

**DENTIST BOOKING SYSTEM**

**Software Design Document**

– Tp Ho Chi Minh, August 2019 –

**Table of Contents**

[I. Overview](#_heading=h.gjdgxs) 10

1. Code Packages/Namespaces10

[2. Database Schema](#_heading=h.1fob9te) 11

[II. Code Designs](#_heading=h.3znysh7) 12

1. Code Packages/Namespaces12

[a. Class Diagram](#_heading=h.tyjcwt) 12

[b. Class Specifications](#_heading=h.3dy6vkm) 13

[c. Sequence Diagram(s)](#_heading=h.1t3h5sf) 14

[d. Database queries](#_heading=h.4d34og8) 14

[2.](#_heading=h.2s8eyo1) Authentication / Forget Password15

a. Class Diagram15

b. Class Specifications15

c. Sequence Diagram(s)16

d. Database queries16

3. Authentication /Register17

a. Class Diagram17

b. Class Specifications17

c. Sequence Diagram(s)18

d. Database queries18

4. User Information /View Profile19

a. Class Diagram19

b. Class Specifications19

c. Sequence Diagram(s)19

d. Database queries20

5. User Information /Edit View Profile21

a. Class Diagram21

b. Class Specifications21

c. Sequence Diagram(s)22

d. Database queries22

6.News Management **/** View News information23

a. Class Diagram23

b. Class Specifications23

c. Sequence Diagram(s)23

d. Database queries24

7. News Management / View News Detail25

a. Class Diagram25

b. Class Specifications25

c. Sequence Diagram(s)25

d. Database queries26

8. News Management / View News Detail26

a. Class Diagram26

b. Class Specifications26

c. Sequence Diagram(s)27

d. Database queries27

9.Dentist Information **/** View dentist information28

a. Class Diagram28

b. Class Specifications28

c. Sequence Diagram(s)28

d. Database queries29

10.Dentist Information **/** View dentist detail29

a. Class Diagram29

b. Class Specifications30

c. Sequence Diagram(s)30

d. Database queries30

11. Service Management / View list Service (chưa có bảng)31

a. Class Diagram31

b. Class Specifications31

c. Sequence Diagram(s)31

d. Database queries32

12.Service Management **/** Search Service (chưa có bảng)32

a. Class Diagram32

b. Class Specifications33

c. Sequence Diagram(s)33

d. Database queries33

13.Service Management **/** Delete Service (chưa có bảng)34

a. Class Diagram34

b. Class Specifications34

c. Sequence Diagram(s)34

d. Database queries35

14.Service Management **/** Add new Service (chưa có bảng)36

a. Class Diagram36

b. Class Specifications36

c. Sequence Diagram(s)36

d. Database queries37

15.Service Management **/** Edit Service (chưa có bảng)38

a. Class Diagram38

b. Class Specifications38

c. Sequence Diagram(s)38

d. Database queries39

16.Service Management **/** View Service Detail (chưa có bảng)39

a. Class Diagram39

b. Class Specifications40

c. Sequence Diagram(s)40

d. Database queries40

17.Booking Management **/** View Booking schedule (chưa có bảng)41

a. Class Diagram41

b. Class Specifications41

c. Sequence Diagram(s)41

d. Database queries42

18.Booking Management **/** Add new Booking (chưa có bảng)42

a. Class Diagram42

b. Class Specifications43

c. Sequence Diagram(s)43

d. Database queries43

19.Booking Management **/** Add service to the Booking (chưa có bảng)44

a. Class Diagram44

b. Class Specifications44

c. Sequence Diagram(s)44

d. Database queries45

20.Booking Management **/** Edit Booking (chưa có bảng)46

a. Class Diagram46

b. Class Specifications46

c. Sequence Diagram(s)46

d. Database queries47

21.Booking Management **/** View History Booking Info (chưa có bảng)48

a. Class Diagram48

b. Class Specifications48

c. Sequence Diagram(s)48

d. Database queries49

22.Booking Management **/** Search History Booking 49

a. Class Diagram49

b. Class Specifications50

c. Sequence Diagram(s)50

d. Database queries50

23.Announcement **/** View Notification (chưa có bảng)51

a. Class Diagram51

b. Class Specifications51

c. Sequence Diagram(s)51

d. Database queries52

24.Announcement **/** Delete Notification 53

a. Class Diagram53

b. Class Specifications53

c. Sequence Diagram(s)53

d. Database queries54

25.Booking Management **/** Check Booking Info54

a. Class Diagram54

b. Class Specifications54

c. Sequence Diagram(s)55

d. Database queries55

26.Booking Management **/** Confirm Booking 56

a. Class Diagram56

b. Class Specifications56

c. Sequence Diagram(s)56

d. Database queries57

27.Booking Management **/** Export Booking Info 57

a. Class Diagram57

b. Class Specifications57

c. Sequence Diagram(s)58

d. Database queries58

28.Booking Management **/** Export Booking Info 59

a. Class Diagram59

b. Class Specifications59

c. Sequence Diagram(s)59

d. Database queries60

29.Clinic Management **/** Add Clinic 60

a. Class Diagram60

b. Class Specifications61

c. Sequence Diagram(s)61

d. Database queries61

30.Clinic Management **/** Update Clinic 62

a. Class Diagram62

b. Class Specifications62

c. Sequence Diagram(s)63

d. Database queries63

31.Clinic Management **/** Delete Clinic63

a. Class Diagram64

b. Class Specifications64

c. Sequence Diagram(s)64

d. Database queries64

32.Clinic Management **/** Search Clinic 65

a. Class Diagram65

b. Class Specifications65

c. Sequence Diagram(s)65

d. Database queries66

33.Customer Service**/** Write Feedback (Thiếu controller)66

a. Class Diagram66

b. Class Specifications67

c. Sequence Diagram(s)67

d. Database queries67

34.Customer Service**/** Update Feedback (Thiếu controller)68

a. Class Diagram68

b. Class Specifications68

c. Sequence Diagram(s)68

d. Database queries69

35.Customer Service**/** Delete Feedback (Thiếu controller)69

a. Class Diagram69

b. Class Specifications70

c. Sequence Diagram(s)70

d. Database queries70

36.User Management**/** View Feedback (Thiếu controller)71

a. Class Diagram71

b. Class Specifications71

c. Sequence Diagram(s)71

d. Database queries72

37.User Management **/** Search Feedback (Thiếu controller)72

a. Class Diagram72

b. Class Specifications73

c. Sequence Diagram(s)73

d. Database queries73

38.Customer Service**/** Chat with staff (Thiếu controller, bảng)74

a. Class Diagram74

b. Class Specifications74

c. Sequence Diagram(s)74

d. Database queries75

39.Customer Service**/** Reply customers (Thiếu controller, bảng)75

a. Class Diagram75

b. Class Specifications75

c. Sequence Diagram(s)76

d. Database queries76

40.User Management**/** Block Account(Thiếu controller, bảng)77

a. Class Diagram77

b. Class Specifications77

c. Sequence Diagram(s)77

d. Database queries78

41.User Management**/** List of statistics (Thiếu controller, bảng)78

a. Class Diagram78

b. Class Specifications79

c. Sequence Diagram(s)79

d. Database queries79

42.User Management**/** List of user register (Thiếu controller, bảng)80

a. Class Diagram80

b. Class Specifications80

c. Sequence Diagram(s)80

d. Database queries81

43.User Management**/** Export file excel (Thiếu controller, bảng)81

a. Class Diagram81

b. Class Specifications82

c. Sequence Diagram(s)82

d. Database queries82

44.Overview Information**/** View Introduction and overview information (Thiếu controller, bảng)83

a. Class Diagram83

b. Class Specifications83

c. Sequence Diagram(s)83

d. Database queries84

45.Overview Information**/** Search Information (Thiếu controller, bảng)84

a. Class Diagram85

b. Class Specifications85

c. Sequence Diagram(s)85

d. Database queries85

2. <Feature/Function Name1>)86

a. Class Diagram86

b. Class Specifications86

c. Sequence Diagram(s)86

d. Database queries87

2. <Feature/Function Name2>87

[III. Database Tables](#_heading=h.17dp8vu) 88

1. Employee88

2. Account88

3. RoomChat89

4. Feedback89

5. Booking89

6. Notify89

7. MessageChat89

8. Hospital90

9. BookingService90

10. Services90

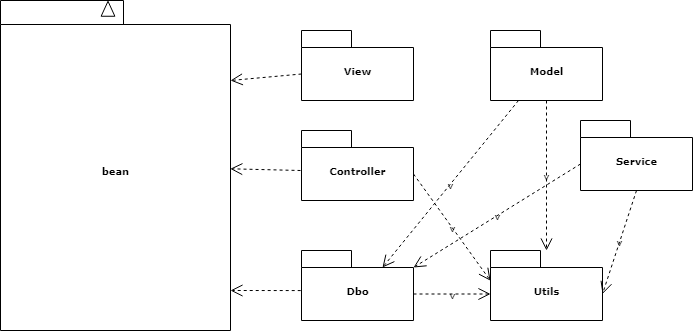
11. Slot91

12. ServiceType92

13. ServiceSlot92

# I. Overview

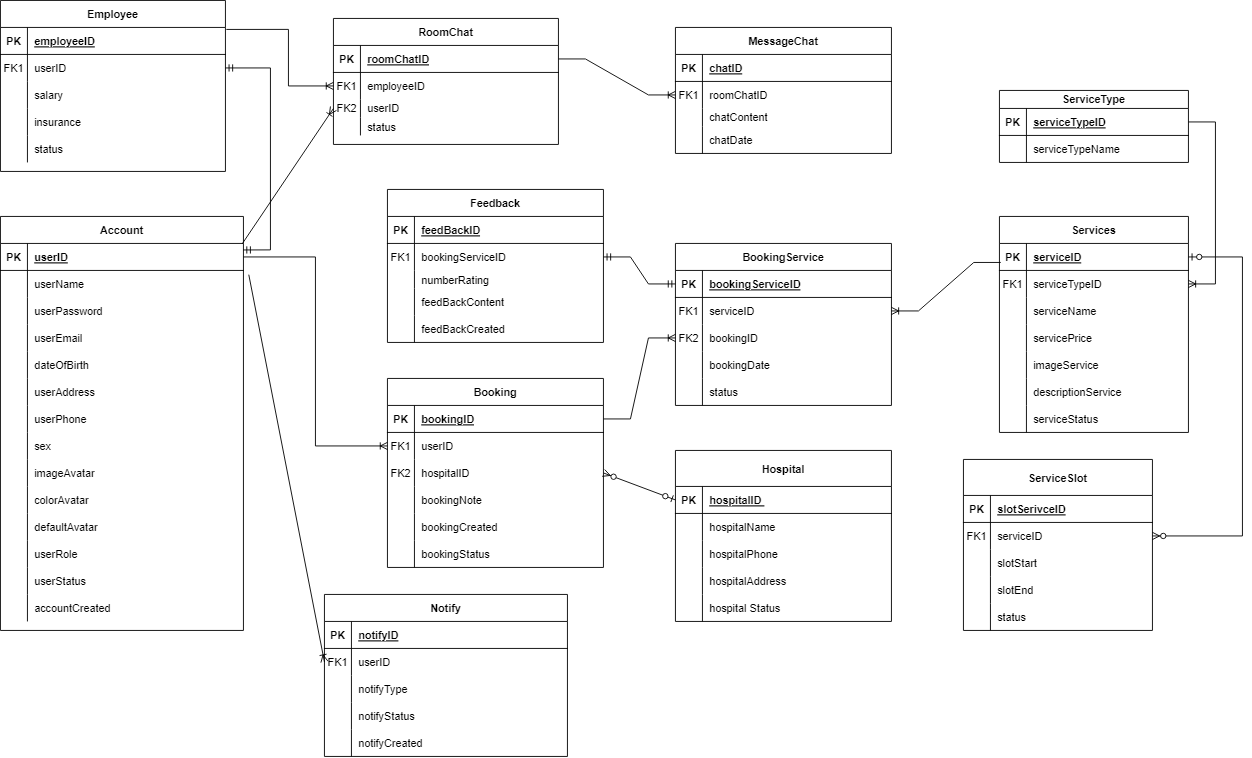
## 1. Code Packages/Namespaces



***Package descriptions & package class naming conventions***

| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *View* | *Where to store jsp files about the user interface*  *<Class naming convention>* |
| *02* | *Model* | *Where to store all model file* |
| *03* | *Controller* | *Where to store all controller servlet*  *<Class naming convention>* |
| *04* | *Dbo* | *Where to store all dbo of the service and user.* |
| *05* | *Utils* | *Where to store the connection which database.*  *<Class naming convention>* |
| *06* | *Service* | *Where to store all service for the service and user.* |

