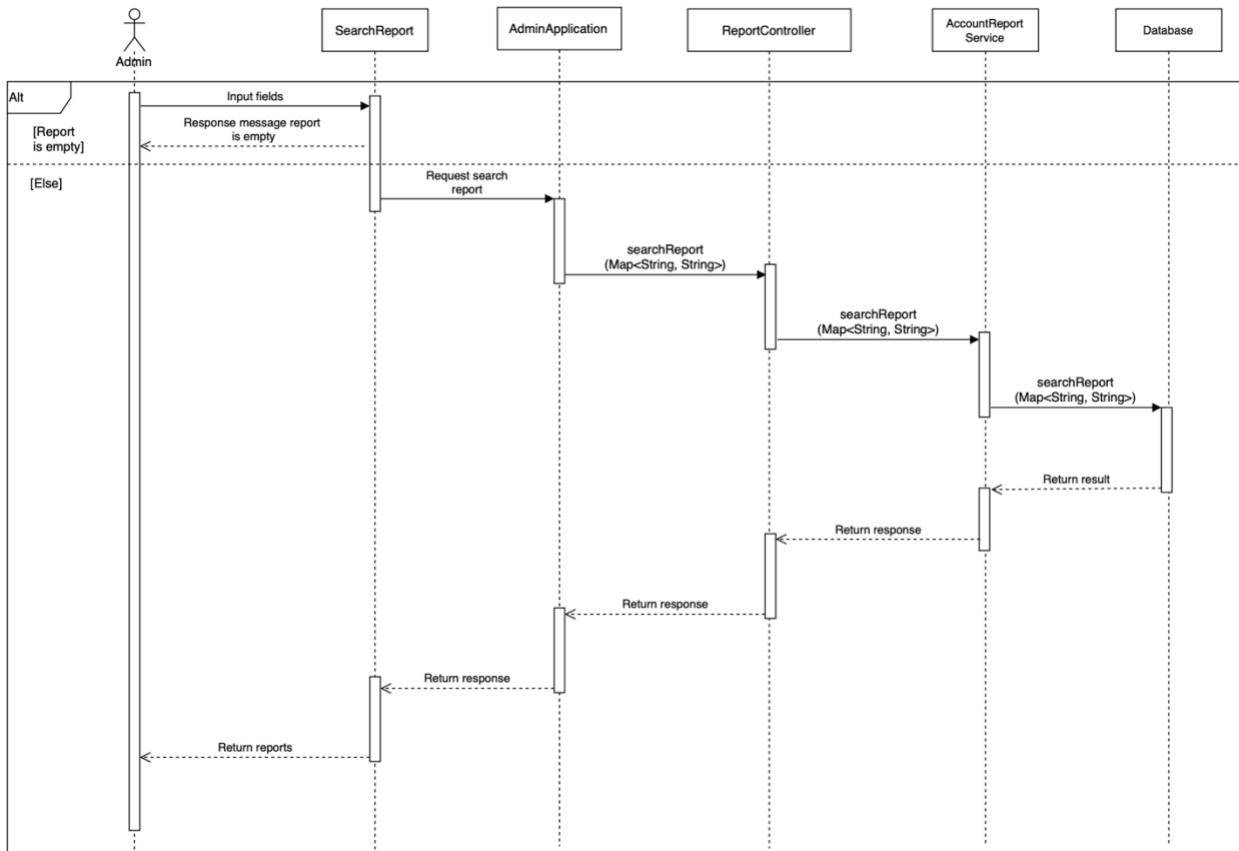


### 3.3.12 Search Report

a. Class Diagram

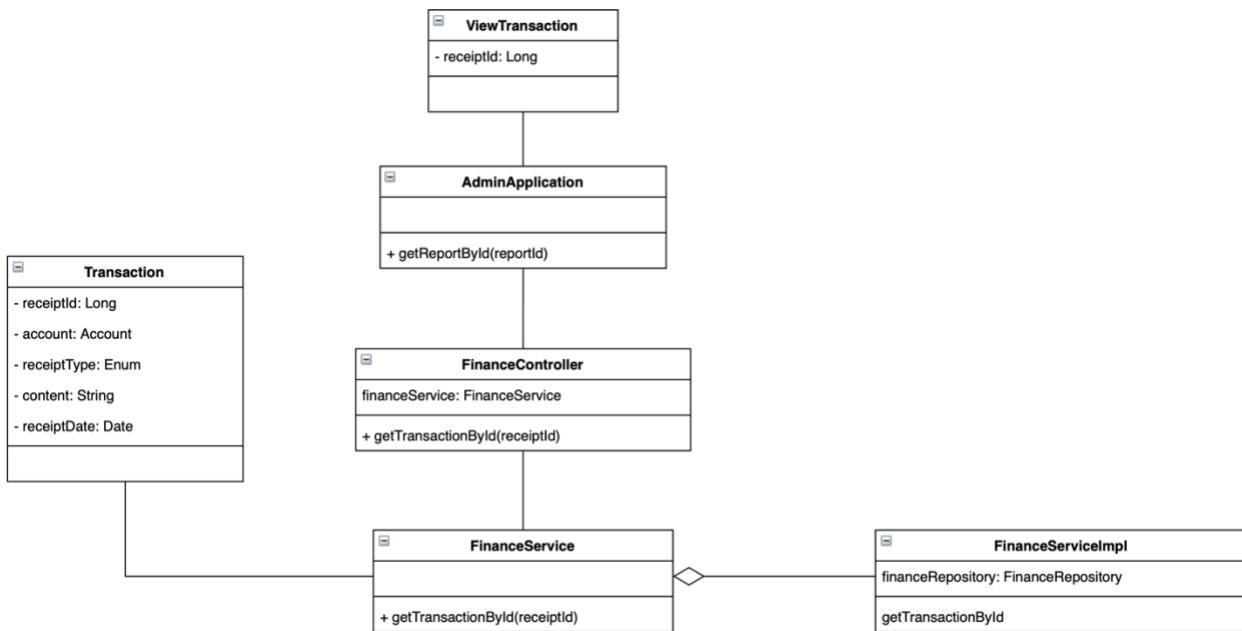


b. Sequence Diagram

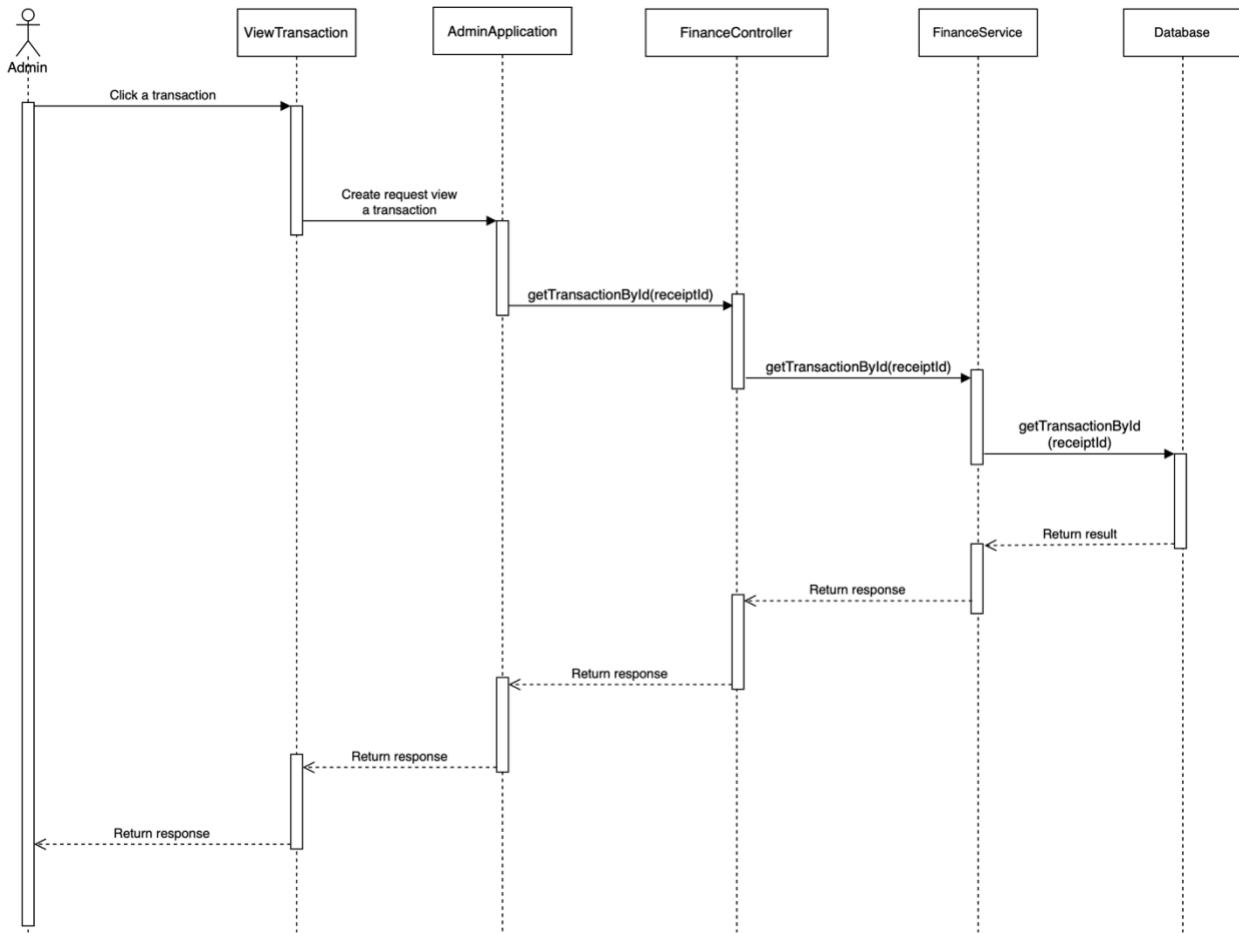


### 3.3.13 View Transaction

a. Class Diagram

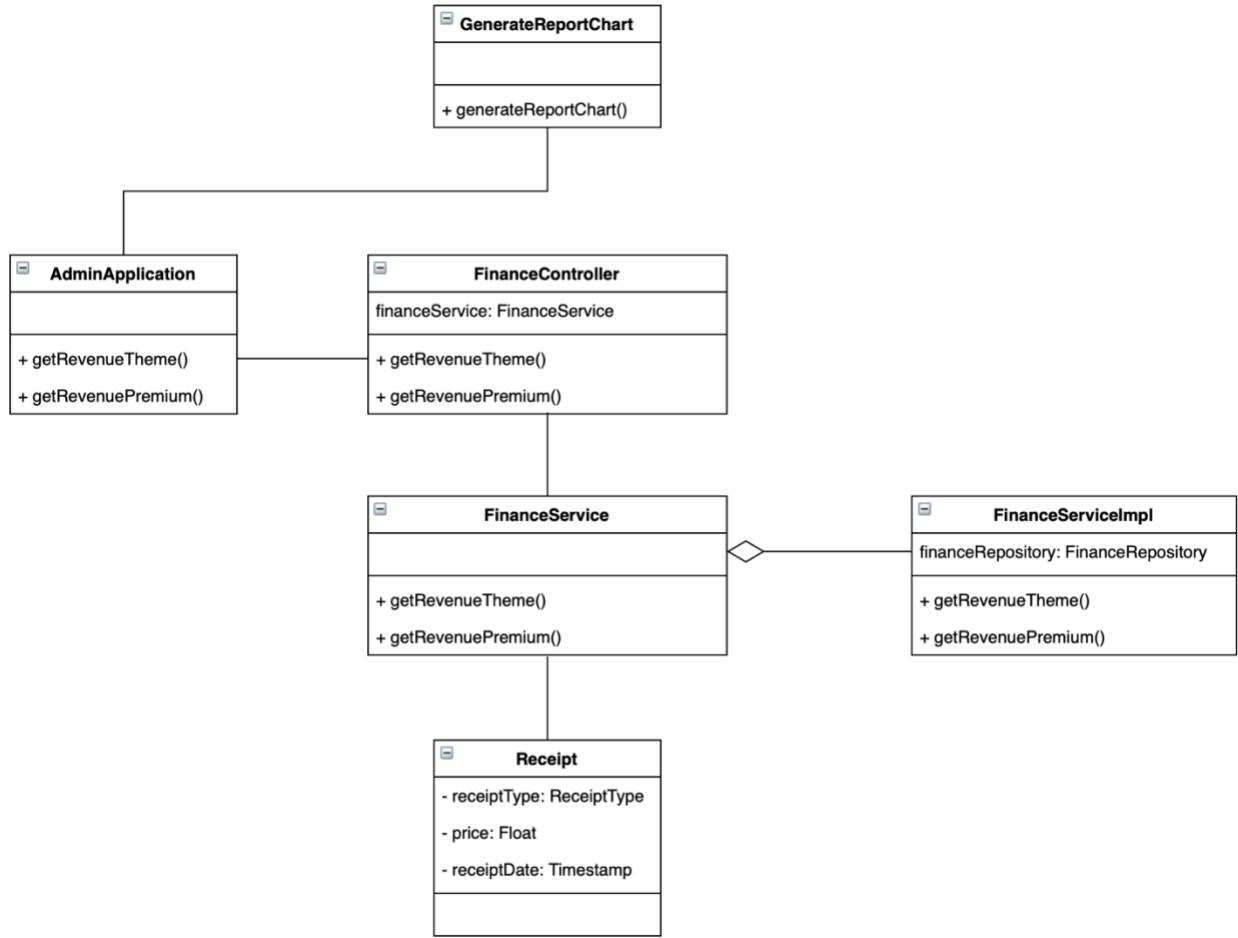


## b. Sequence Diagram

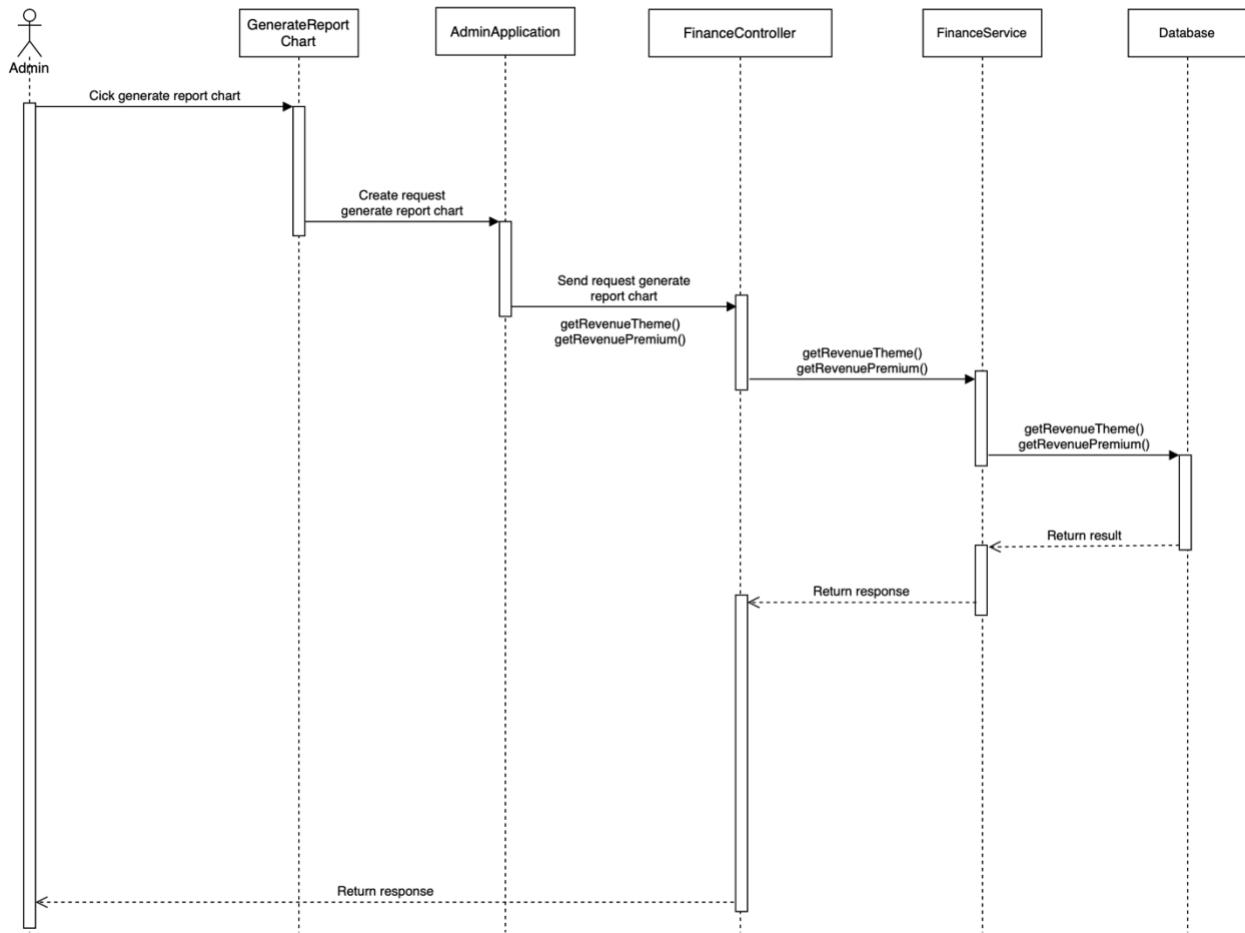


### 3.3.14 Generate Report Chart

a. Class Diagram

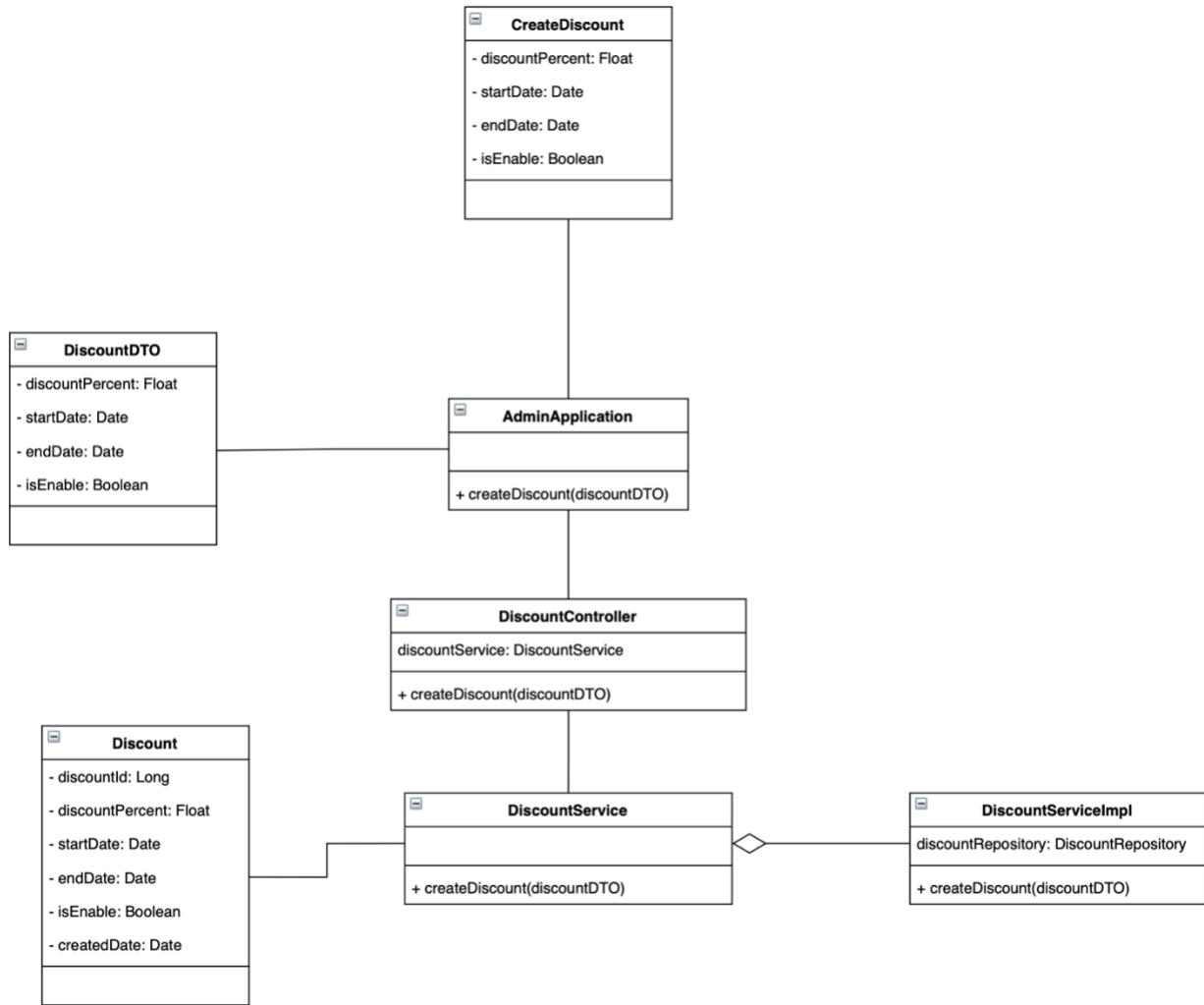


b. Sequence Diagram

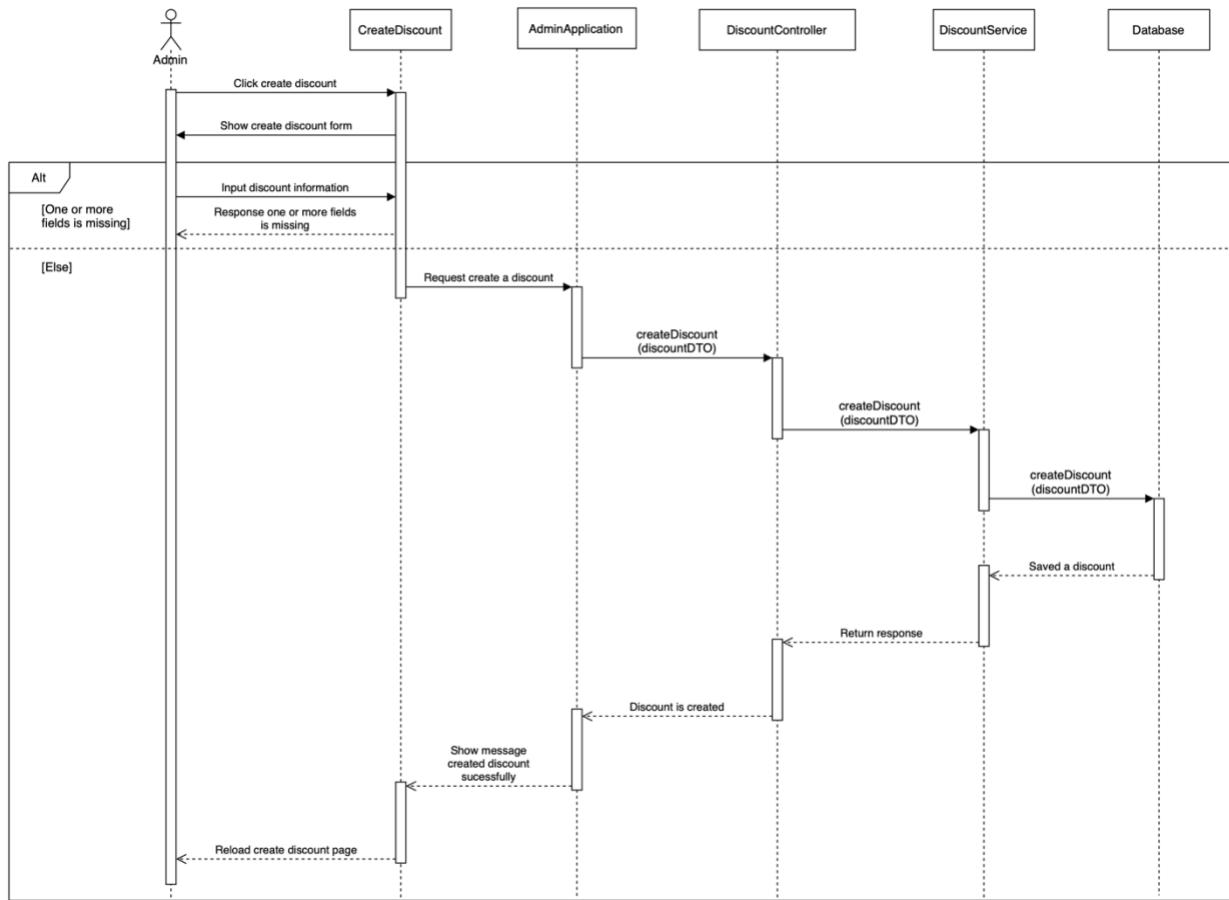


### 3.3.15 Create Discount

a. Class Diagram

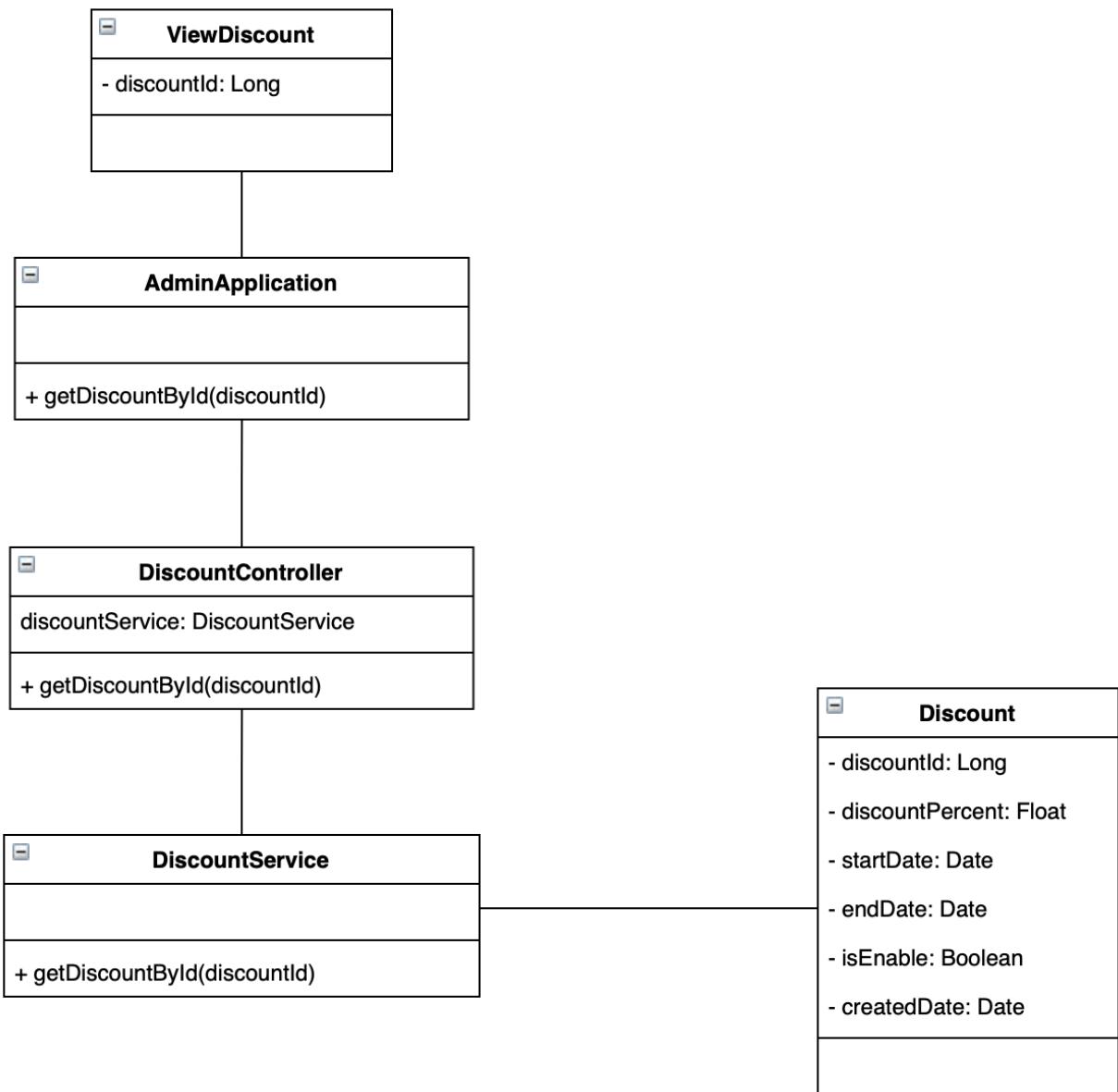


b. Sequence Diagram

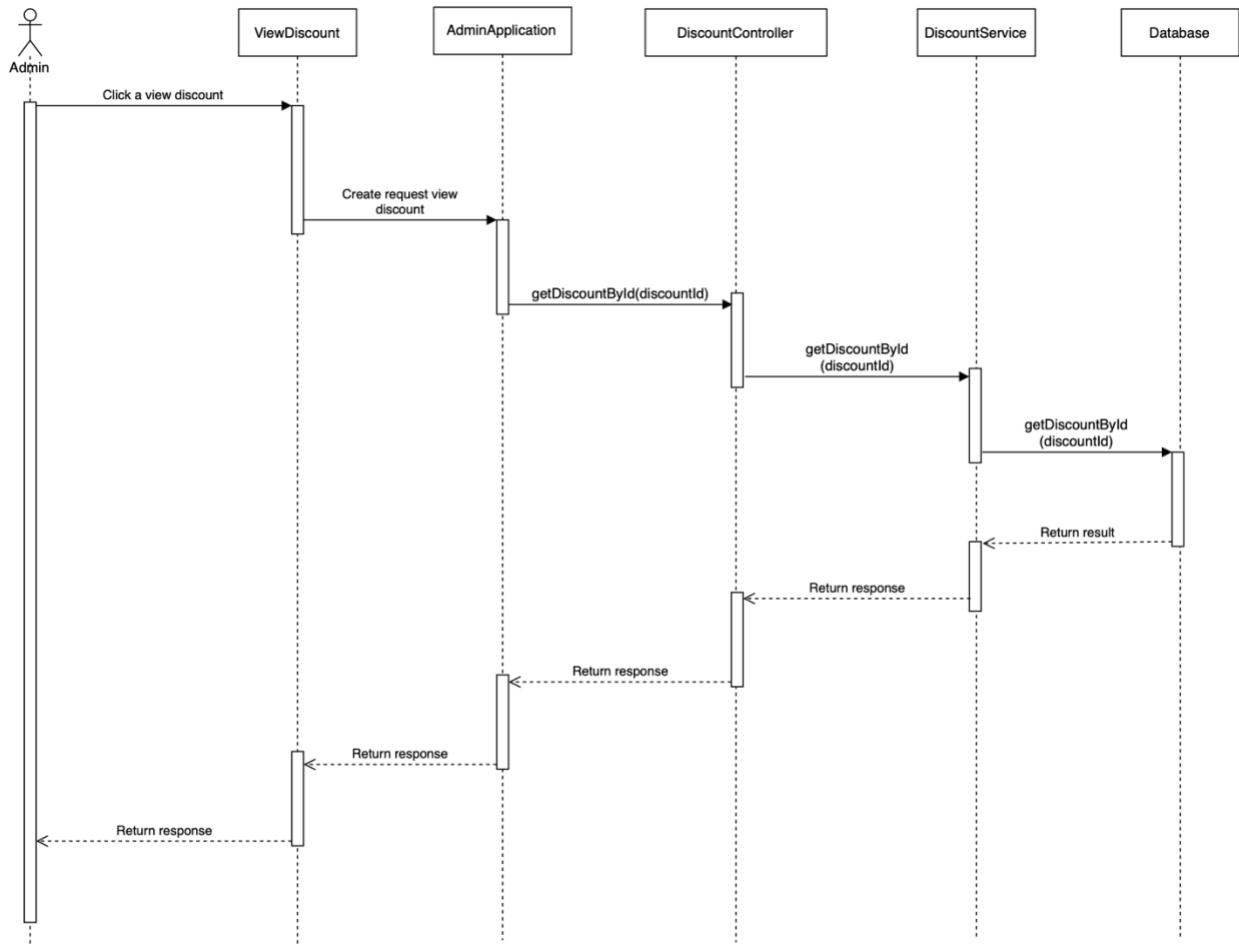


### 3.3.16 View Discount

a. Class Diagram

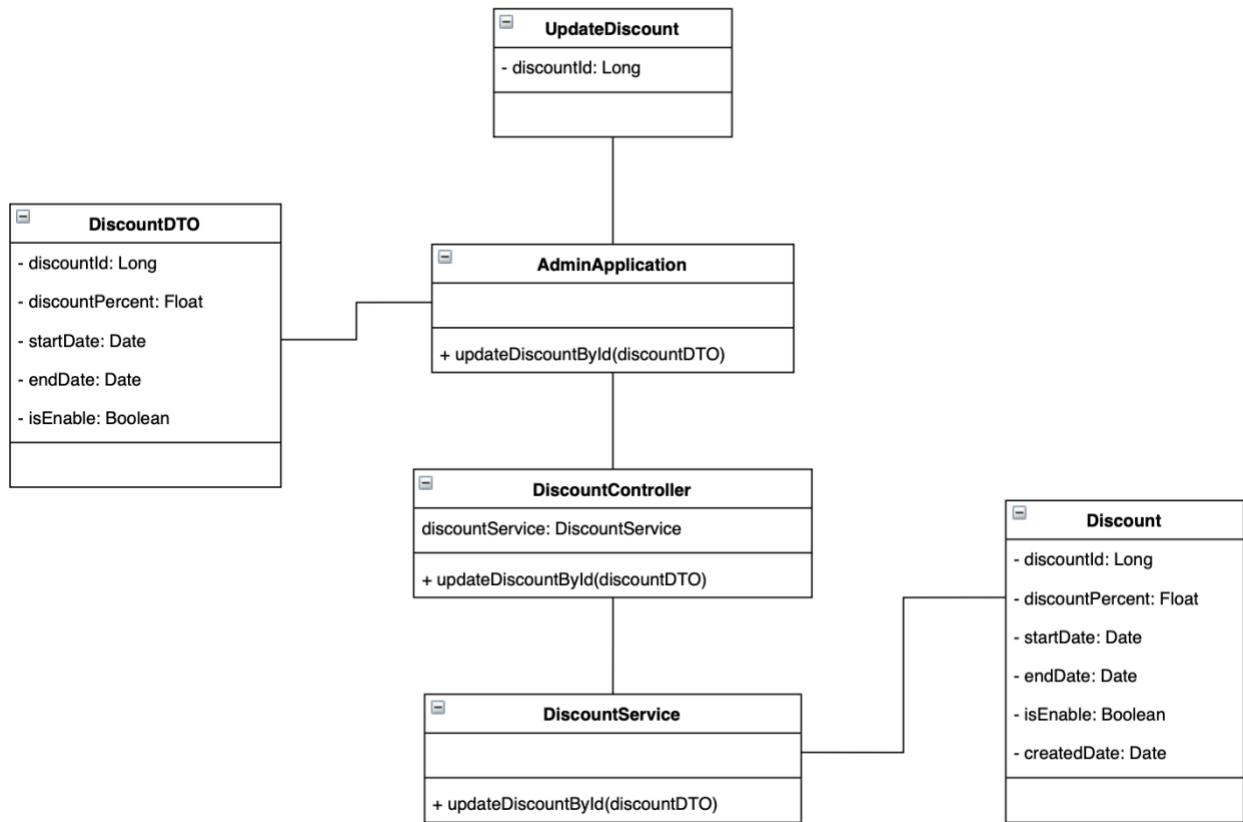


b. Sequence Diagram

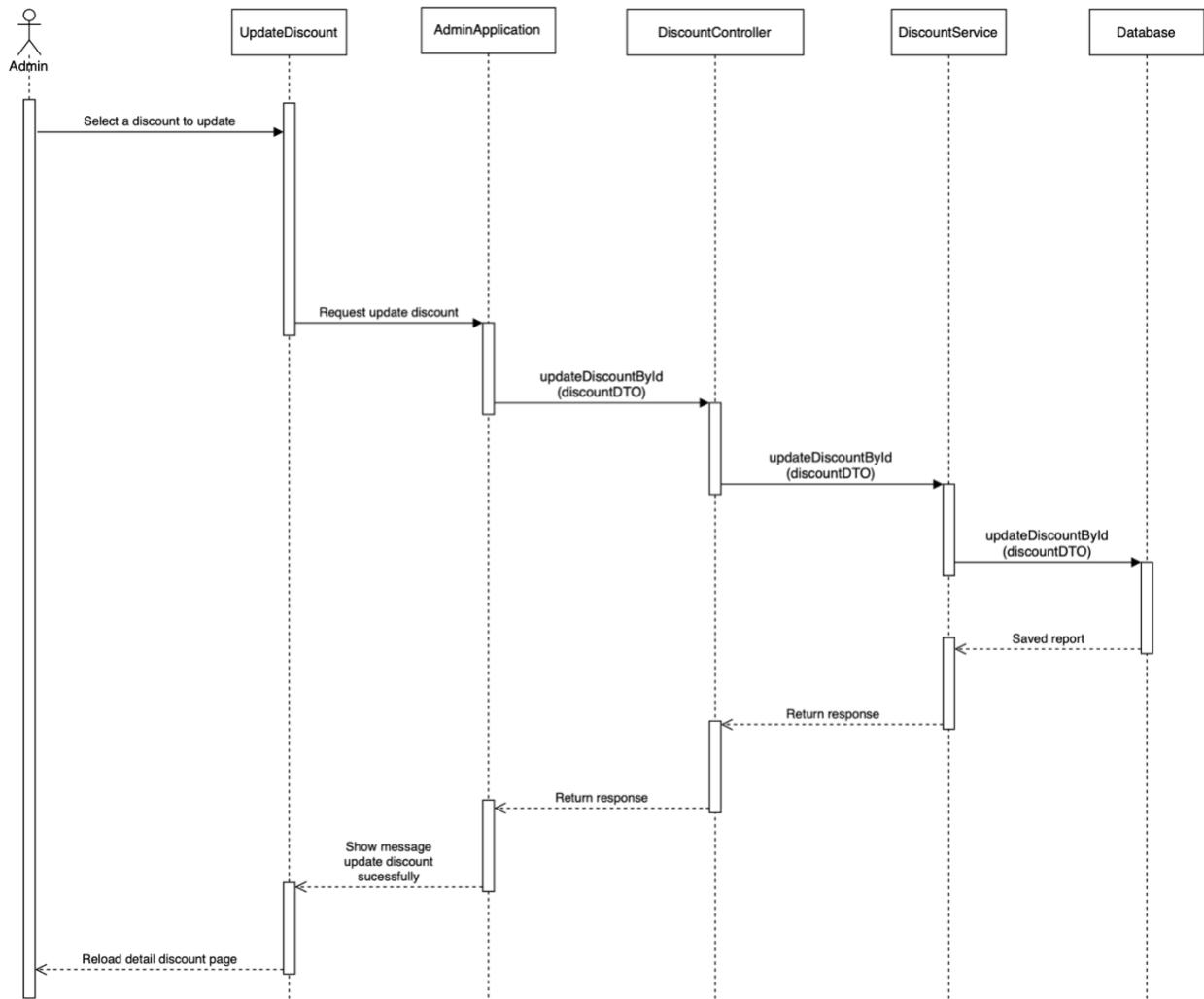


### 3.3.17 Update Discount

a. Class Diagram

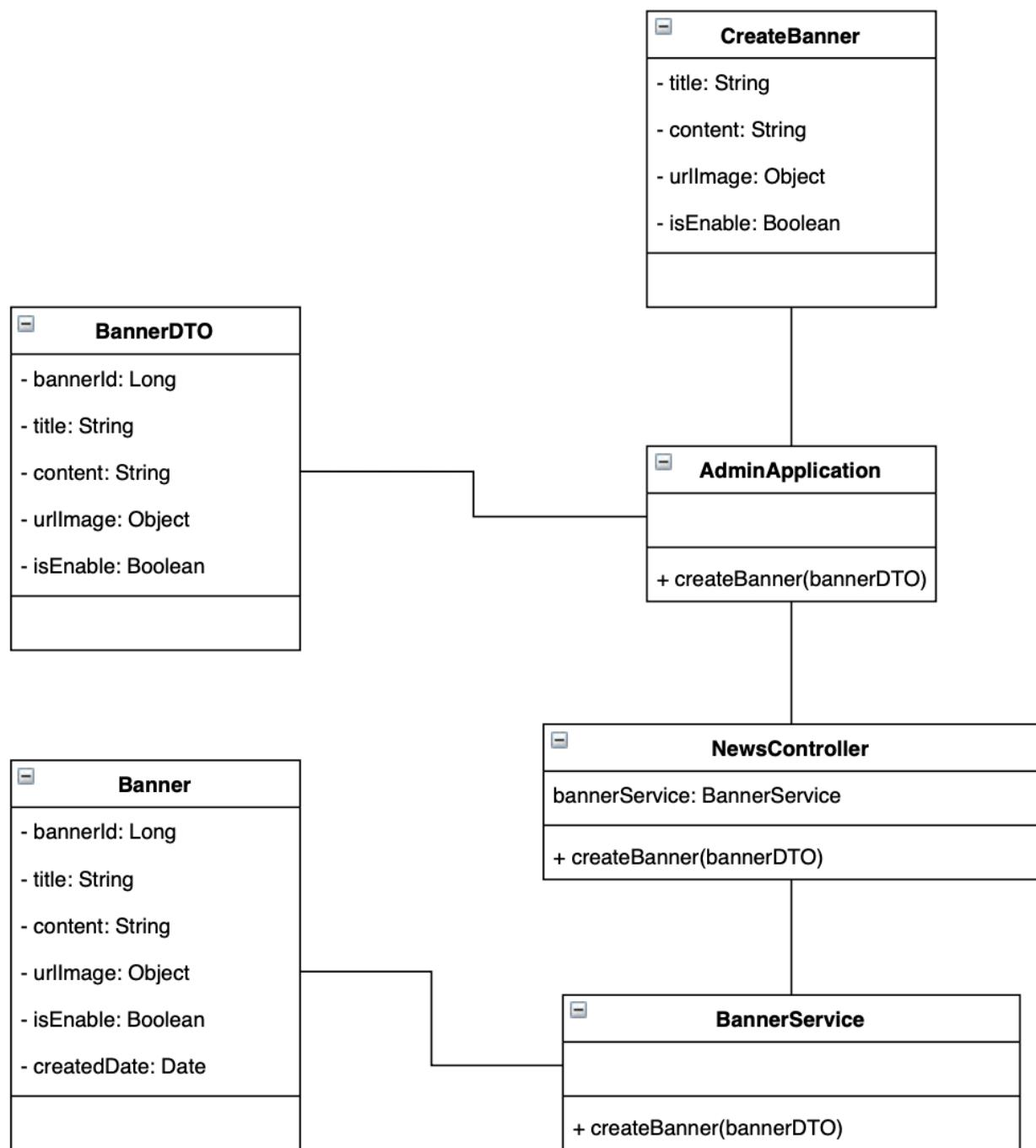


b. Sequence Diagram

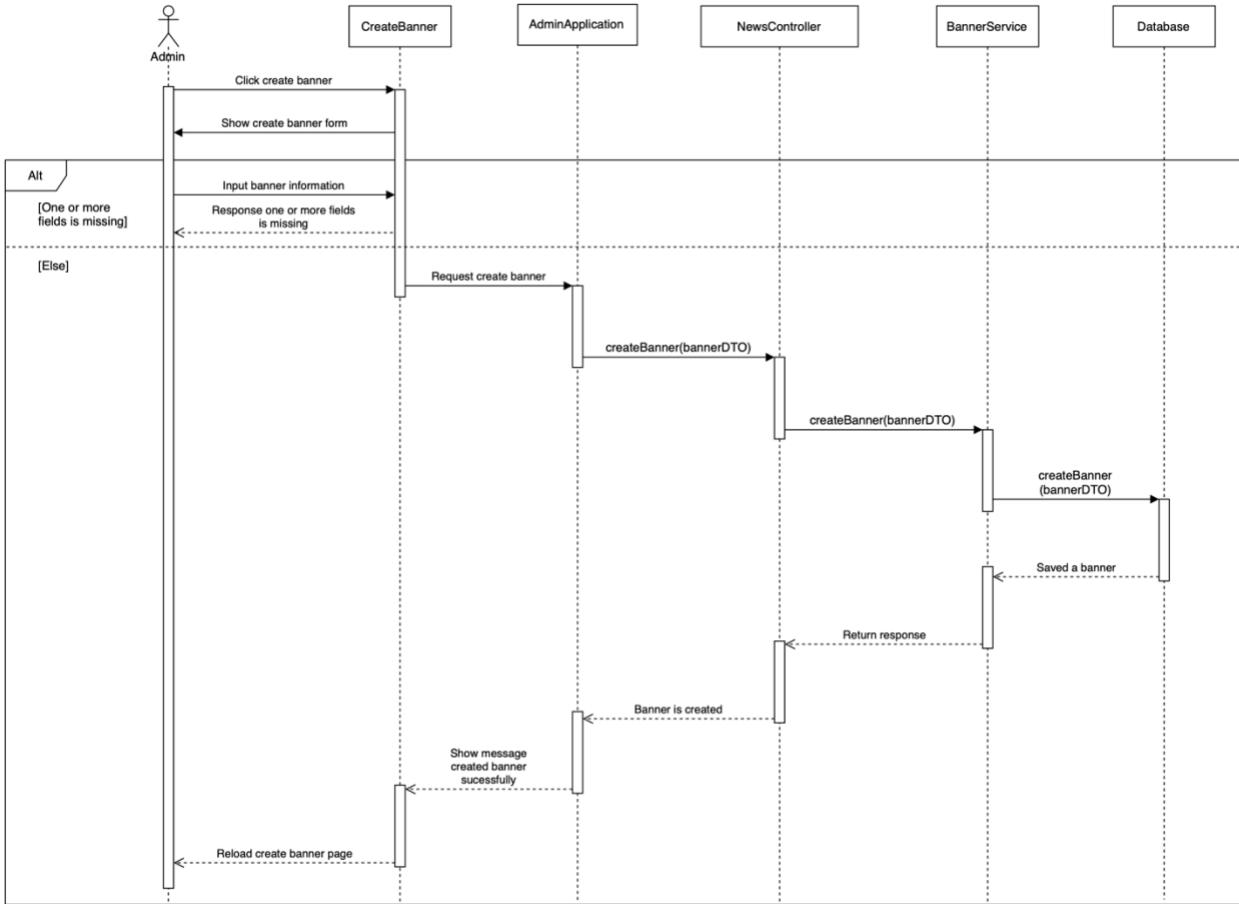


### 3.3.18 Create Banner

a. Class Diagram

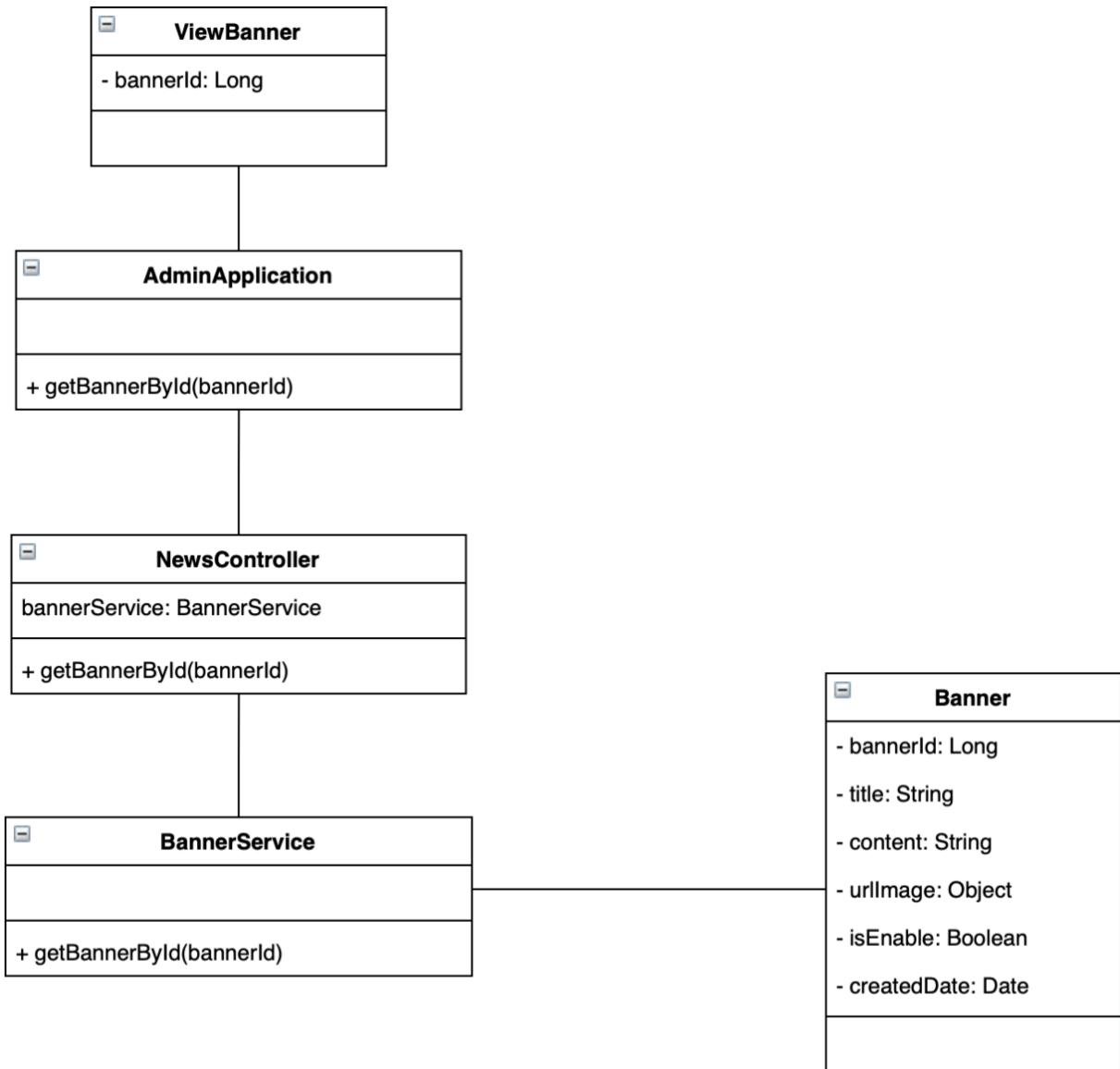


b. Sequence Diagram

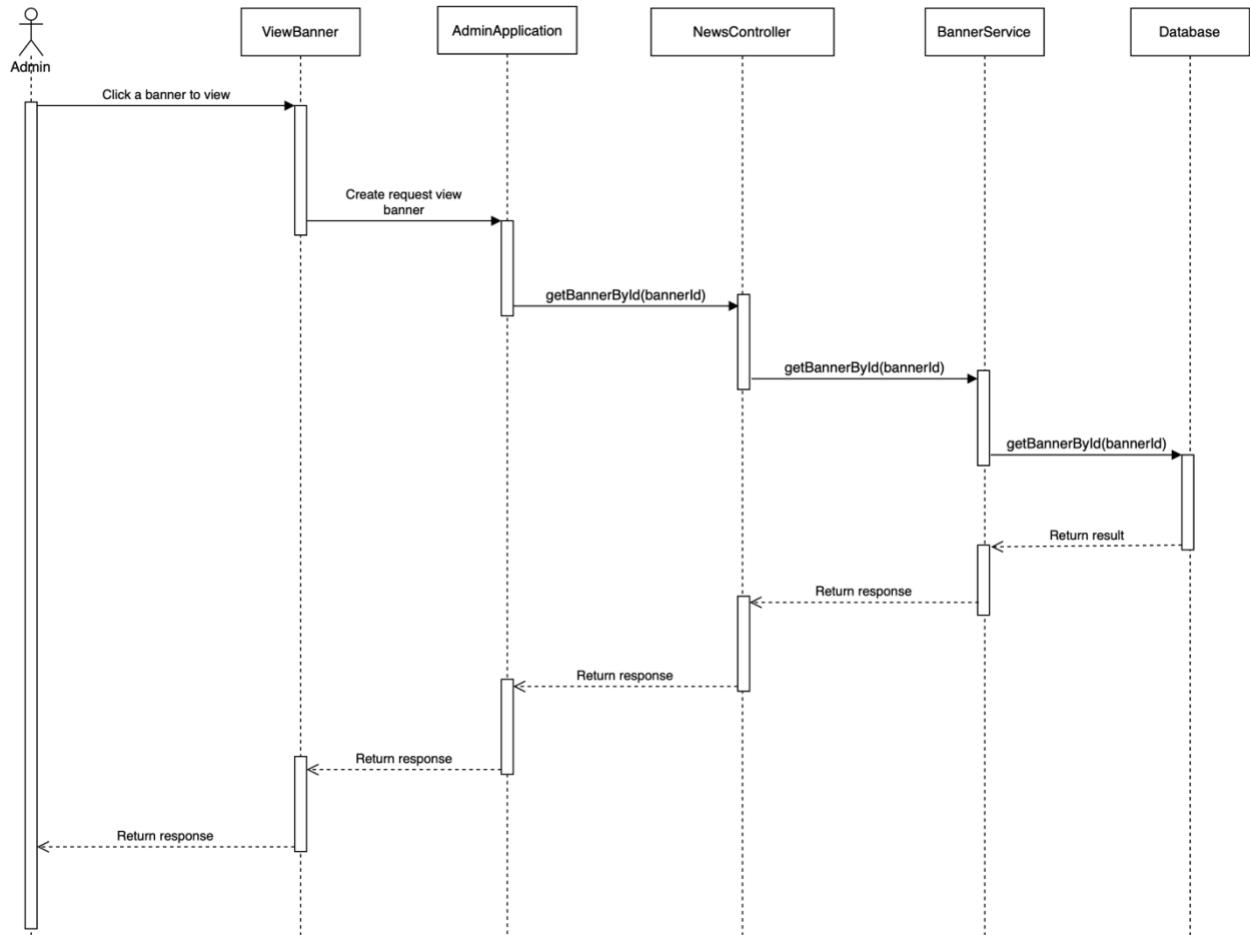


### 3.3.19 View Banner

a. Class Diagram

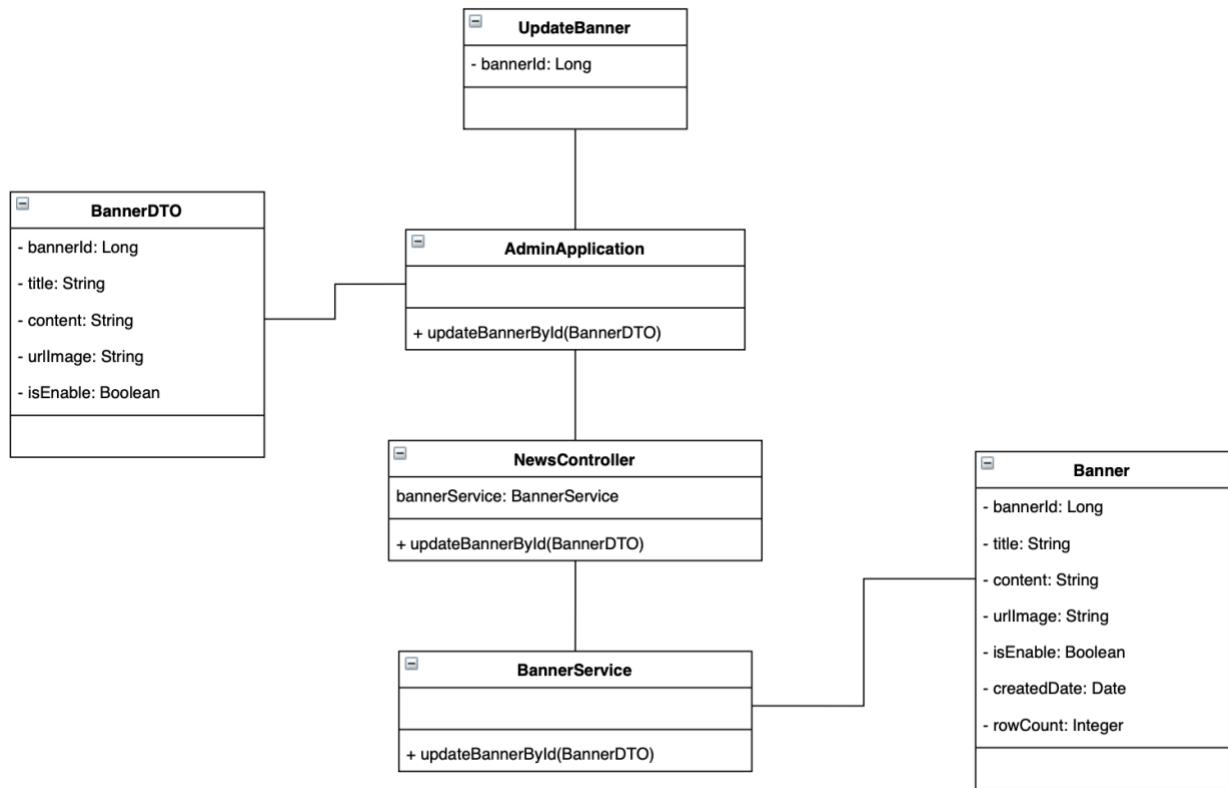


b. Sequence Diagram

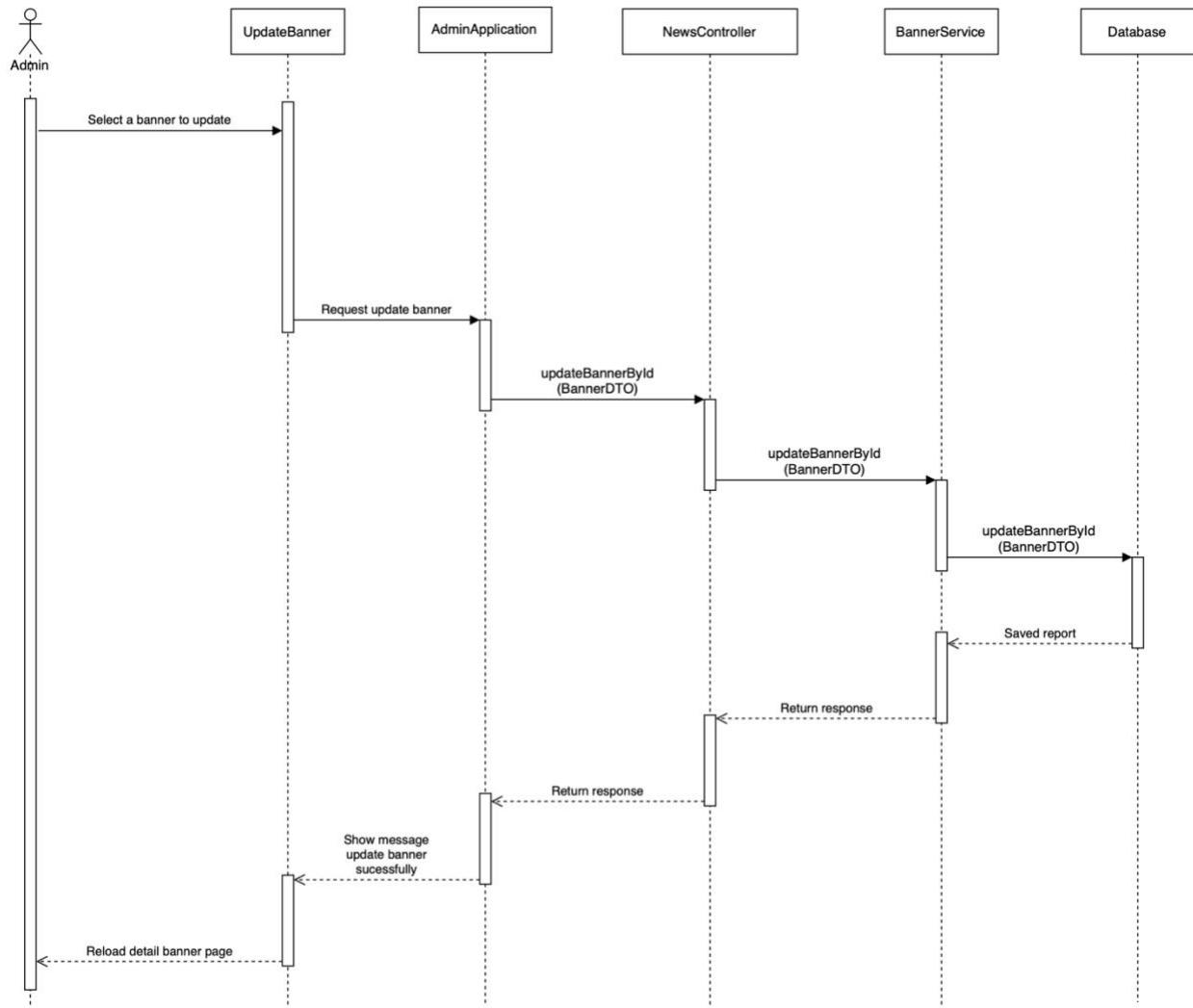


### 3.3.20 Update Banner

a. Class Diagram

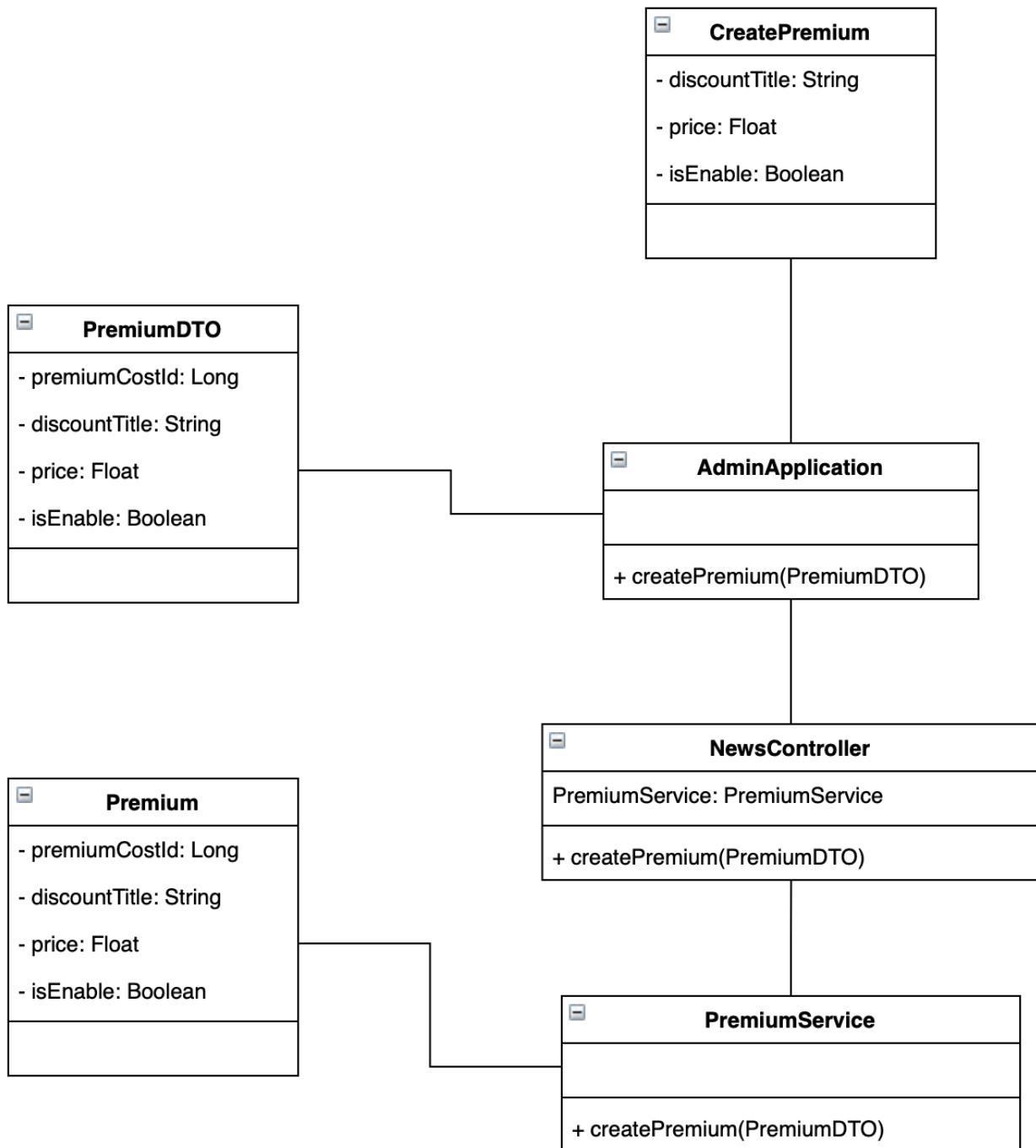


b. Sequence Diagram

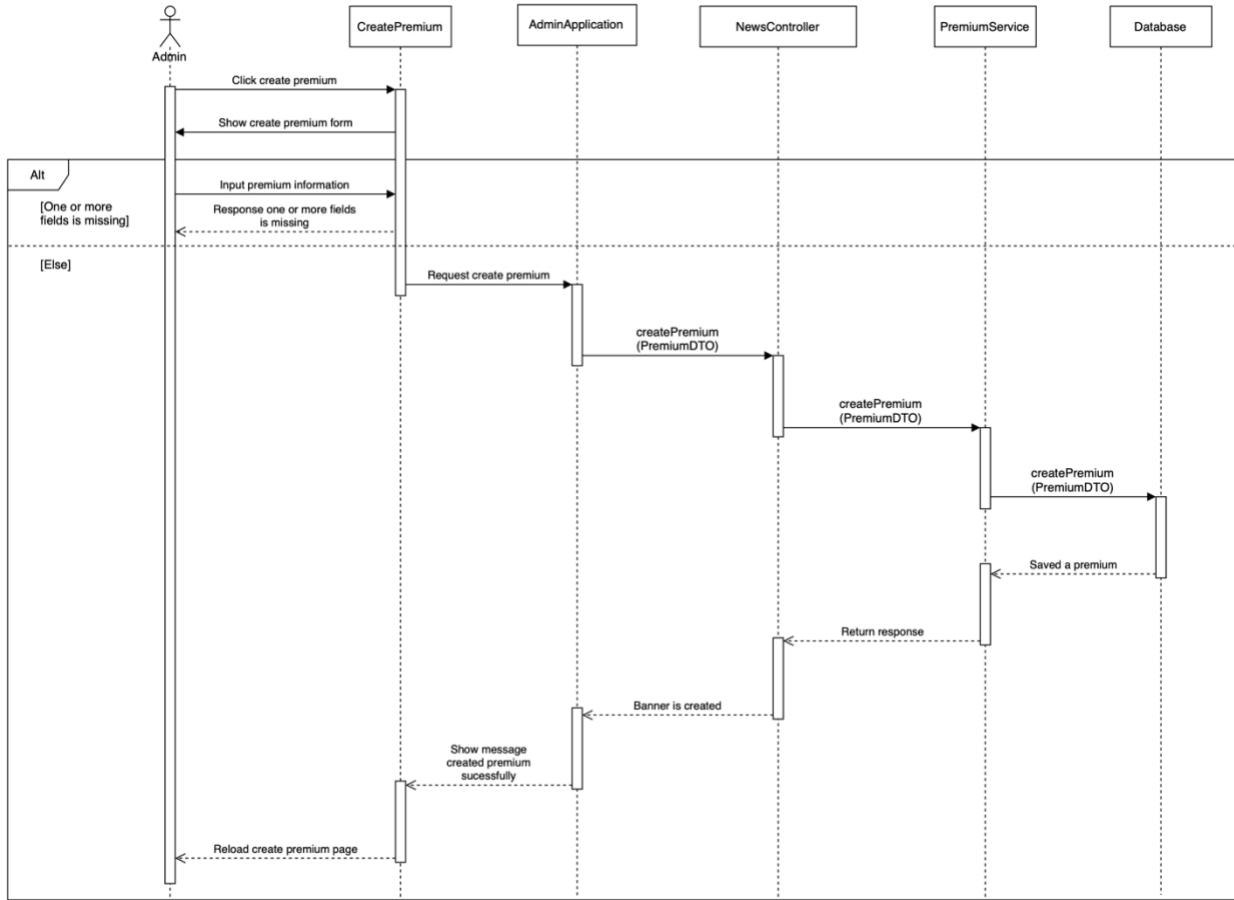


### 3.3.21 Create Premium

a. Class Diagram

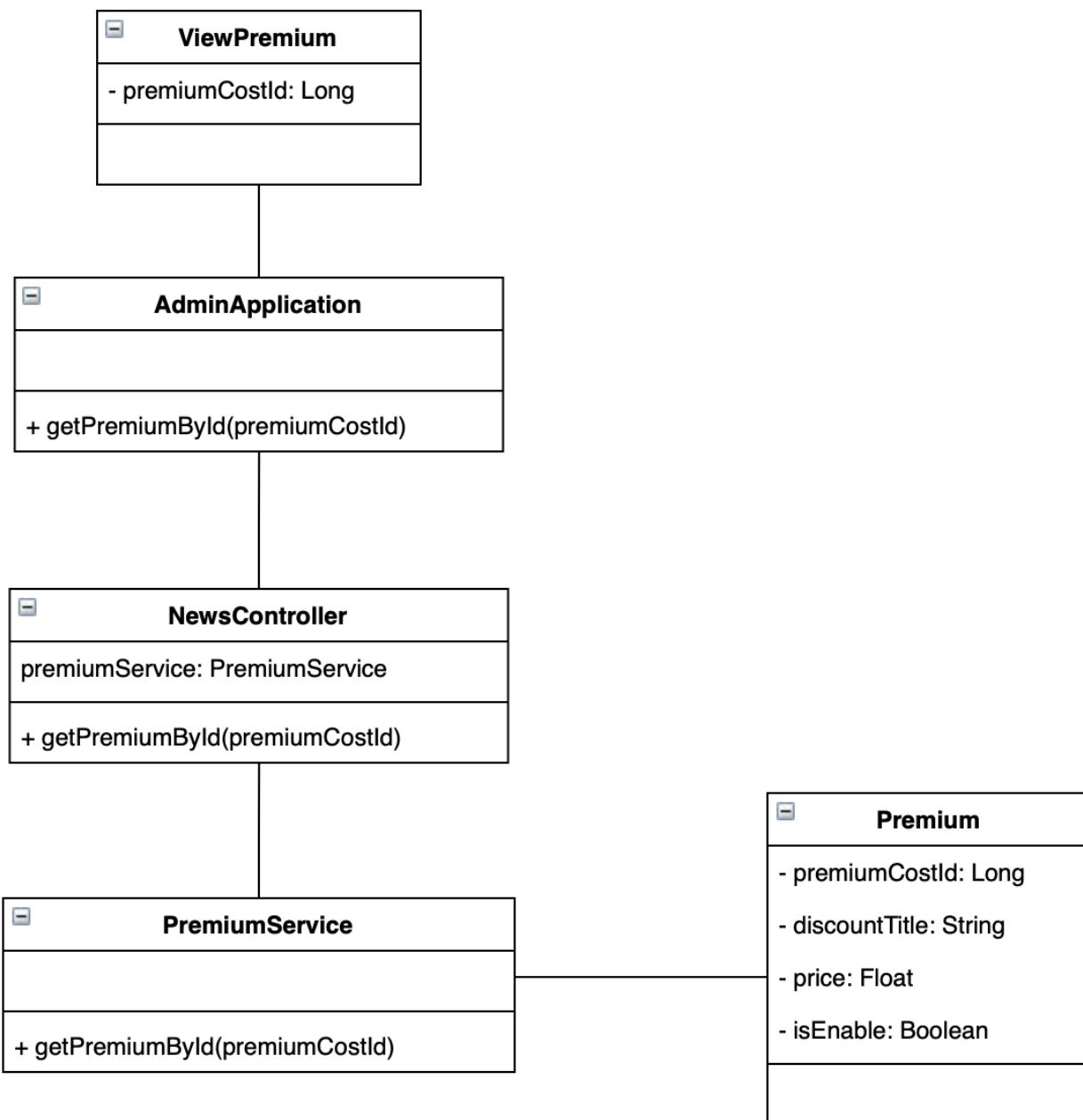


## b. Sequence Diagram

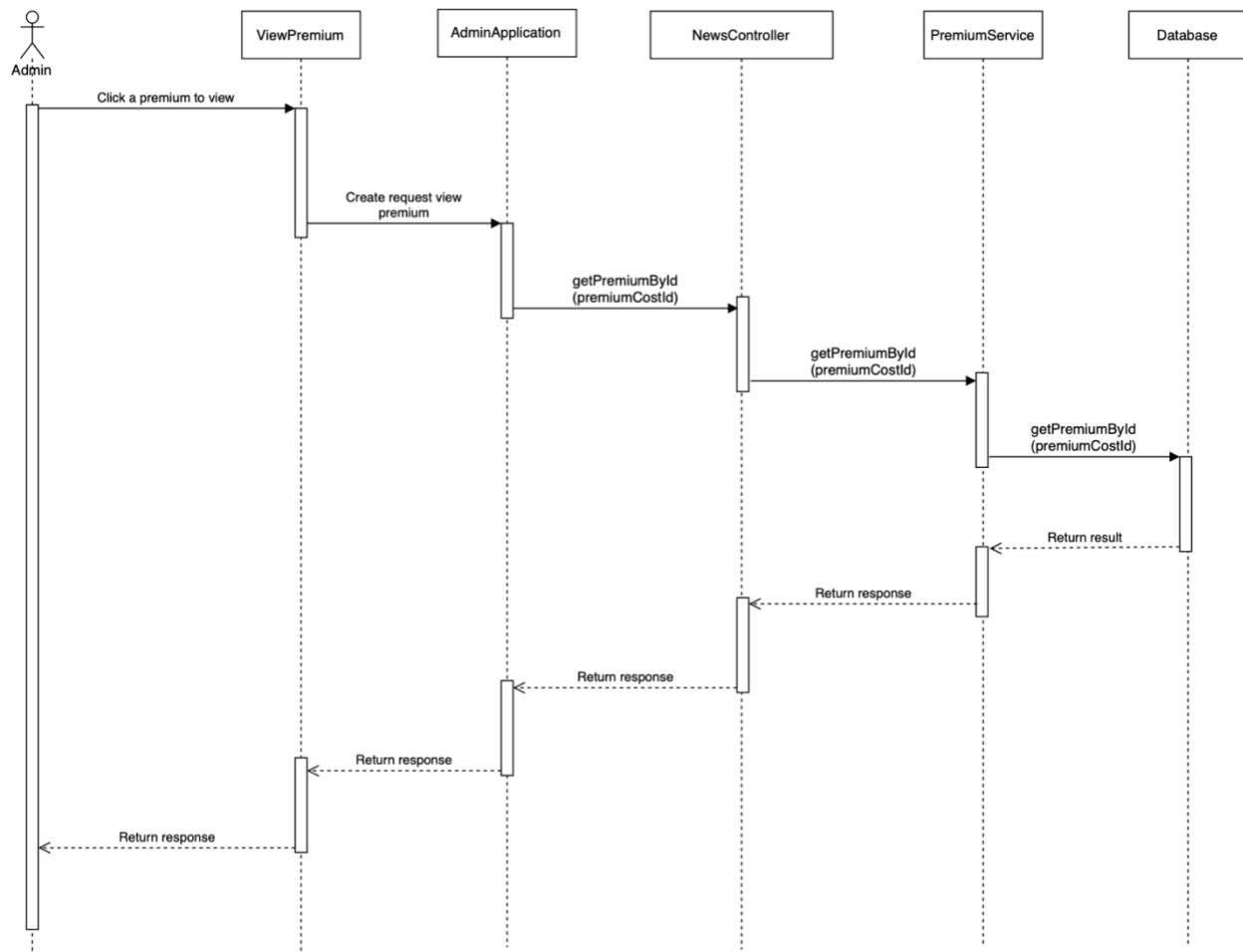


### 3.3.22 View Premium

#### a. Class Diagram

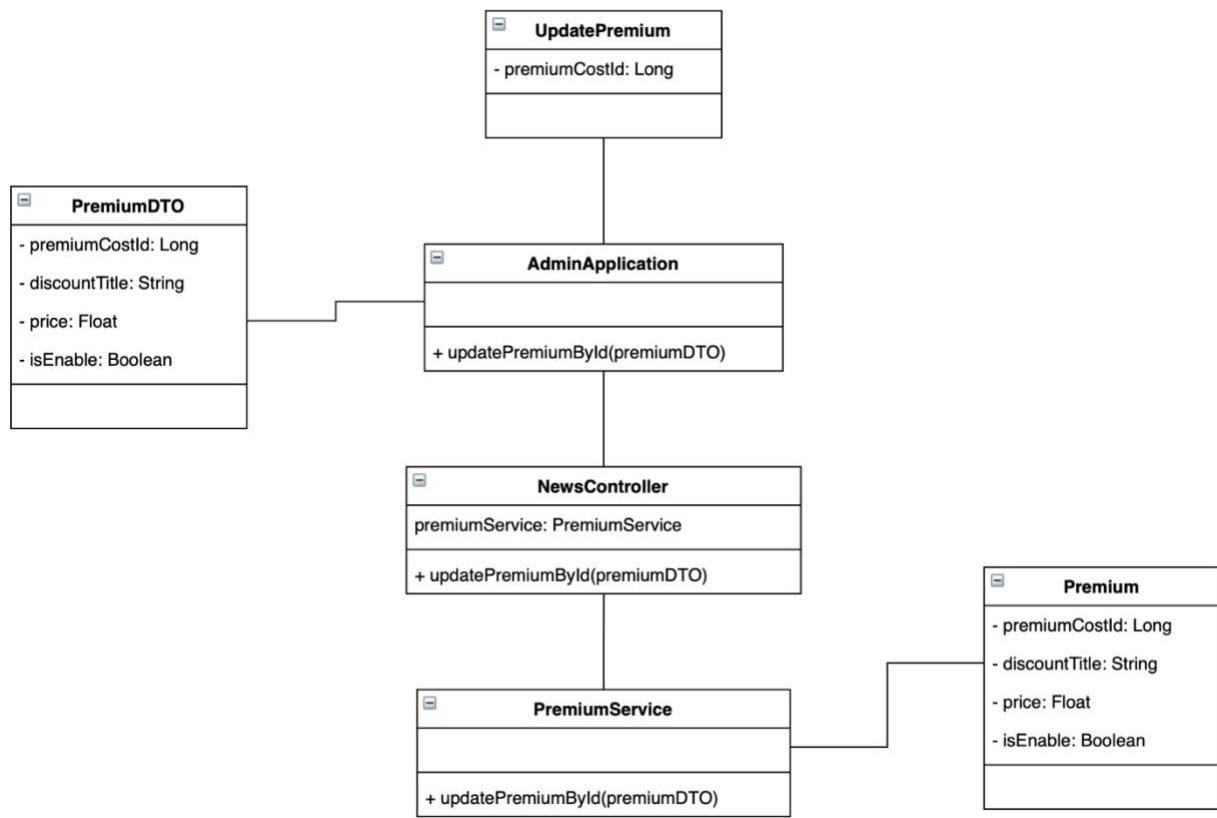


b. Sequence Diagram

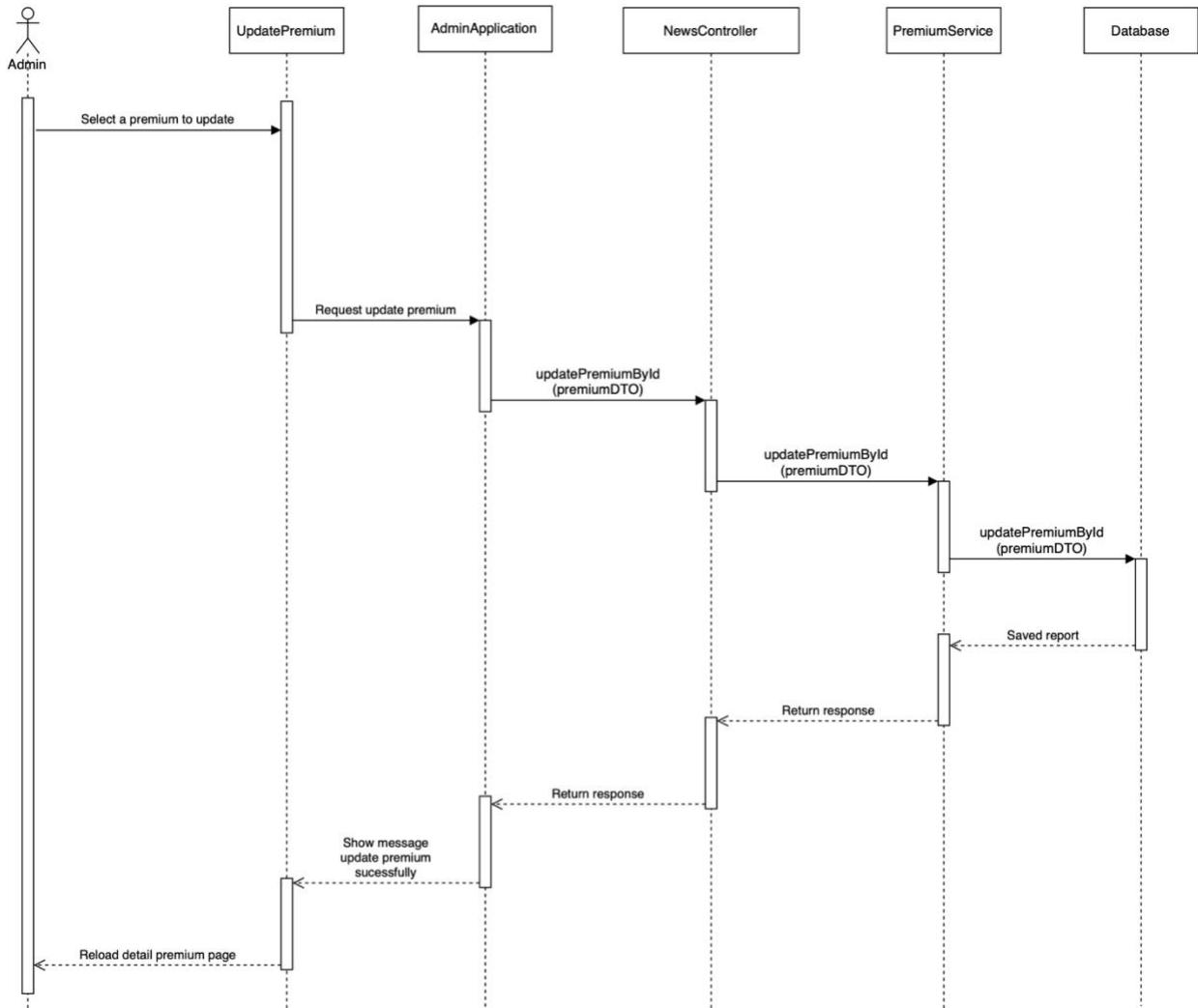


### 3.3.23 Update Premium

a. Class Diagram



b. Sequence Diagram



## 4. Class Specifications

### 4.1 SAFO Back-end

#### 4.1.1 User Controller

##### 4.1.1.1 AccountController

No	Method	Description
1	sendSms	Using to send sms.
2	getAccountById	Using to get account by id.
3	getAccountByJwtToken	Using to get account by JWT Token.

#### 4.1.1.2 AccountReportController

No	Method	Description
1	getAllReportByAccId	Using to get all reports by account id.
2	removeReportAccountById	Using to remove report account by id.
3	create	Using to create report and send to the system.

#### 4.1.1.3 CardBoxController

No	Method	Description
1	get	Using to get a card box by id.
2	save	Using to save a card box.
3	saveAll	Using to save all card boxes.
4	update	Using to update a card box.
5	getAll	Using to get all card boxes.
6	delete	Using to remove a card box by id.

#### 4.1.1.4 CardInformationController

No	Method	Description
1	updateInfor	Using to update information.
2	getInfor	Using to get information.