## 2. Database Schema



***Table descriptions & package class naming conventions are as below***

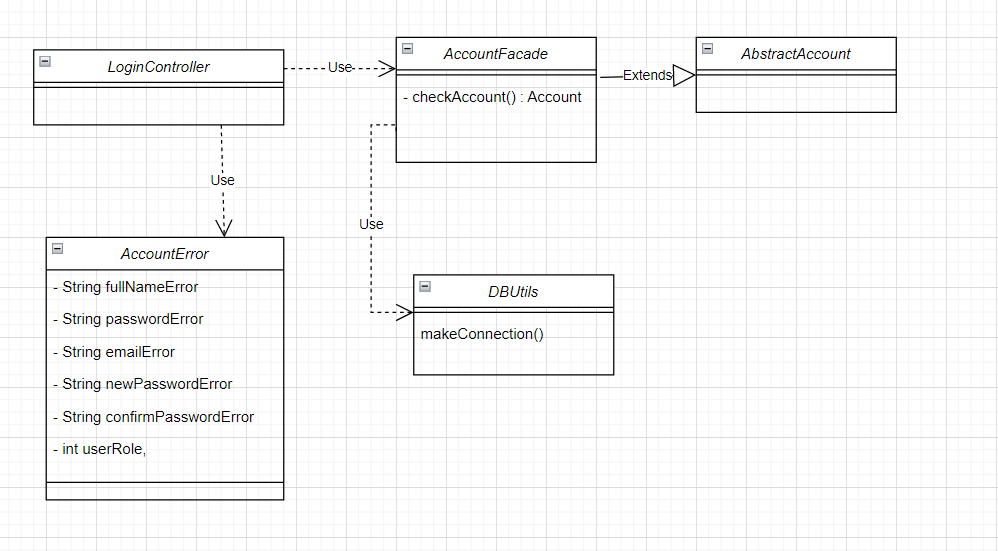
| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *Account* | *Save account as well as customer information it also saves employee information.*  *- Primary keys: UserID* |
| *02* | *Notify* | *Where to store notices for customers*  *- Primary keys: NotifyID*  *- Foreign keys: UserID* |
| *03* | *Booking* | *Store information about the booking.*  *- Primary keys: BookingID*  *- Foreign keys: UserID*  *- Foreign keys: HospitalID* |
| *04* | *Hospital* | *Store information about the hospital.*  *- Primary keys: HospitalID* |
| *05* | *BookingService* | *Store information about the booking details.*  *- Primary keys: BookingServiceID*  *- Foreign keys: ServiceID*  *- Foreign keys: BookingID*  *- Foreign keys: SlotServiceID* |
| *06* | *Services* | *Store information about the services of the system.*  *- Primary keys: ServiceID*  *- Foreign keys: ServiceTypeID* |
| *07* | *SerivcesTypes* | *Store information about the type of service.*  *- Primary keys: ServiceTypeID* |
| *08* | *ServiceSlot* | *Store information about the slot of each service.*  *- Primary keys: SlotServiceID*  *- Foreign keys: ServiceID*  *- Foreign keys: SlotID* |
| *09* | *Slot* | *Store information about the slot that available in the clinic.*  *- Primary keys: SlotID* |
| *10* | *Feedback* | *Store the feedback of the customer about our service.*  *-Primary keys: FeedBackID*  *-Foreign keys:BookingService* |
| *11* | *Employee* | *Store information about the employee*  *-Primary keys: EmployeeID*  *-Foreign keys : UserID* |
| *12* | *RoomChat* | *The room chat between Employee and Customer*  *-Primary keys: RoomChatID*  *-Foreign keys: EmployeeID*  *-Foreign keys: UserID* |
| *13* | *Message* | *Store all the message chat between Employee and Customer*  *-Primary keys :ChatID*  *-Foreign keys :RoomChatID* |

# 

# II. Code Designs

## 1. Authentication /Login

### a. Class Diagram

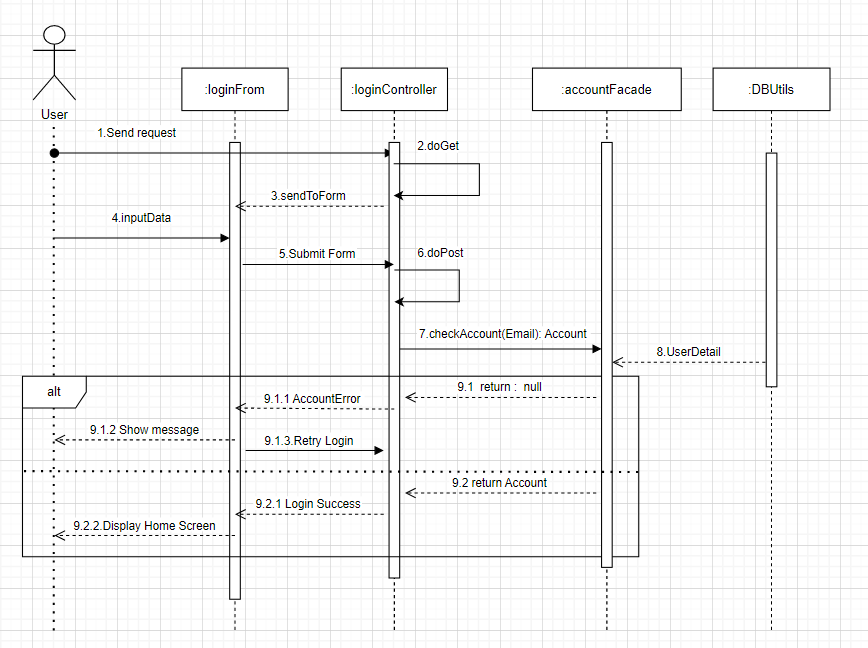


### b. Class Specifications

#### AccountFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoAccountFromSQL* | *get information of account, input ResultSet resultSet, output Account* |
| *02* | *registerAccount* | *create account, input (Connection connection, Account account), output String* |
| *03* | *updateAccount* | *update account, input (Connection connection, Account account, Object object),*  *output (true or false)* |
| *04* | *getAccount* | *get account, input (Connection connection, Object object, Object action, Object status),*  *output (List<Account>)* |
| *05* | *checkAccount* | *check account, input (Connection connection, Object object),*  *output (Account)* |
| *06* | *countAccount* | *count account, input (Connection connection),*  *output (int)* |

### c. Sequence Diagram(s)



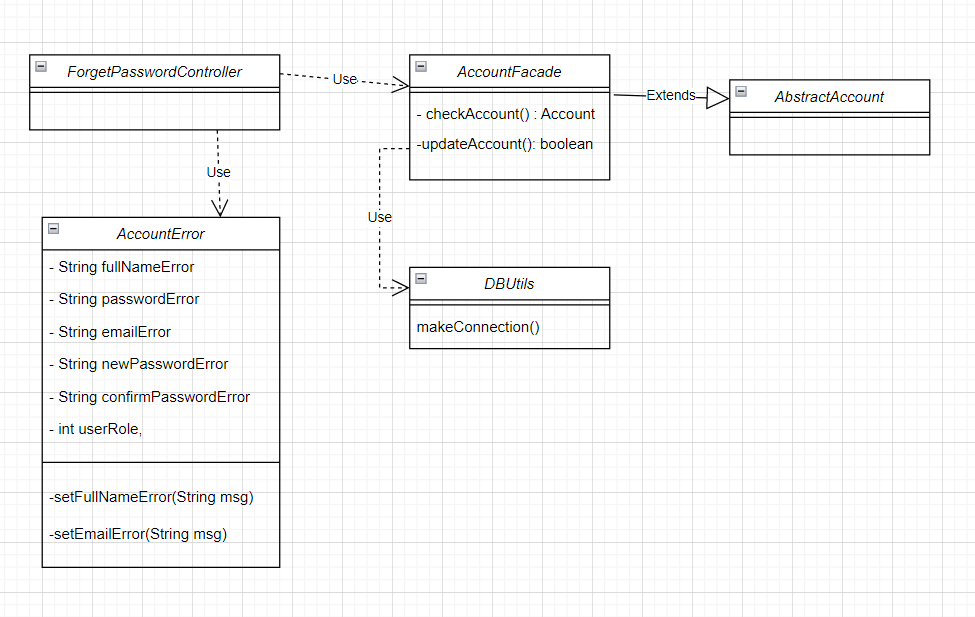
### d. Database queries

checkAccount(Email) : Account

+ "SELECT \* FROM Account WHERE UserEmail = ?"

## 2. Authentication / Forget Password

### a. Class Diagram

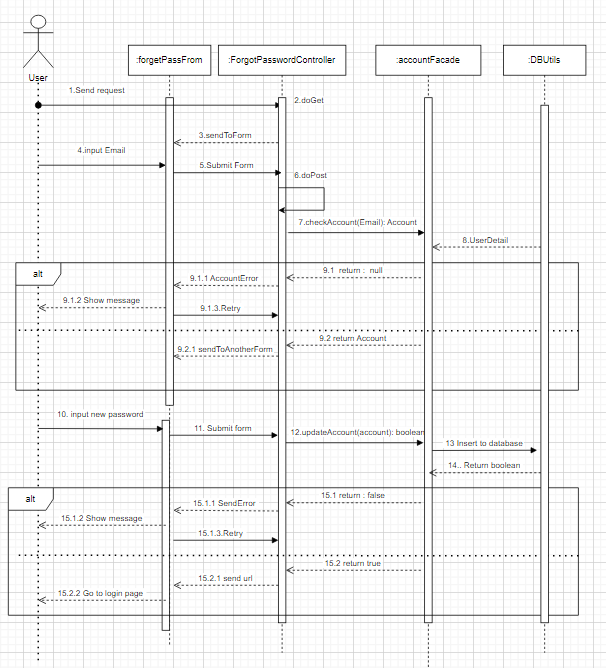


### b. Class Specifications

#### AccountFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoAccountFromSQL* | *get information of account, input ResultSet resultSet, output Account* |
| *02* | *registerAccount* | *create account, input (Connection connection, Account account), output String* |
| *03* | *updateAccount* | *update account, input (Connection connection, Account account, Object object),*  *output (true or false)* |
| *04* | *getAccount* | *get account, input (Connection connection, Object object, Object action, Object status),*  *output (List<Account>)* |
| *05* | *checkAccount* | *check account, input (Connection connection, Object object),*  *output (Account)* |
| *06* | *countAccount* | *count account, input (Connection connection),*  *output (int)* |

### c. Sequence Diagram(s)



### d. Database queries

checkAccount(Email) : Account

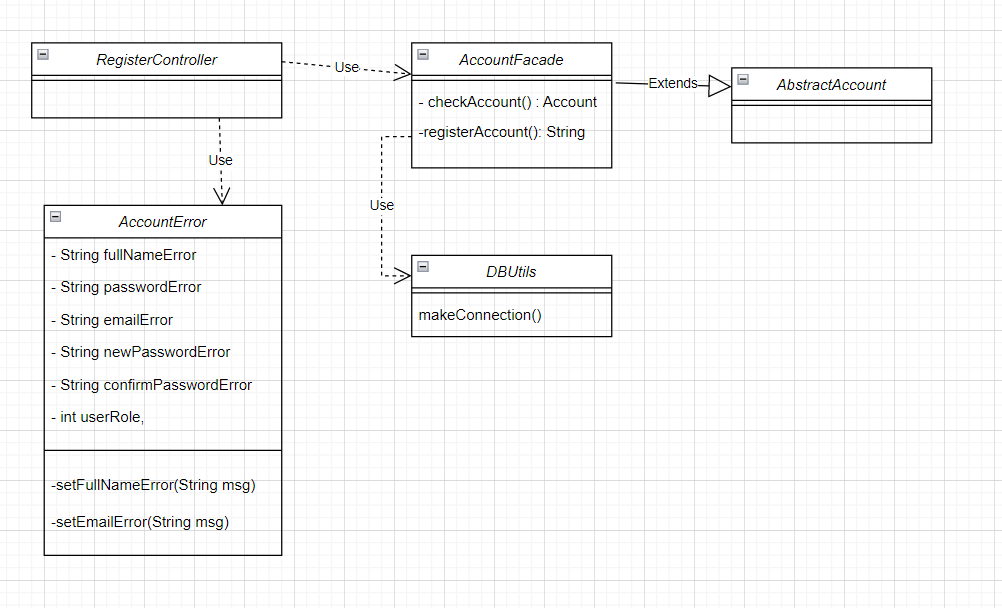
+ "SELECT \* FROM Account WHERE UserEmail = ?"

UpdateAccount(Account): Boolean

+ "UPDATE Account SET UserPassword = ? WHERE UserEmail = ?";

## 3. Authentication /Register

### a. Class Diagram

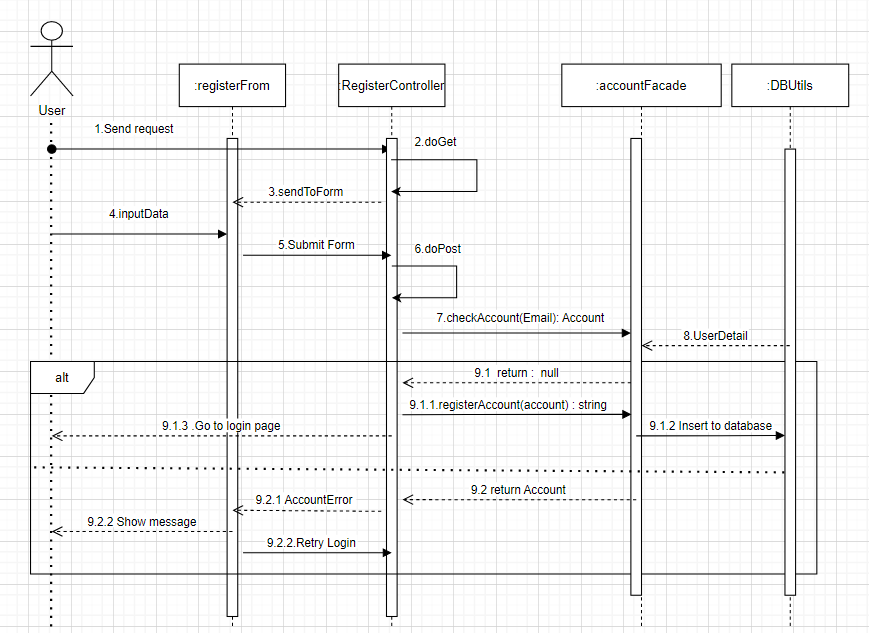


### b. Class Specifications

#### AccountFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoAccountFromSQL* | *get information of account, input ResultSet resultSet, output Account* |
| *02* | *registerAccount* | *create account, input (Connection connection, Account account), output String* |
| *03* | *updateAccount* | *update account, input (Connection connection, Account account, Object object),*  *output (true or false)* |
| *04* | *getAccount* | *get account, input (Connection connection, Object object, Object action, Object status),*  *output (List<Account>)* |
| *05* | *checkAccount* | *check account, input (Connection connection, Object object),*  *output (Account)* |
| *06* | *countAccount* | *count account, input (Connection connection),*  *output (int)* |

### c. Sequence Diagram(s)



### d. Database queries

checkAccount(Email) : Account

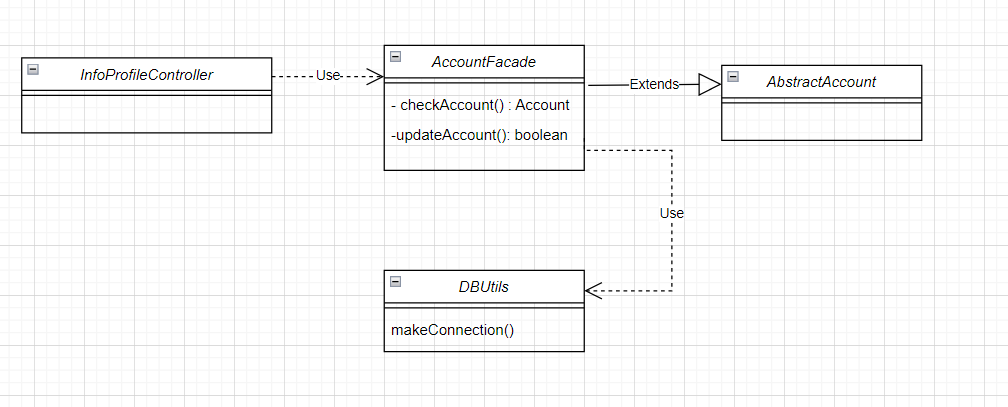
+ "SELECT \* FROM Account WHERE UserEmail = ?";

registerAccount(Account) : String

+"INSERT INTO Account(UserID, FullName, UserPassword, UserEmail, DefaultAvatar, ColorAvatar) VALUES(?, ?, ?, ?, ?, ?)";

## 4. User Information /View Profile

### a. Class Diagram

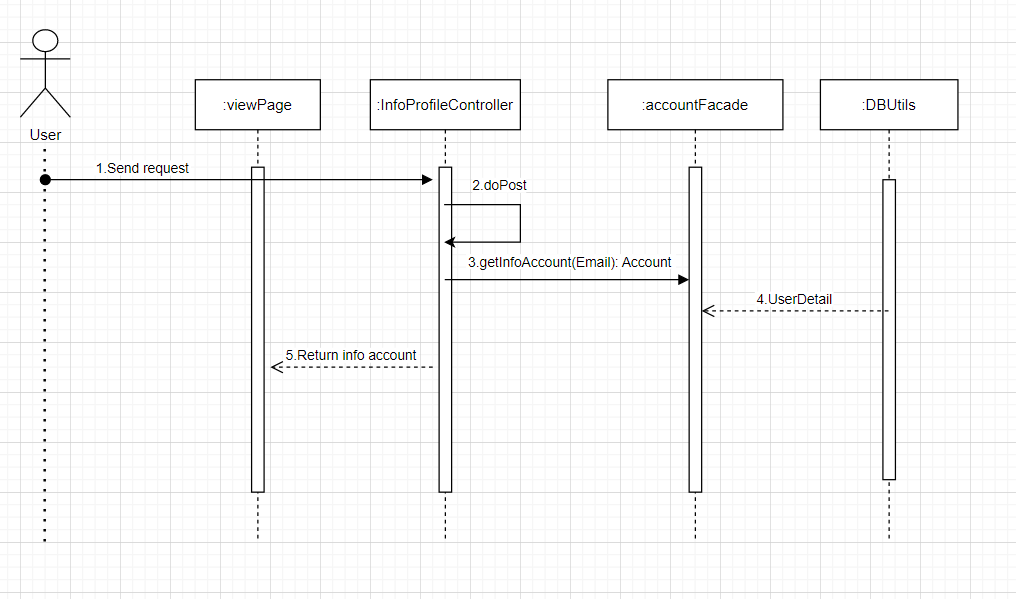


### b. Class Specifications

#### AccountFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoAccountFromSQL* | *get information of account, input ResultSet resultSet, output Account* |
| *02* | *registerAccount* | *create account, input (Connection connection, Account account), output String* |
| *03* | *updateAccount* | *update account, input (Connection connection, Account account, Object object),*  *output (true or false)* |
| *04* | *getAccount* | *get account, input (Connection connection, Object object, Object action, Object status),*  *output (List<Account>)* |
| *05* | *checkAccount* | *check account, input (Connection connection, Object object),*  *output (Account)* |
| *06* | *countAccount* | *count account, input (Connection connection),*  *output (int)* |

### c. Sequence Diagram(s)



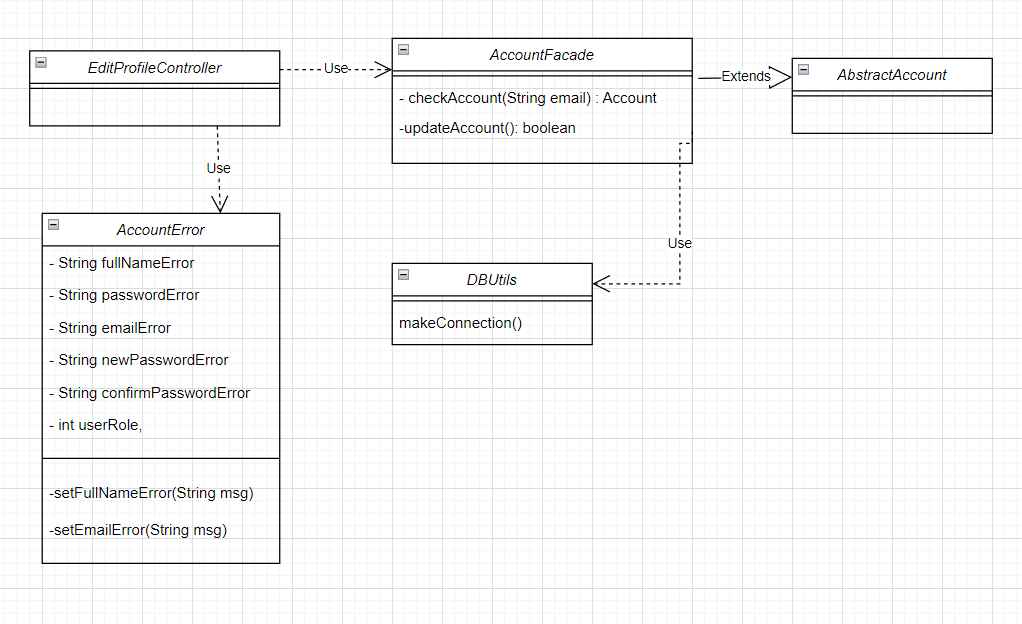
### d. Database queries

getInfoAccount(Email): Account

"SELECT \* FROM Account WHERE UserEmail = ?";

## 5. User Information /Edit View Profile

### a. Class Diagram



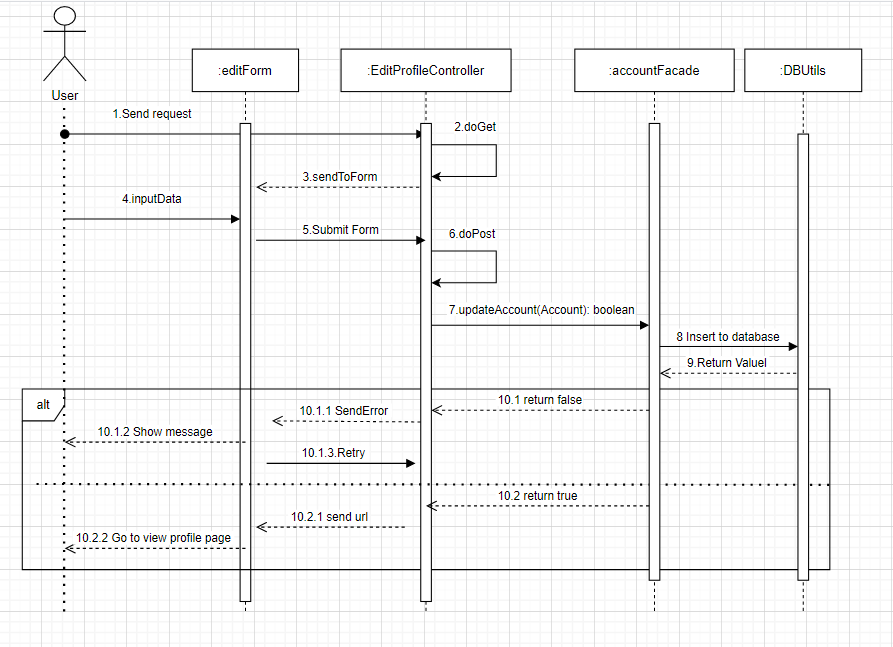
### b. Class Specifications

#### AccountFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoAccountFromSQL* | *get information of account, input ResultSet resultSet, output Account* |
| *02* | *registerAccount* | *create account, input (Connection connection, Account account), output String* |
| *03* | *updateAccount* | *update account, input (Connection connection, Account account, Object object),*  *output (true or false)* |
| *04* | *getAccount* | *get account, input (Connection connection, Object object, Object action, Object status),*  *output (List<Account>)* |
| *05* | *checkAccount* | *check account, input (Connection connection, Object object),*  *output (Account)* |
| *06* | *countAccount* | *count account, input (Connection connection),*  *output (int)* |

### 

### c. Sequence Diagram(s)



### 

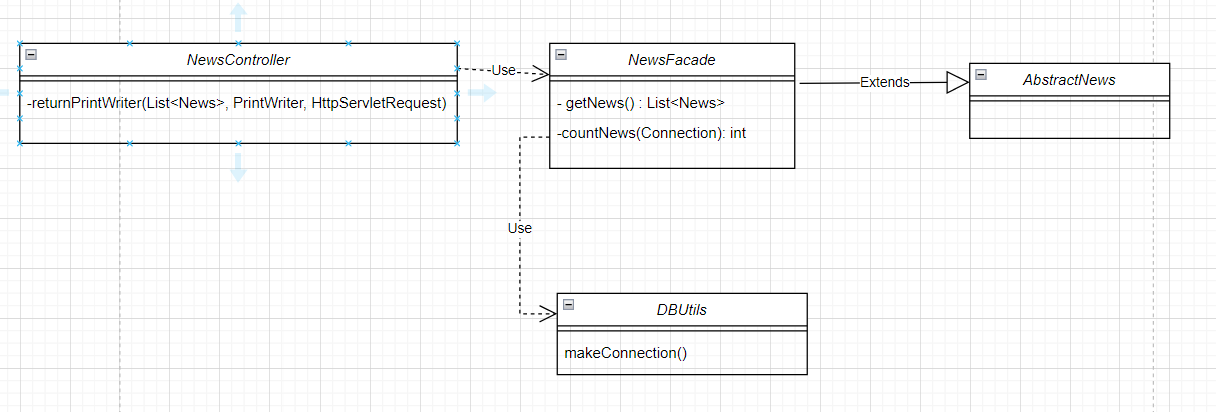
### d. Database queries

UpdateAccount(Account): Boolean

+ "UPDATE Account SET UserPassword = ? WHERE UserEmail = ?";