3	saveAllInfor	Using to save all information.
4	deleteOneInfor	Using to delete information.
5	deleteAllInfor	Using to delete all information.

#### 4.1.1.5 DefaultInforValueController

No	Method	Description
1	getAllDefaultInfor	Using to get all default information value.
2	getInfor	Using to get information.
3	saveAllInfor	Using to save all information.
4	deleteOneInfor	Using to delete information.
5	deleteAllInfor	Using to delete all information.

#### 4.1.1.6 GroupController

No	Method	Description
1	getById	Using to get a group by id.
2	getAllByAccountId	Using to get all groups by account id.
3	create	Using to create a group.
4	removeById	Using to remove a group by id.
5	updateById	Using to update a group by id.

#### 4.1.1.7 ReceiptController

No	Method	Description
1	create	Using to create a receipt.

#### 4.1.1.8 ScannedCardController

No	Method	Description
1	getAllScannedCardByAccountId	Using to get all scanned card by account id.
2	getById	Using to get a scanned card by id.
3	create	Using to create a scanned card.

4	update	Using to update a scanned card.
5	remove	Using to remove a scanned card.

#### 4.1.2 Admin Controller

##### 4.1.2.1 AccountAdminController

No	Method	Description
1	getAllAccount	Using to get all accounts of user.
2	changeStatusById	Using to change status of a account by account id.

##### 4.1.2.2 DiscountController

No	Method	Description
1	getAll	Using to get all discount in the system.
2	getById	Using to find discount by id.
3	updateDiscount	Using to update the discount.
4	createDiscount	Using to create a discount.
5	removeById	Using to remove a discount by id.

##### 4.1.2.3 EmailController

No	Method	Description
1	sendEmail	Using to send a email to user.
2	sendEmailAttachment	Using to send a email to user with attachment.
3	sendTemplateMail	Using to send a template email to user to guide user get signature email.

#### 4.1.2.4 FinanceController

No	Method	Description
1	getAllReceiptCurrent Semester	Using to get all current receipts in a semester.
2	getAllReceiptFilter	Using to get all receipts with filter.

#### 4.1.2.5 History Controller

No	Method	Description
1	getLog	Using to get log of history in the system.

#### 4.1.2.6 News Controller

No	Method	Description
1	getTop2Banner	Using to get two banners.
2	getBannerById	Using to get a banner by id.
3	getAllBanner	Using to get all banners.
4	updateBanner	Using to update a banner.
5	removeBanner	Using to remove a banner.
6	createBanner	Using to create a banner.
7	getAllPremiumCost	Using to get all premium cost.
8	getPremiumById	Using to get premium by id.
9	updatePremium	Using to update premium.
10	removePremium	Using to remove premium.
11	create	Using to create premium.