## 6. News Management / View News information

### a. Class Diagram

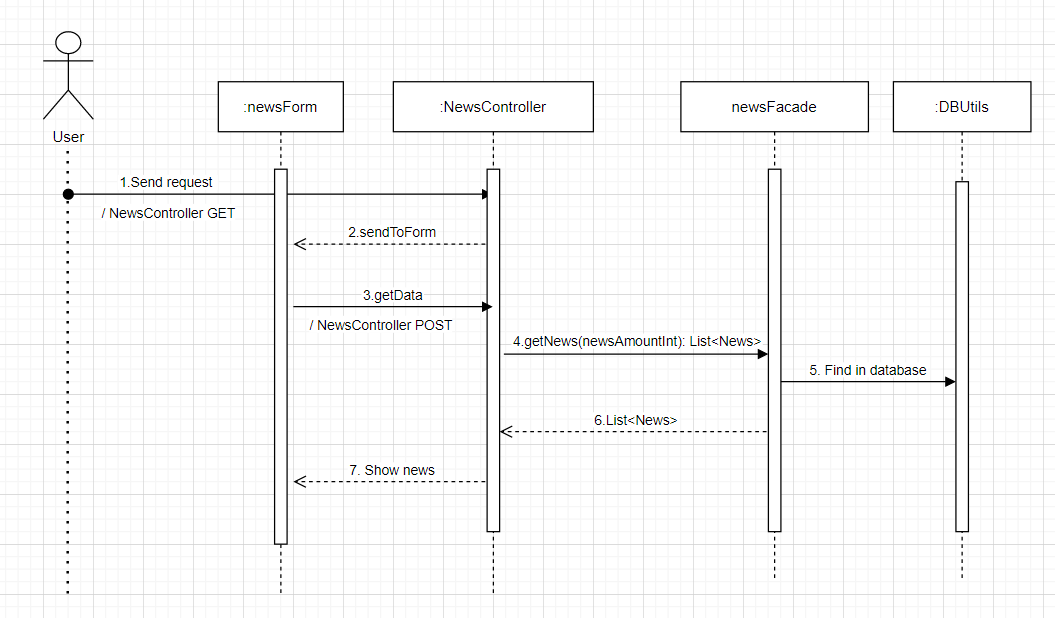


### b. Class Specifications

#### NewsDetailFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoNewsDetailFromSQL* | *get News Detail, input (ResultSet resultSet),*  *output (NewsDetail)* |
| *02* | *getNewsDetail* | *add new News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |
| *03* | *addNewsDetail* | *update News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |
| *04* | *updateNewsDetail* | *update News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |

### c. Sequence Diagram(s)



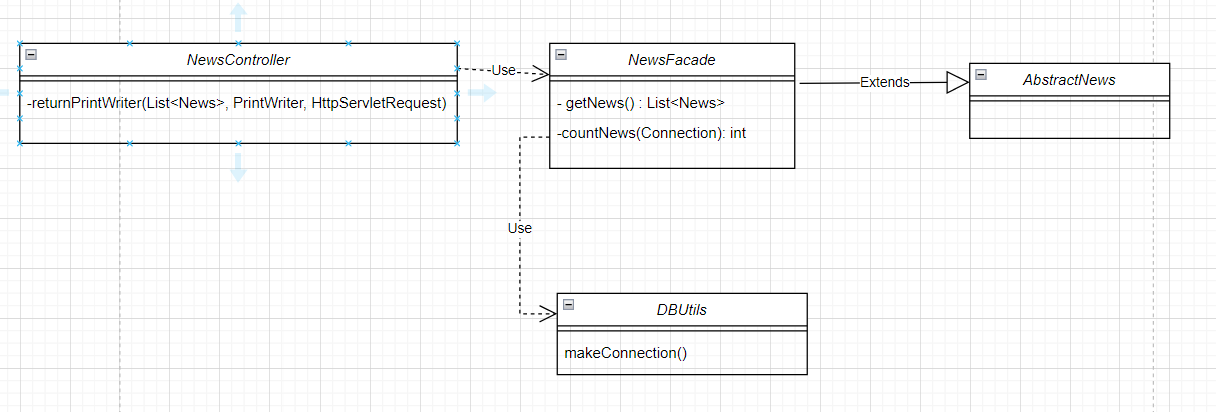
### d. Database queries

getNews( int AmountInt) : List<News>

+ "SELECT TOP 8 \* FROM News WHERE PostDate < GETDATE() ORDER BY PostDate DESC";

## 7. News Management / View News Detail

### a. Class Diagram



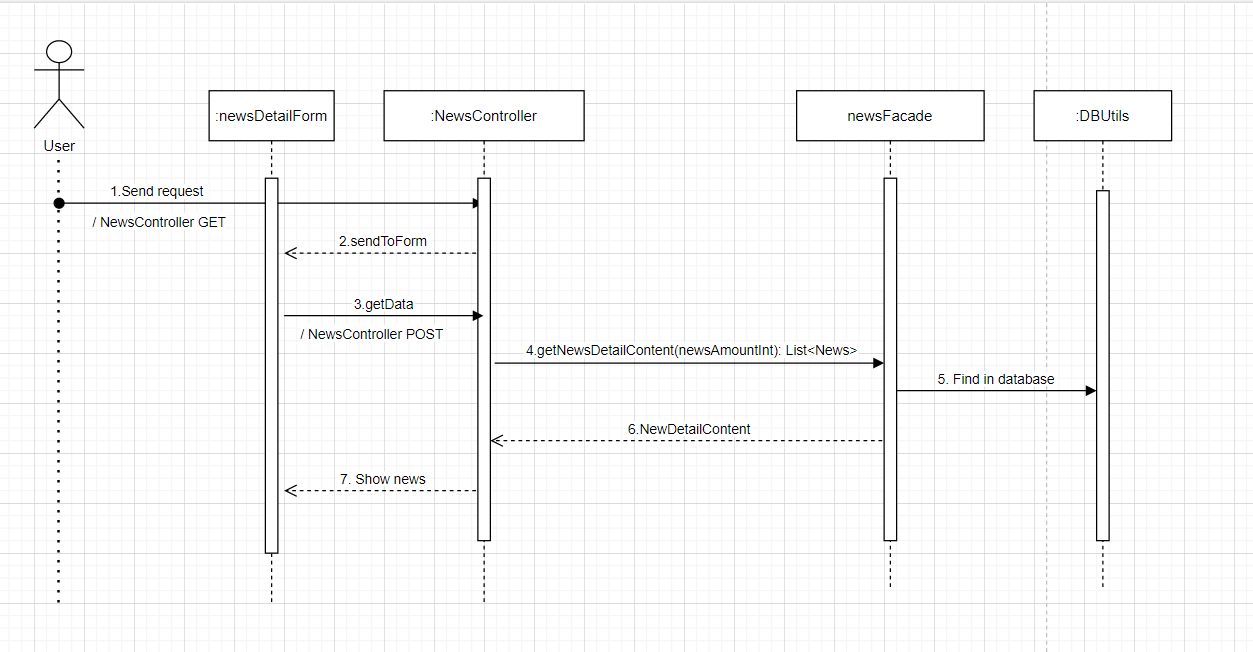
### b. Class Specifications

#### NewsDetailFacade

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoNewsDetailFromSQL* | *get News Detail, input (ResultSet resultSet),*  *output (NewsDetail)* |
| *02* | *getNewsDetail* | *add new News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |
| *03* | *addNewsDetail* | *update News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |
| *04* | *updateNewsDetail* | *update News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |

### c. Sequence Diagram(s)



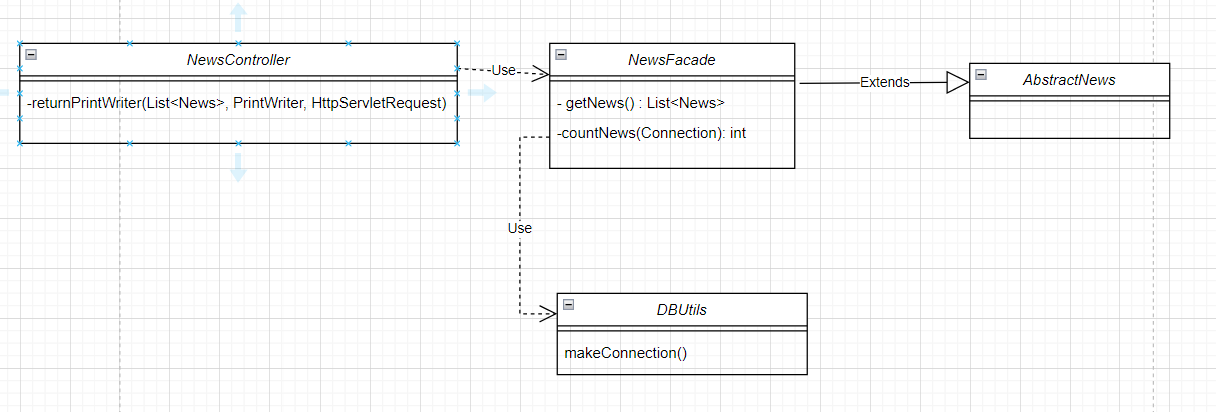
### d. Database queries

getNews( int AmountInt) : List<News>

+ "SELECT TOP 8 \* FROM News WHERE PostDate < GETDATE() ORDER BY PostDate DESC";

## 8. News Management / Search News

### a. Class Diagram

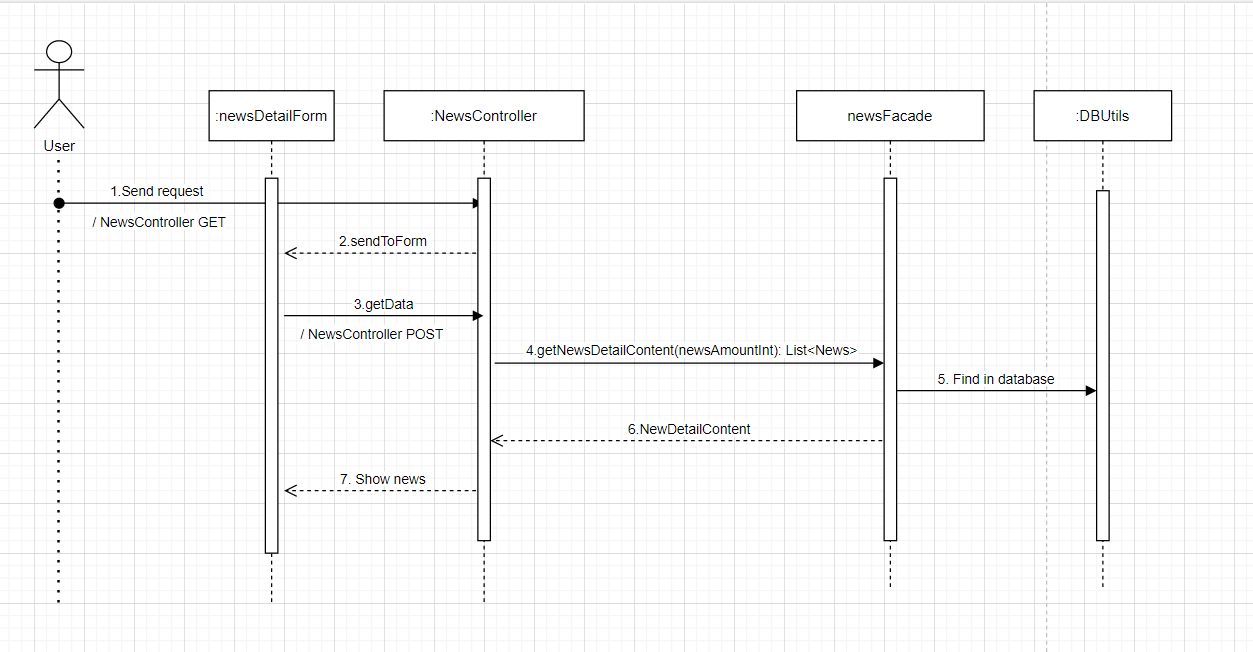


### b. Class Specifications

#### NewsDetailFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoNewsDetailFromSQL* | *get News Detail, input (ResultSet resultSet),*  *output (NewsDetail)* |
| *02* | *getNewsDetail* | *add new News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |
| *03* | *addNewsDetail* | *update News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |
| *04* | *updateNewsDetail* | *update News Detail, input (Connection connection, NewsDetail newsDetail), output (boolean)* |

### c. Sequence Diagram(s)



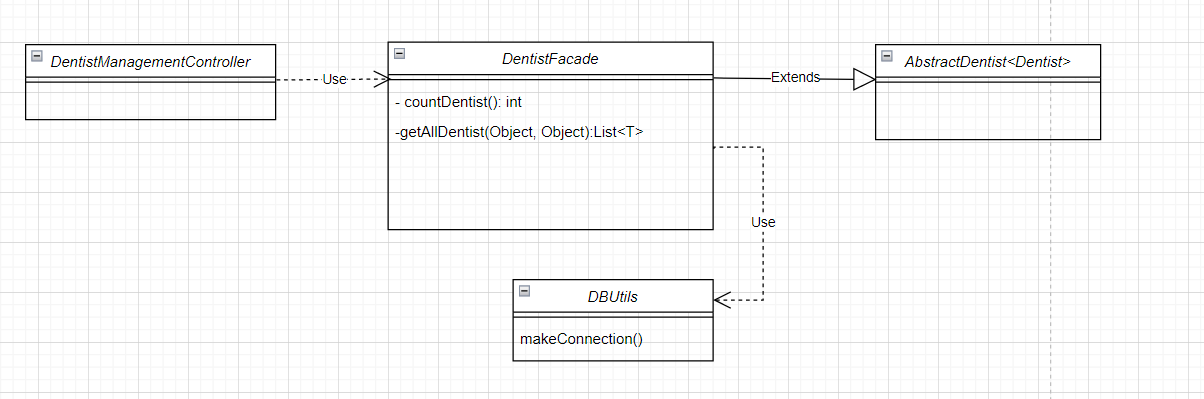
### d. Database queries

List<News> getNews :

+ "SELECT \* FROM News WHERE NameOfNews LIKE ?";