#### 4.1.2.7 ReportController

No	Method	Description
1	getAllReportAccount	Using to get all reports of a account.
2	getAllReport	Using to get all reports.
3	getByReportId	Using to get a report by id.
4	updateReportAccount	Using to update a report.
5	removeReportAccountById	Using to remove a report by id.

### 4.1.3 Common Controller

#### 4.1.3.1 CredentialController

No	Method	Description
1	login	Using to login into the system.
2	genRefreshToken	Using to generate refresh token.
3	signUp	Using to sign up into the system.
4	signout	Using to sign out the system.
5	checkExistPhoneToSignUp	Using to check existed phone number to sign up.
6	verifyOtp	Using to verify OTP code.
7	resetPassword	Using to reset password.
8	changePassword	Using to change password.

#### 4.1.3.2 ImageModelController

No	Method	Description
1	uploadImage	Using to upload a image.
2	getImage	Using to get images.

#### 4.1.3.3 PrivateController

No	Method	Description
1	accessPrivate	Using to access private.

#### 4.1.4 Service

##### 4.1.4.1 AccountReportService

No	Method	Description
1	update	Using to save the user.
2	getAllByAccountId	Using to save the information into the database.
3	deleteById	Using to reject a report by report id.
4	save	Using to save a report.
5	getAllByAccountId	Using to get all report by account id.
6	getById	Using to get a report by id.