## 9. Dentist Information / View dentist information

### a. Class Diagram



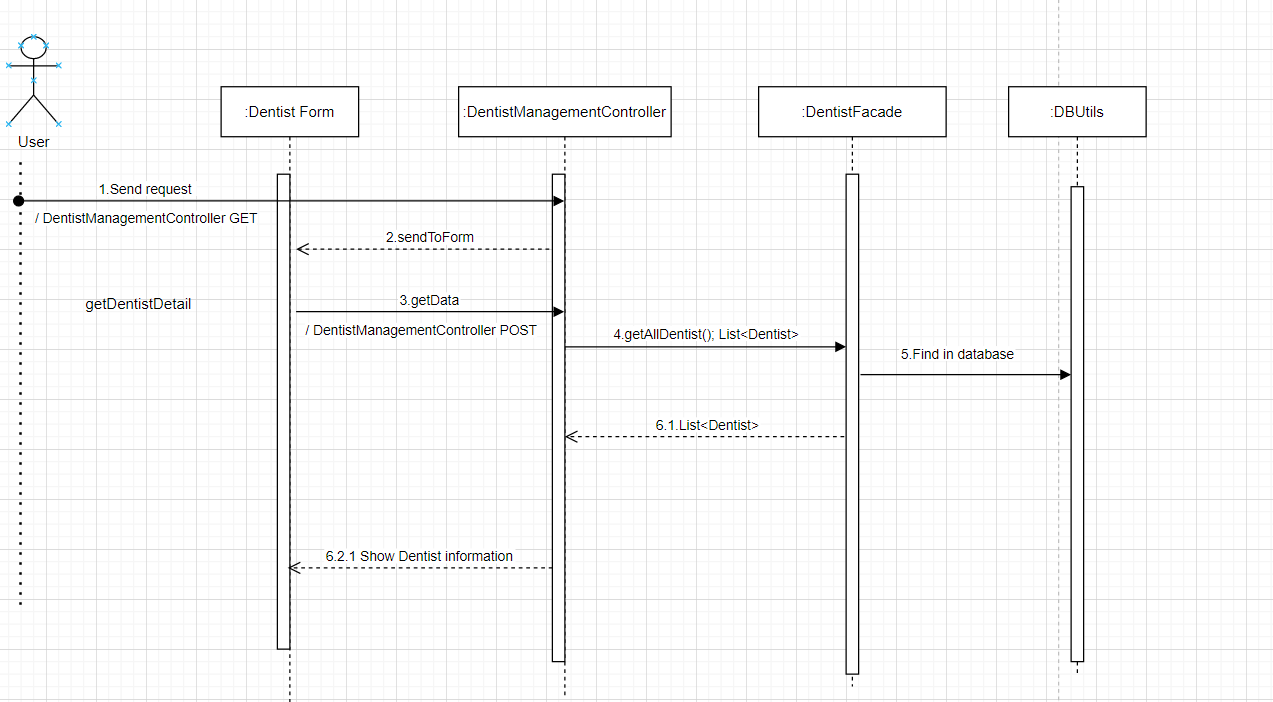
### b. Class Specifications

#### DentistFacade

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoDentistFromSQL* | *get dentist information, input (ResultSet resultSet),*  *output (Dentist)* |
| *02* | *getAllDentist* | *get all dentist information, input (Connection connection, Object object, Object action), output (List<Dentist>)* |
| *03* | *addDentist* | *add new dentist, input (Connection connection, Dentist dentist), output (boolean)* |
| *04* | *updateDentist* | *update dentist information, input (Connection connection, Dentist dentist), output (boolean)* |
| *05* | *countDentist* | *count dentist, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



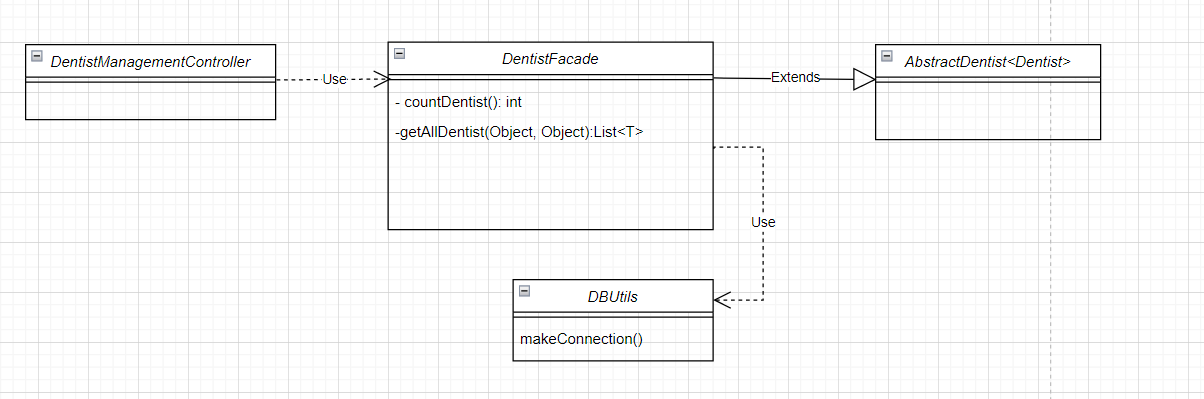
### d. Database queries

List<Dentist> getAllDentist:

+ "SELECT \* FROM Dentist";

## 10. Dentist Information / View dentist detail

### a. Class Diagram



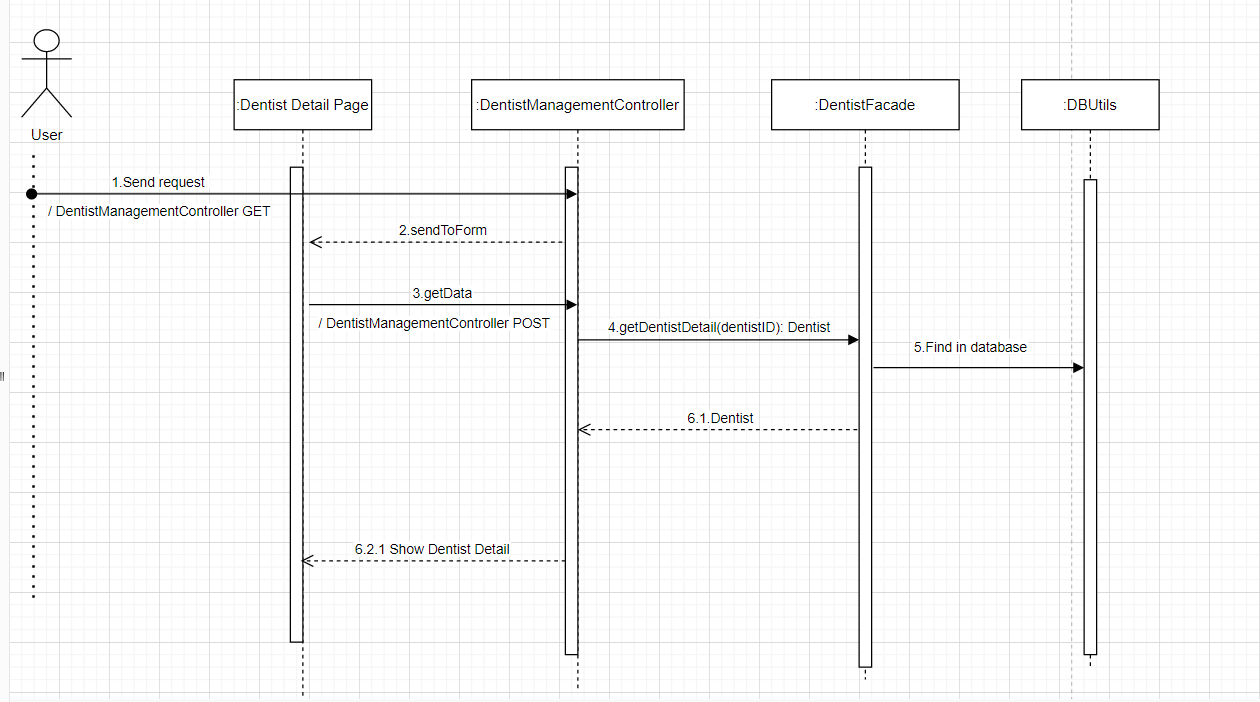
### b. Class Specifications

#### DentistFacade

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoDentistFromSQL* | *get dentist information, input (ResultSet resultSet),*  *output (Dentist)* |
| *02* | *getAllDentist* | *get all dentist information, input (Connection connection, Object object, Object action), output (List<Dentist>)* |
| *03* | *addDentist* | *add new dentist, input (Connection connection, Dentist dentist), output (boolean)* |
| *04* | *updateDentist* | *update dentist information, input (Connection connection, Dentist dentist), output (boolean)* |
| *05* | *countDentist* | *count dentist, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



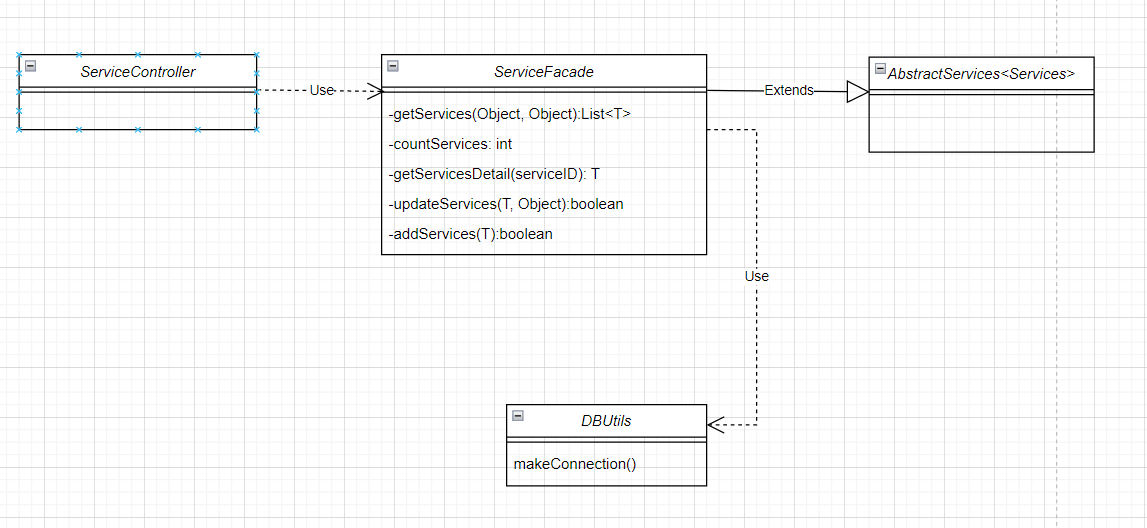
### d. Database queries

List<Dentist> getDentistDetail:

+"SELECT \* FROM Dentist WHERE DentistID = ?";

## 11. Service Management / View list Service (chưa có bảng)

### a. Class Diagram

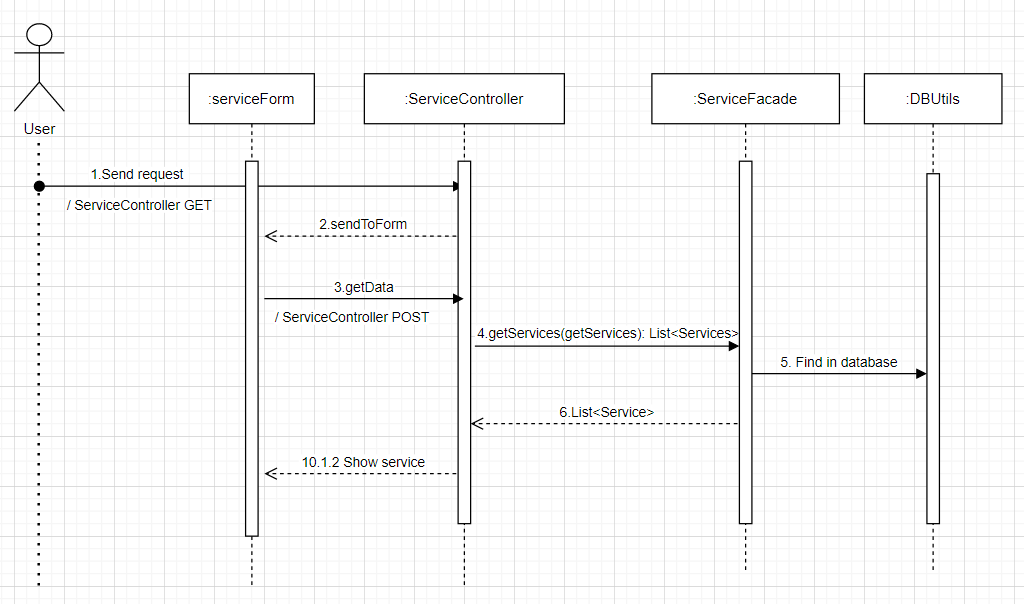


### b. Class Specifications

#### ServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |
| *06* | *getServicesDetail* | *get serviceDetail, input(Connection connection, Object serviceDetail), output(Services)* |