##### 4.1.4.2 AccountService

No	Method	Description
1	getAll	Using to get all informations of a account.
2	changeStatusById	Using to change status by account id.
3	lockById	Using to lock the account by id.
4	login	Using to login into the system.
5	signUp	Using to sign up the system.
6	resetPassword	Using to reset password.
7	changePassword	Using to change password.
8	checkExistPhoneNumber	Using to check existed phone number.
9	verifyOtp	Using to verify OTP.
10	genTokenByRefresh	Using to generate token by refresh.

11	findByUsername	Using to find by username.
12	findByDetailToken	Using to find by detail token.
13	findCurrentUserByJwtToken	Using to find current user by JWT token.
14	returnRoleByJwtToken	Using to return the role by Jwt token.
15	isAdmin	Using to check a account is admin or not.
16	returnId	Using to return id.

#### 4.1.4.3 AdminLogService

No	Method	Description
1	createLog	Using to create log.
2	getAll	Using to get all log in the system.
3	logCreate	Using to get all log with the type is CREATE.
4	logUpdate	Using to get all log with the type is UPDATE.
5	logRemove	Using to get all log with the type is REMOVE.
6	logGet	Using to get all log with the type is GET.

#### 4.1.4.4 BannerService

No	Method	Description
1	getAll	Using to get all banners in the system.
2	getTopNewBanners	Using to get top new banners.