### c. Sequence Diagram(s)



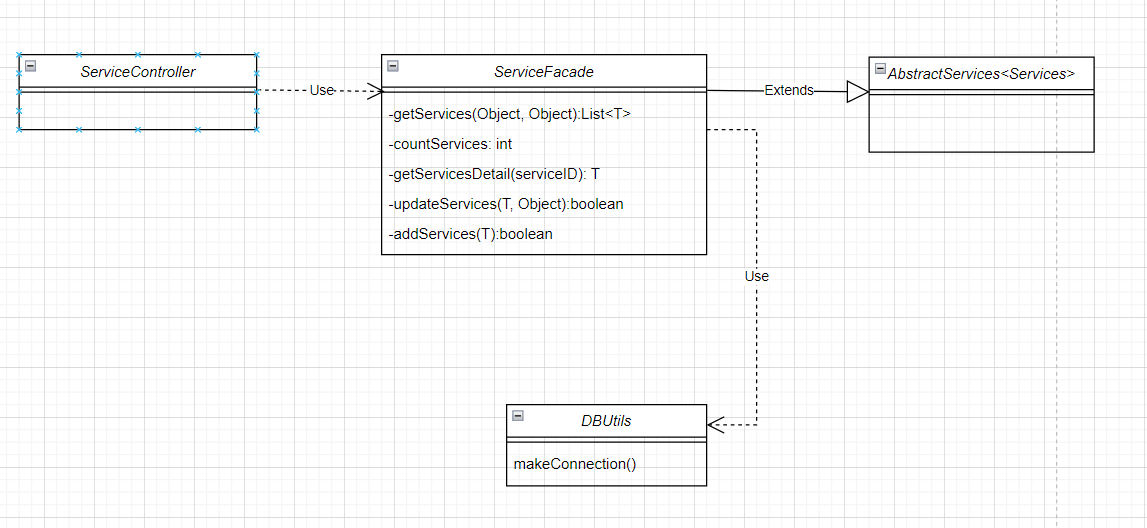
### d. Database queries

List<Services> getServices:

+"SELECT \* FROM [Services] WHERE ServiceStatus = ?";

## 12. Service Management / Search Service (chưa có bảng)

### a. Class Diagram

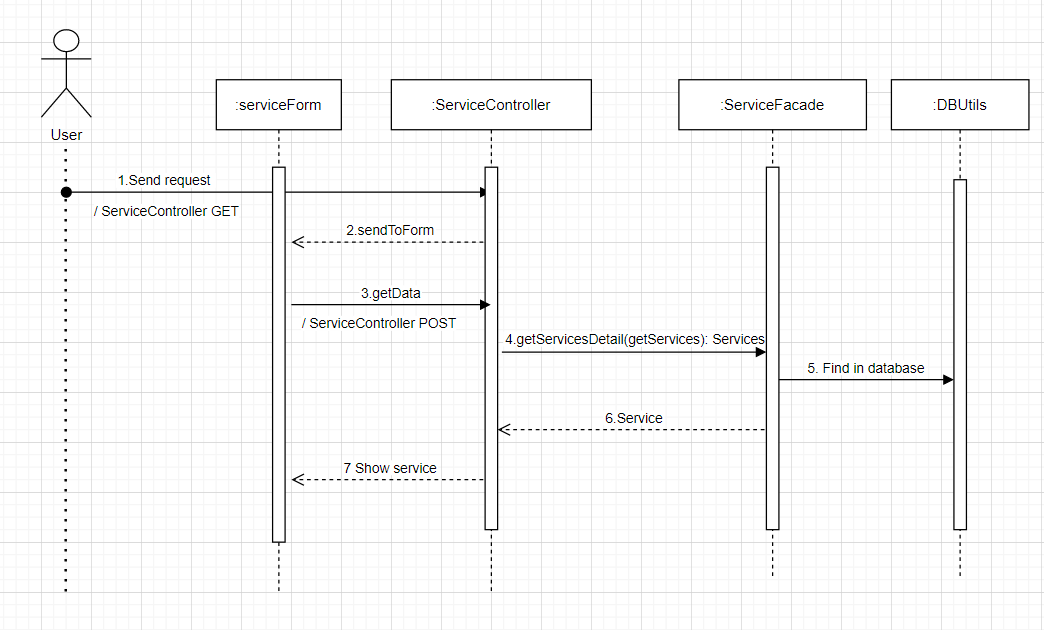


### b. Class Specifications

#### ServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |
| *06* | *getServicesDetail* | *get serviceDetail, input(Connection connection, Object serviceDetail), output(Services)* |

### c. Sequence Diagram(s)



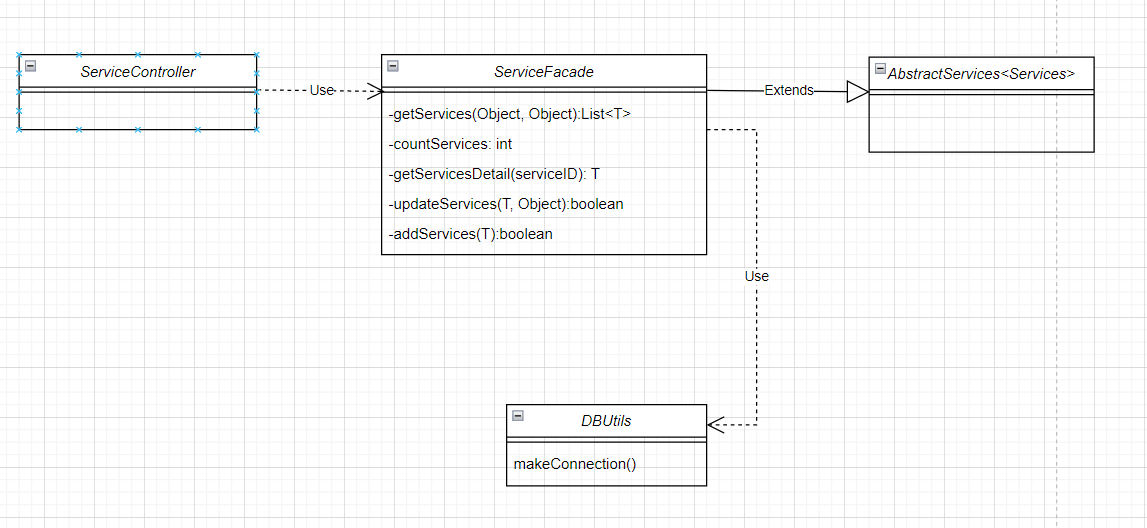
### d. Database queries

List<Services> getServices:

+"SELECT \* FROM [Services] WHERE ServiceID = ?";

## 13. Service Management / Delete Service (chưa có bảng)

### a. Class Diagram

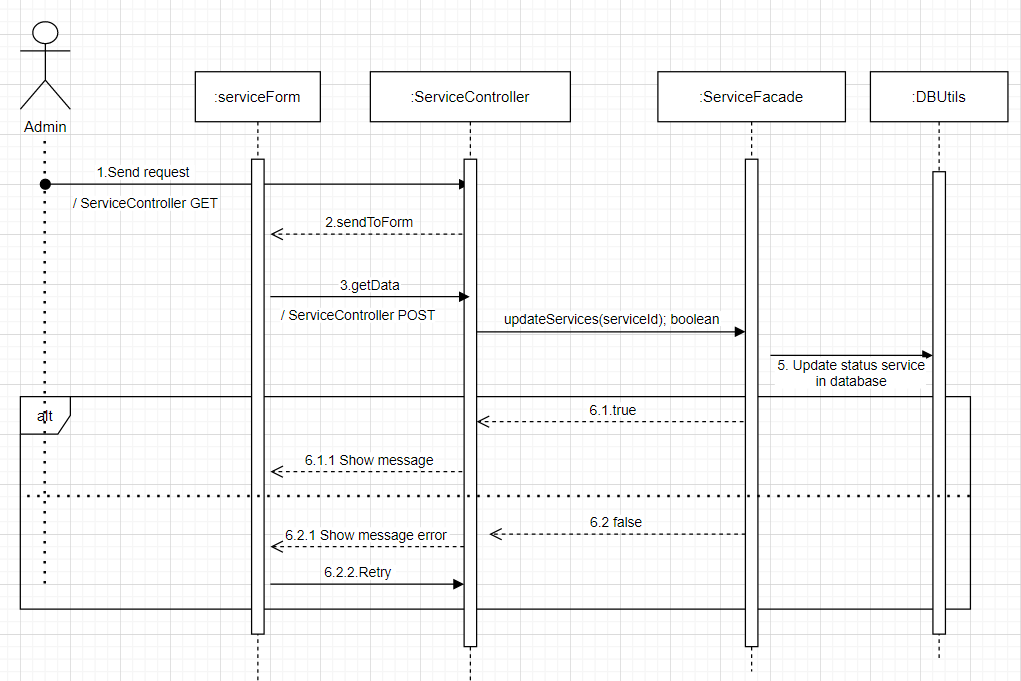


### b. Class Specifications

#### ServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |
| *06* | *getServicesDetail* | *get serviceDetail, input(Connection connection, Object serviceDetail), output(Services)* |

### c. Sequence Diagram(s)



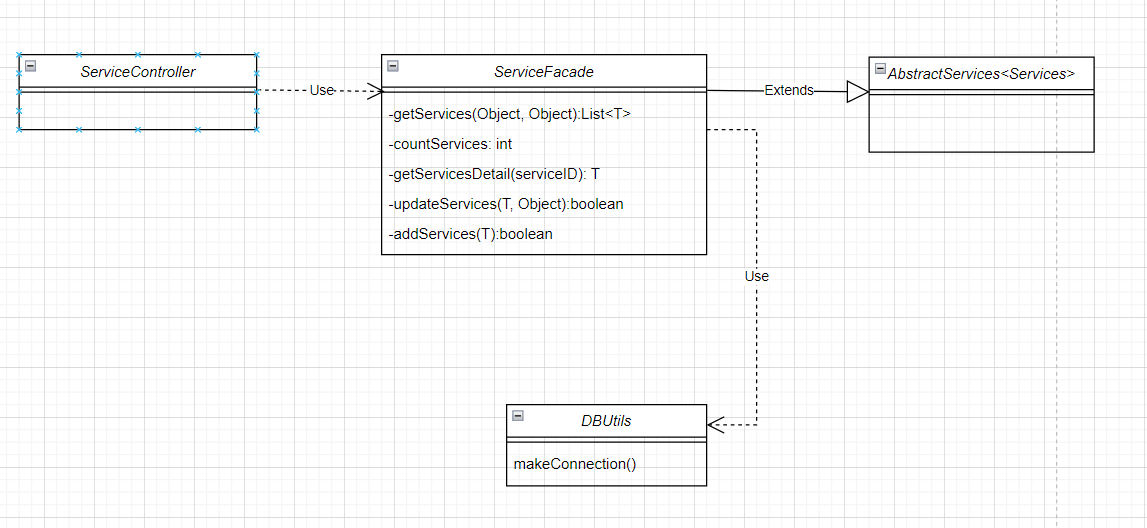
### d. Database queries

boolean updateServices(serviceId):

+"UPDATE [Services] SET ServiceStatus = ? WHERE ServiceID = ?";

## 14. Service Management / Add new Service (chưa có bảng)

### a. Class Diagram

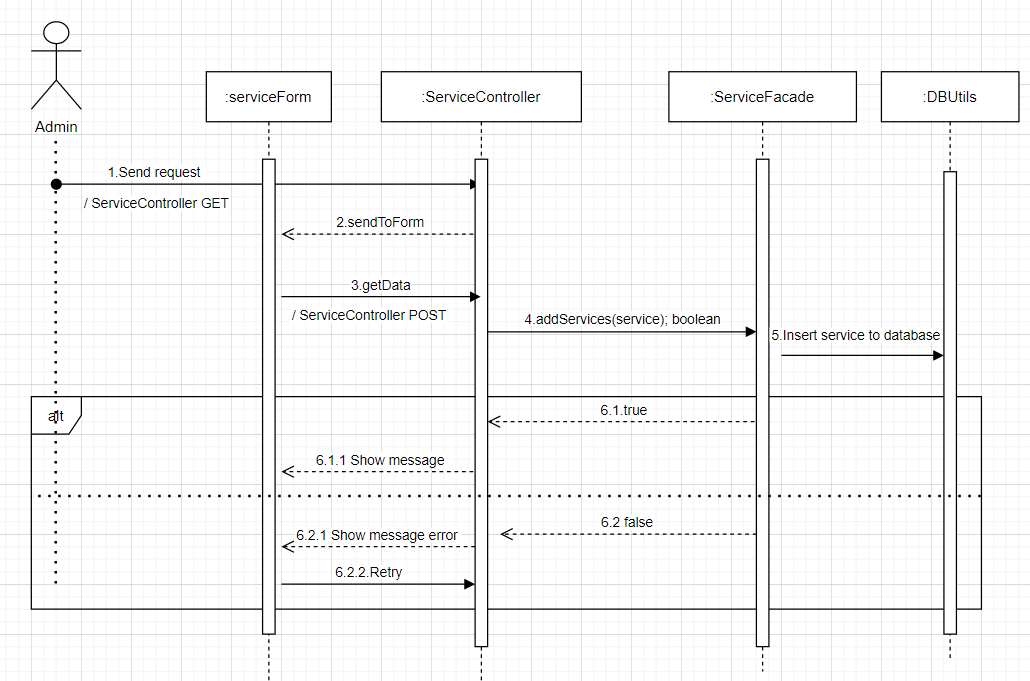


### b. Class Specifications

#### ServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |
| *06* | *getServicesDetail* | *get serviceDetail, input(Connection connection, Object serviceDetail), output(Services)* |