3	removeById	Using to remove a banner by id.
4	update	Using to update a banner.
5	save	Using to save a banner.
6	getById	Using to get a banner by id.

#### 4.1.4.5 BaseService

No	Method	Description
1	getById	Using to get by id.
2	getAll	Using to get all.
3	save	Using to save.
4	update	Using to update.
5	removeById	Using to remove by id.

#### 4.1.4.6 CardBoxService

No	Method	Description
1	saveAll	Using to save all information of a card box.
2	getPaging	Using to get paging.
3	deleteAll	Using to delete all information.
4	getAllByAccountId	Using to get all by account id.

#### 4.1.4.7 CardInformationService

No	Method	Description
1	getByDataTag	Using to get card box by data tag.
2	getInforByCardBoxId	Using to get information by card box id.
3	updateByCardBoxId	Using to update by card box id.
4	createInfor	Using to create information.
5	removeInforByCardBoxId	Using to remove information by card box id.

#### 4.1.4.8 DefaultInforValueService

No	Method	Description
1	getAccountByEmail	Using to get account by email.

2	getAllDefaultInforById	Using to get all default information by id.
---	------------------------	---

#### 4.1.4.9 DiscountService

No	Method	Description
1	getAll	Using to get all discounts.
2	getById	Using to get a discount by id.
3	removeById	Using to remove a discount by id.
4	update	Using to update a discount.
5	save	Using to save a discount.

#### 4.1.4.10 FinanceService

No	Method	Description
1	getAll	Using to get all finances.
2	createReceipt	Using to create a receipt.
3	createPremiumReceipt	Using to create a premium receipt.
4	createTemplateReceipt	Using to create a template receipt.
5	getAllPremiumReceipt	Using to get all premium receipts.
6	getAllTemplateReceipt	Using to get all template receipts.

#### 4.1.4.11 GroupService

No	Method	Description
1	updateTitleById	Using to update title of group by group id.
2	getAllByAccountId	Using to get all groups by account id.

#### 4.1.4.12 ImageService

No	Method	Description
1	save	Using to save a image.
2	getImageBase64	Using to get a image base 64.
3	getImageWithMediaType	Using to get a image with media type.

#### 4.1.4.13 PremiumCostService

No	Method	Description
1	save	Using to save a premium cost.
2	getAllPremiumCost	Using to get all premium cost.
3	getById	Using to get premium by id.
4	update	Using to update premium.
5	removeById	Using to remove premium by id.

#### 4.1.4.14 RefreshTokenService

No	Method	Description
1	generateRefreshToken	Using to generate refresh token.
2	validateRefreshToken	Using to validate refresh token.
3	deleteRefreshToken	Using to delete refresh token.
4	checkExistRefreshToken	Using to check exist refresh token.
5	getRefreshToken	Using to get refresh token.
6	returnRefreshToken	Using to return refresh token.

#### 4.1.4.15 ScannedCardService

No	Method	Description
1	getAllByAccountId	Using to get all scanned card by account id.

#### 4.1.4.16 SmsService

No	Method	Description
1	sendSms	Using to send a message.

#### 4.1.4.17 TemplateCardService

No	Method	Description
1	getAll	Using to get all template cards.

## 4.2 SAFO Front-end

### 4.2.1 SAFO User

No	Method	Description
1	isPhoneLinkedAccountExist	Using to check the phone is linked with account exist.
2	getAccountByJWT	Using to get account by JWT
3	signInUsingAccount	Using to sign in into the system.
4	getRefreshJWT	Using to get refresh JWT.
5	signUpUsingAccount	Using to sign up account to sign in into the system.
6	doSignOut	Using to sign out the system.
7	onAPIResponseDefault	Using to get on API response default.
8	onAPIRequestFail	Using to get on API request fail.
9	createDefaultCardBox	Using to create default card box.
10	createCardBox	Using to create a card box.

11	getAllCardBox	Using to get all card boxes.
12	getAllDataTagConfig	Using to get all data tag config.
13	getAllDefaultInformationById	Using to get all default information by id.
14	updateDefaultInformation	Using to update default information.
15	getAllGroupFromUser	Using to get all groups from user.
16	getAllScannedCardFromUser	Using to get all scanned cards from user.
17	verifyOTP	Using to verify OTP code.
18	resendOTP	Using to resend OTP code.
19	getHotTemplateCard	Using to get all hot template cards.
20	getTopTemplateCard	Using to get all top template cards.
21	getEndPointBuilder	Using to get end point builder.
22	buildJson	Using to build json.
23	makeAPI	Using to make API.
24	loadCachedAccount	Using to load cached account.
25	getLoginJWTCredential	Using to get login JWT credential.
26	saveJWTCredential	Using to save JWT credential.
27	alternativeAction	Using to alter action.
28	isValid	Using to check valid phone number.
29	showAllIntent	Using to show all intents which helps Debug.
30	getLanguage	Using to get language.
31	setLocale	Using to set locale.
32	getPersistedData	Using to get persisted data.
33	updateResources	Using to update locale resources.

34	updateResourcesLegacy	Using to update locale resources legacy.
35	convertFormatPhoneToNormal	Using to convert format phone number to normal.

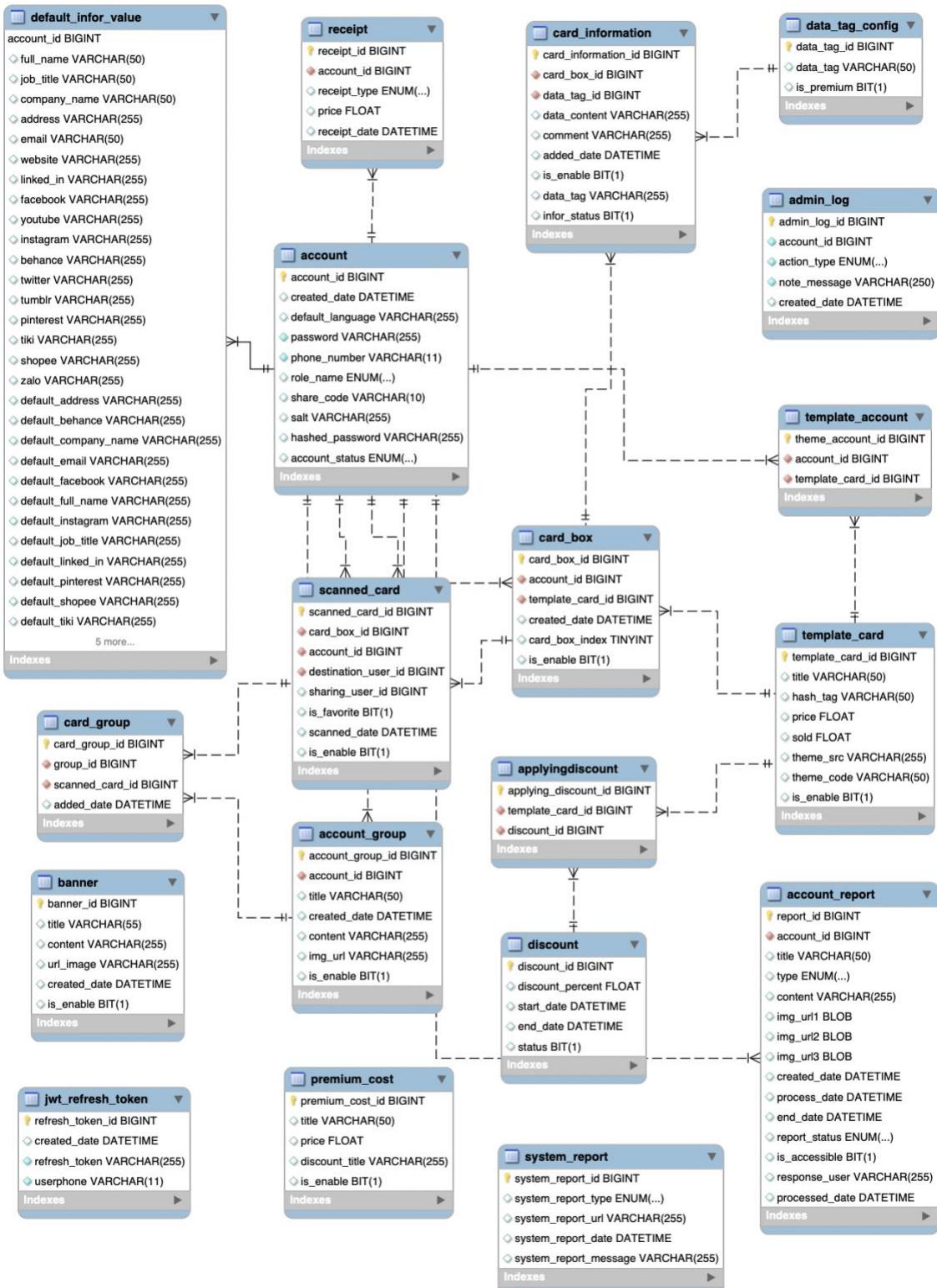
#### 4.2.2 SAFO Admin

No	Method	Description
1	paginate	Using to paginate.
2	changeStatus	Using to change a account status.
3	getInforAccount	Using to get information of a account.
4	createBanner	Using to create a banner.
5	updateBanner	Using to update a banner.
6	deleteBanner	Using to delete a banner by id.
7	getInforBanner	Using to get information of a banner.
8	createCard	Using to create a template card.
9	updateCard	Using to update a template card.
10	deleteCard	Using to delete a template card.
11	getInforCard	Using to get information of a template card.
12	createDiscount	Using to create a discount.
13	updateDiscount	Using to update a discount.
14	deleteDiscount	Using to delete a discount.
15	getInforDiscount	Using to get information of a discount.
16	getInforHistory	Using to get information of histories.
17	getInforReceipt	Using to get information of receipts.
18	updateReport	Using to update a report.
19	getInforReport	Using to get information of a report.

20	checkFormValid	Using to check form valid.
21	patchForm	Using to patch form.
22	bindEnumToOptions	Using to bind enum to options.
23	patchSearchFormToPaginateOption	Using to patch search form to paginate option.
24	bindDateCriteriaValueToDateSearch	Using to bind date criteria value to date search.
25	removeAllDateCriteria	Using to remove all date criteria.
26	bindDateCriteriaValueToPaginateOption	Using to bind date criteria value to paginate option.
27	passwordMatchValidator	Using to check password match validator.
28	customRequiredValidator	Using to custom required validator.
29	requiredNumberValidator	Using to required number validator.

# 5. Data & Database Design

## 5.1 Database Design



### 5.1.1. Account

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	accountId	Long		Y	N		Auto-generation id.
2	roleName	Enum		N	Y		Role of account.
3	phoneNumbe	String		N	N		Phone number of account.
4	password	String		N	N		Password of account.
5	createdDate	Date		N	Y		Date time of the created account.
6	defaultLanguage	String		N	Y		Set default language for the account.
7	salt	String		N	Y		Salt of password.
8	hashedPassword	String		N	Y		Hash of password.
9	accountStatus	Enum		N	Y		Status of account.

### 5.1.2. AccountGroup

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	accountGroupId	Long		Y	N		Auto-generation id.
2	account	Object		N	N		Account of group.
3	title	String		N	Y		Title of group.
4	createdDate	Date		N	Y		Date time of the created group.
5	isEnable	Boolean		N	Y		Account group is enable or not.

### 5.1.3. AccountReport

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	userReportId	Long		Y	N		Auto-generation id.
2	account	Object		N	N		Account of report.
3	type	Enum		N	N		Type of report.
4	content	String		N	N		Content of report.
5	imgUrl1	String		N	Y		Image of report.
6	imgUrl2	String		N	Y		Image of report.
7	imgUrl3	String		N	Y		Image of report.
8	createdDate	Date		N	Y		Date time of the created report.
9	processDate	Date		N	Y		Date time of the process report.

10	endDate	Date		N	Y		Date time of the end report.
11	reportStatus	Enum		N	N		Status of report.
12	responseUser	String		N	Y		Response of user.
13	accessible	Boolean		N	Y		Accessible status of report.

#### 5.1.4. AdminLog

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	adminLogId	Long		Y	N		Auto-generation id.
2	accountId	Long		N	N		Account id of a log.
3	actionType	Enum		N	N		Action type of a log.
4	noteMessage	String		N	Y		Note message of a log.
5	createdDate	Date		N	Y		Date time of the created log.

#### 5.1.5. ApplyingDiscount

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	applyingDiscountId	Long		Y	N		Auto-generation id.
2	templateCardId	Long		N	N		Template card of discount.
3	discountId	Long		N	N		Discount which apply to the template card.
4	startDate	Date		N	Y		Date time of the start discount.
5	endDate	Date		N	Y		Date time of the end discount.

#### 5.1.6. Banner

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	bannerId	Long		Y	N		Auto-generation id.
2	title	String		N	N		Title of banner.
3	content	String		N	Y		Content of banner.
4	urlImage	String		N	N		Url image of banner.
5	createdDate	Date		N	Y		Date time of the created banner.
6	isEnabled	Boolean		N	N		Banner is enable or not.

### 5.1.7. CardBox

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	cardBoxID	Long		Y	N		Auto-generation id.
2	createdDate	Date		N	Y		Date time of the created card box.
3	cardBoxIndex	Number		N	Y		Index of card box.
4	templateCard	Object		N	N		Template card of card box.
5	account	Object		N	N		Account of card box.
6	informationList	Array		N	N		Information list of card box.
7	isEnabled	Boolean		N	N		Card box is enable or not.

### 5.1.8. Card Group

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	cardGroupId	Long		Y	N		Auto-generation id.
2	groupId	Long		N	N		Group id of card group.
3	scannedCardId	Long		N	N		Scanned card id of card group.
4	addedDate	Date		N	Y		Date time of add card group.

### 5.1.9. Card Information

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	cardInformationId	Long		Y	N		Auto-generation id.
2	cardBox	Object		N	N		Card box of information box.
3	dataTagConfig	Object		N	N		Date tag config of card information.
4	dataContent	String		N	N		Data content of card information.
5	comment	String		N	Y		Comment of card information.
6	addedDate	Date		N	Y		Date time of the added card information.
7	isEnabled	Boolean		N	Y		Card information is enable or not.

### 5.1.10. DataTagConfig

#	Field name	Type	Size	Unique	Nullable	PK/FK	Notes
1	cardInformationId	Long		Y	N		Auto-generation id.
2	cardBox	Object		N	N		Card box of information box.
3	dataTagConfig	Object		N	N		Date tag config of card information.
4	dataContent	String		N	N		Data content of card information.
5	comment	String		N	Y		Comment of card information.
6	addedDate	Date		N	Y		Date time of the added card information.
7	isEnable	Boolean		N	Y		Card information is enable or not.