### c. Sequence Diagram(s)



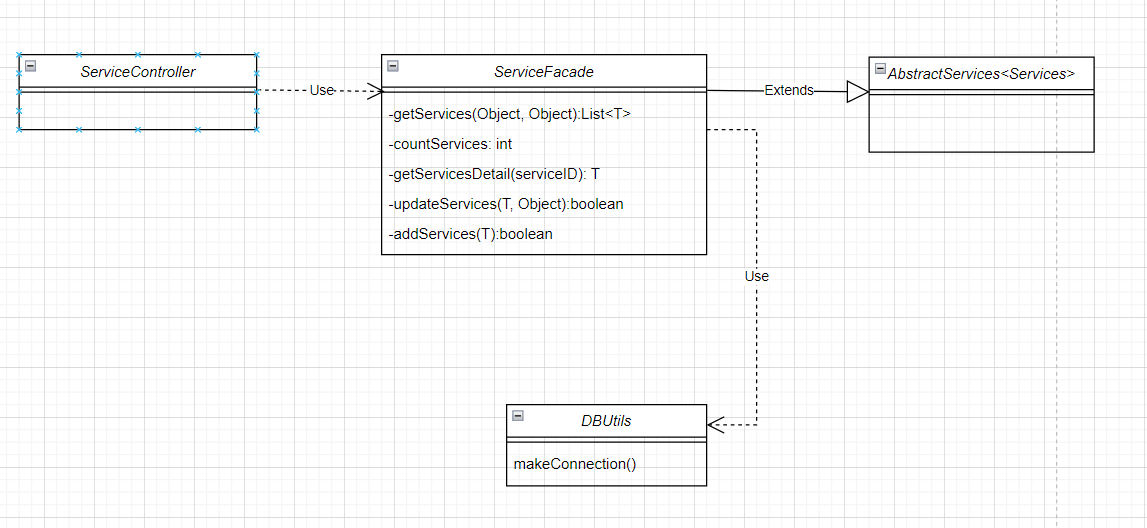
### d. Database queries

boolean addServices (service):

+"INSERT INTO [Services](ServiceID, ServiceName, ServicePrice, ImageService, DescriptionService, ServiceStatus) VALUES(?, ?, ?, ?, ?, ?)";

## 15. Service Management / Edit Service (chưa có bảng)

### a. Class Diagram

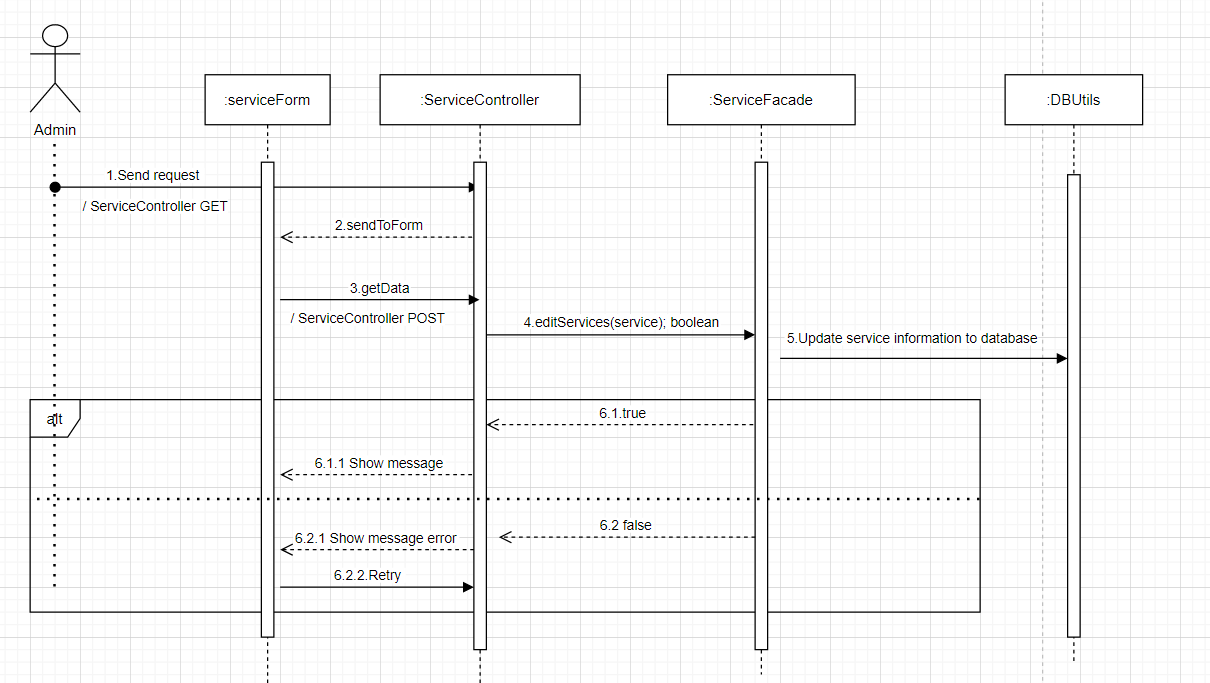


### b. Class Specifications

#### ServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |
| *06* | *getServicesDetail* | *get serviceDetail, input(Connection connection, Object serviceDetail), output(Services)* |

### c. Sequence Diagram(s)



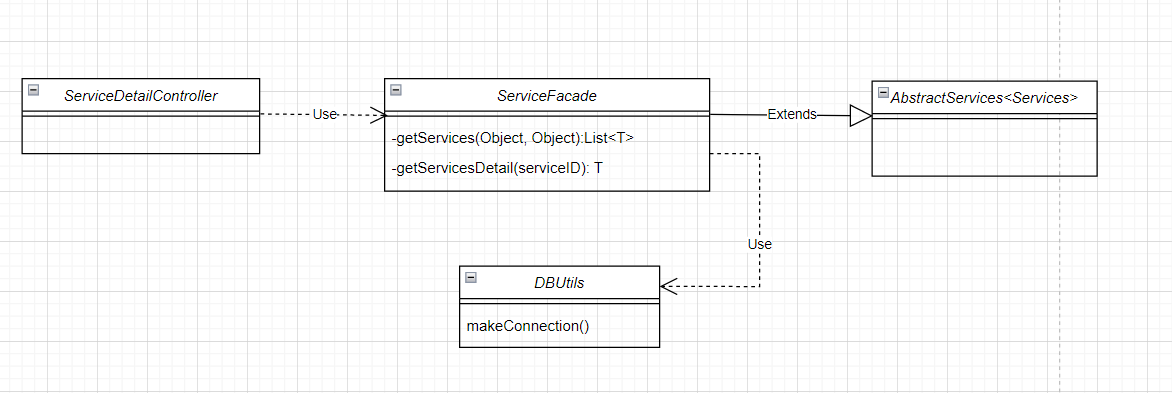
### d. Database queries

Boolean updateServices (service):

+"UPDATE [Services] SET ServiceName = ?, ServicePrice = ?, ImageService = ?, DescriptionService = ? WHERE ServiceID = ?";

## 16. Service Management / View Service Detail (chưa có bảng)

### a. Class Diagram

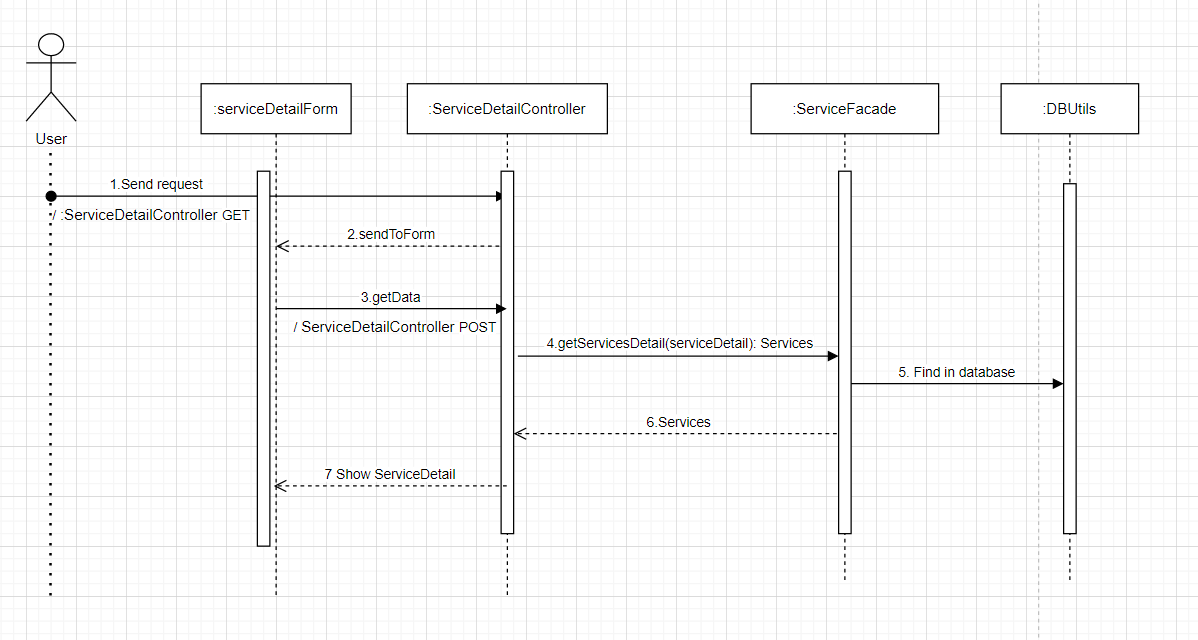


### b. Class Specifications

#### ServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |
| *06* | *getServicesDetail* | *get serviceDetail, input(Connection connection, Object serviceDetail), output(Services)* |

### c. Sequence Diagram(s)



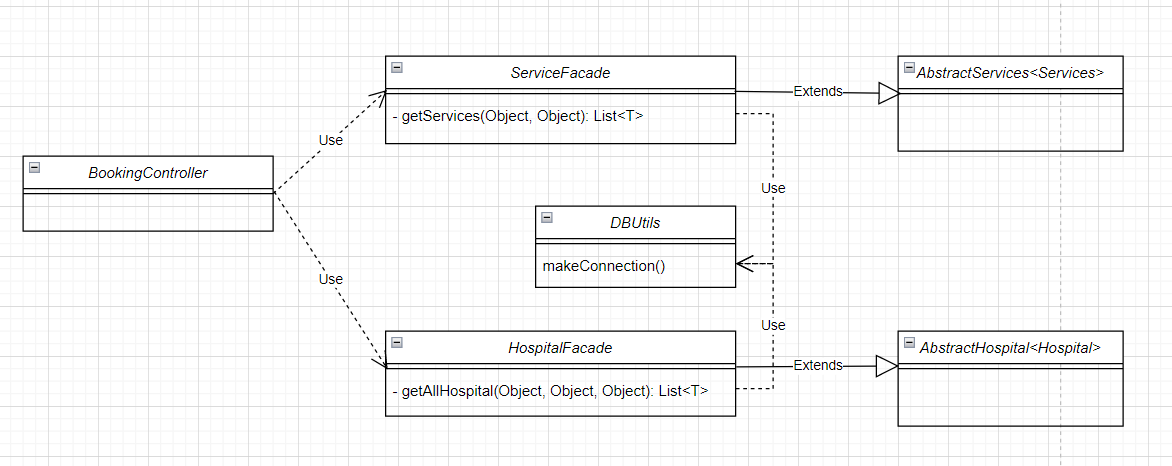
### d. Database queries

Services getServicesDetail (serviceId):

+"SELECT \* FROM [Services] WHERE ServiceID = ?";

## 17. Booking Management / View Booking schedule (chưa có bảng)

### a. Class Diagram

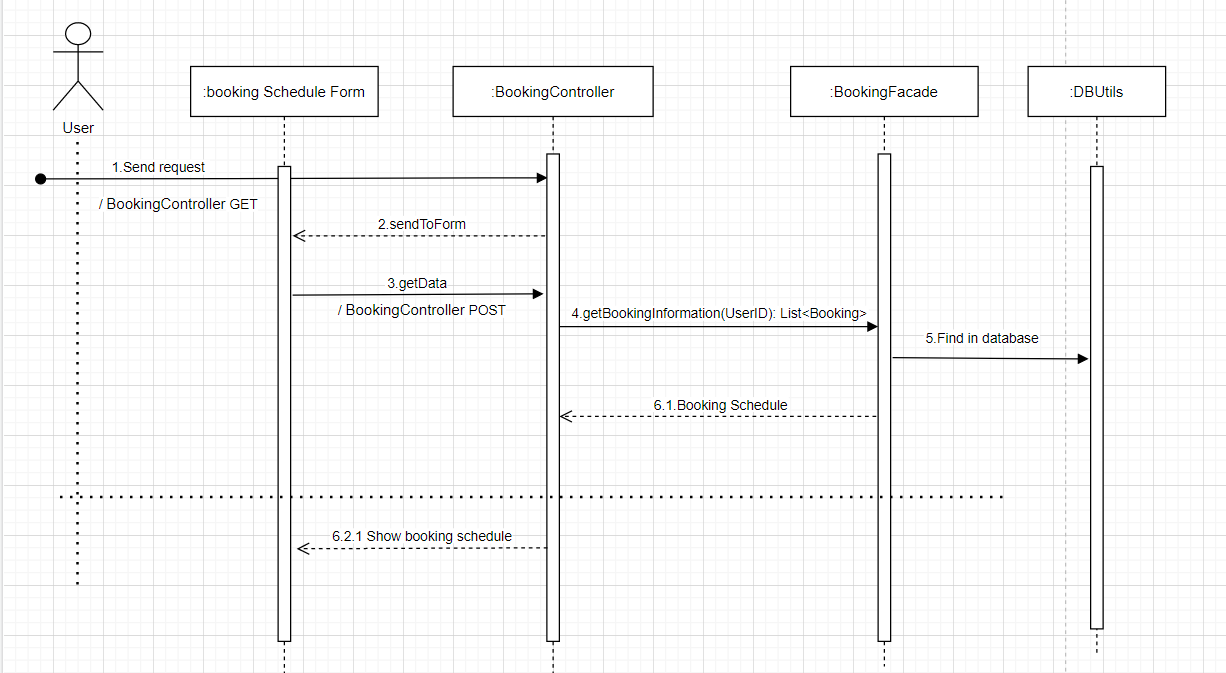


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



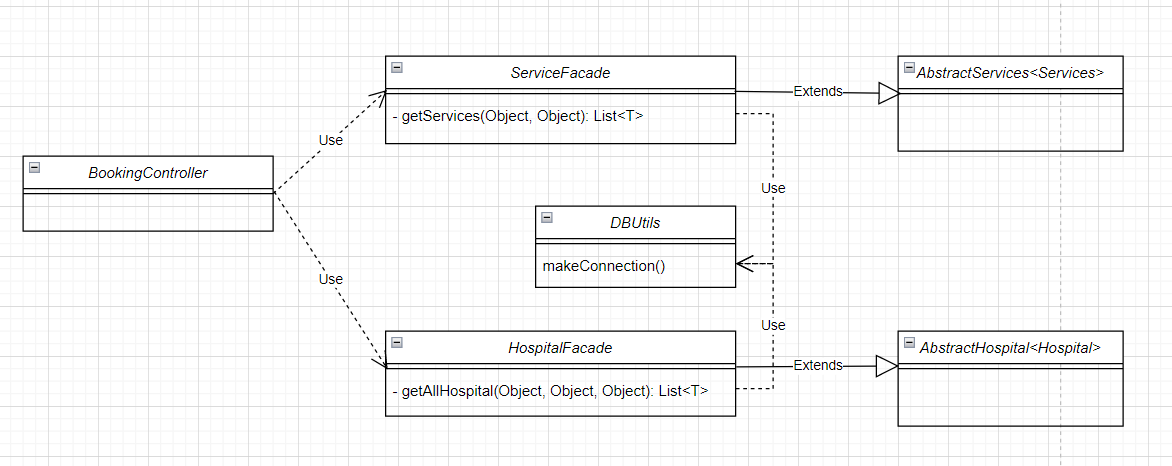
### d. Database queries

List<BookingService> getAllBookingService(BookingService):

+"SELECT \* FROM BookingService WHERE ServiceID = ? AND BookingDate = ?";

## 18. Booking Management / Add new Booking (chưa có bảng)

### a. Class Diagram

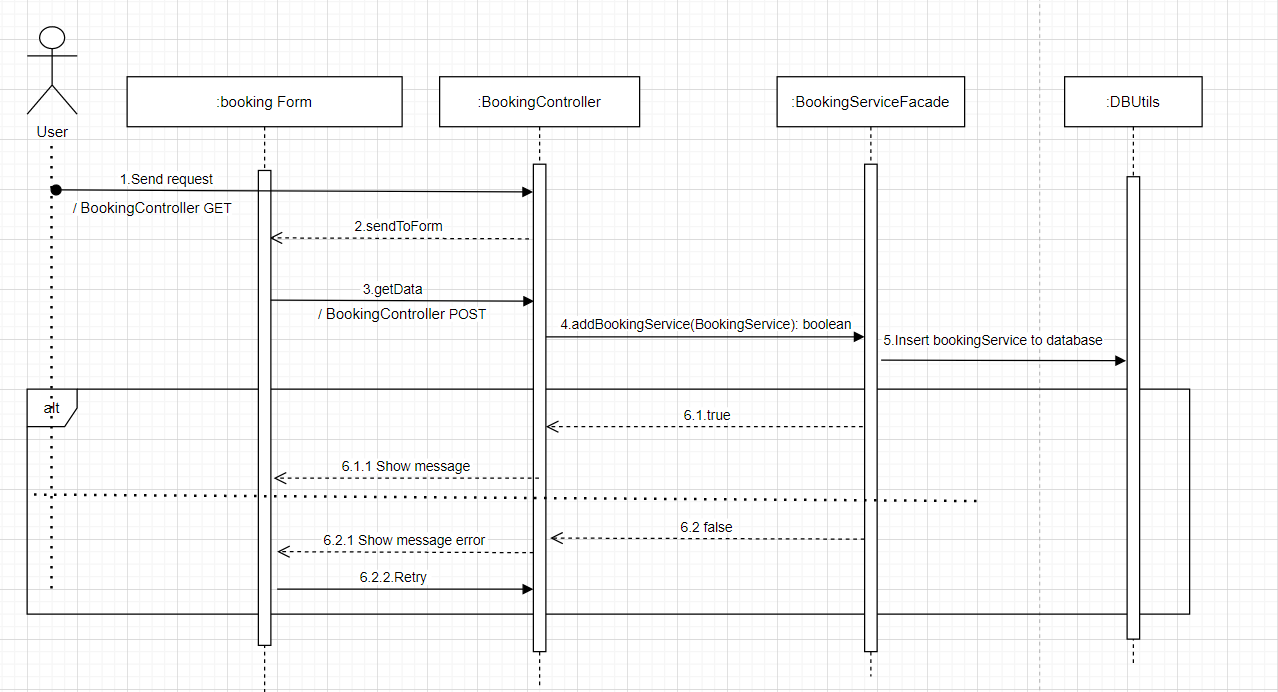


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



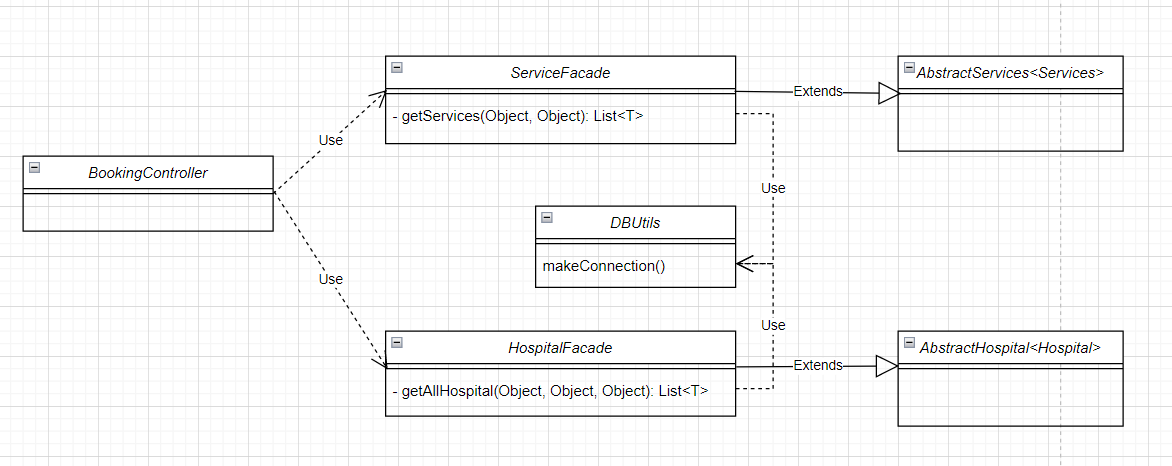
### d. Database queries

boolean addBookingService( BookingService)

+"INSERT INTO BookingService(BookingServiceID, ServiceID, BookingID, SlotServiceID, BookingDate) VALUES(?, ?, ?, ?, ?)";

## 19. Booking Management / Add service to the Booking (chưa có bảng)

### a. Class Diagram

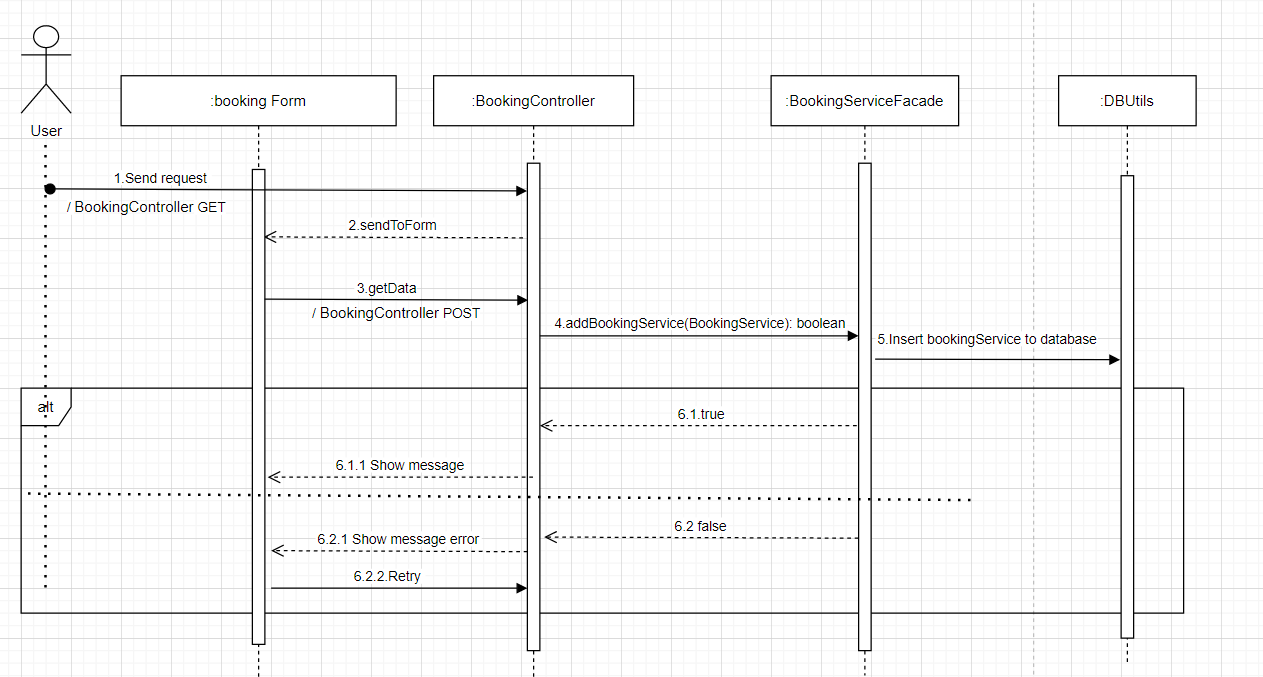


### b. Class Specifications

#### BookingServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingServiceFromSQL* | *get booking service information, input (ResultSet resultSet), output (BookingService)* |
| *02* | *getAllBookingService* | *get all booking service, input (Connection connection), output (List<BookingService>)* |
| *03* | *addBookingService* | *add new Booking, input (Connection connection,  BookingService bookingService), output (boolean)* |
| *04* | *updateBookingService* | *update updateBookingService, input (Connection connection, Object bookingServiceID), output (boolean)* |
| *05* | *countBookingService* | *count number of booking service, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



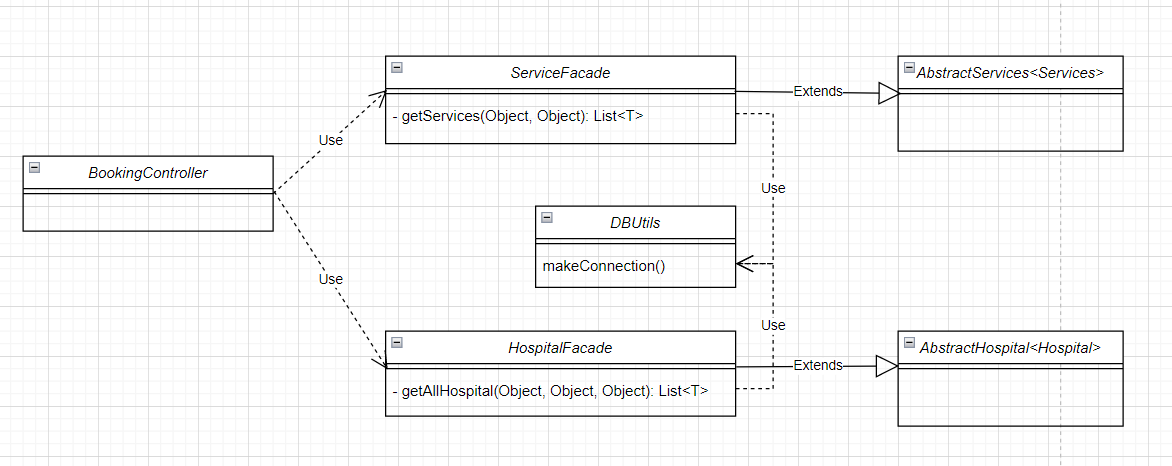
### d. Database queries

boolean addBookingService( BookingService)

+"INSERT INTO BookingService(BookingServiceID, ServiceID, BookingID, SlotServiceID, BookingDate) VALUES(?, ?, ?, ?, ?)";

## 20. Booking Management / Edit Booking (chưa có bảng)

### a. Class Diagram

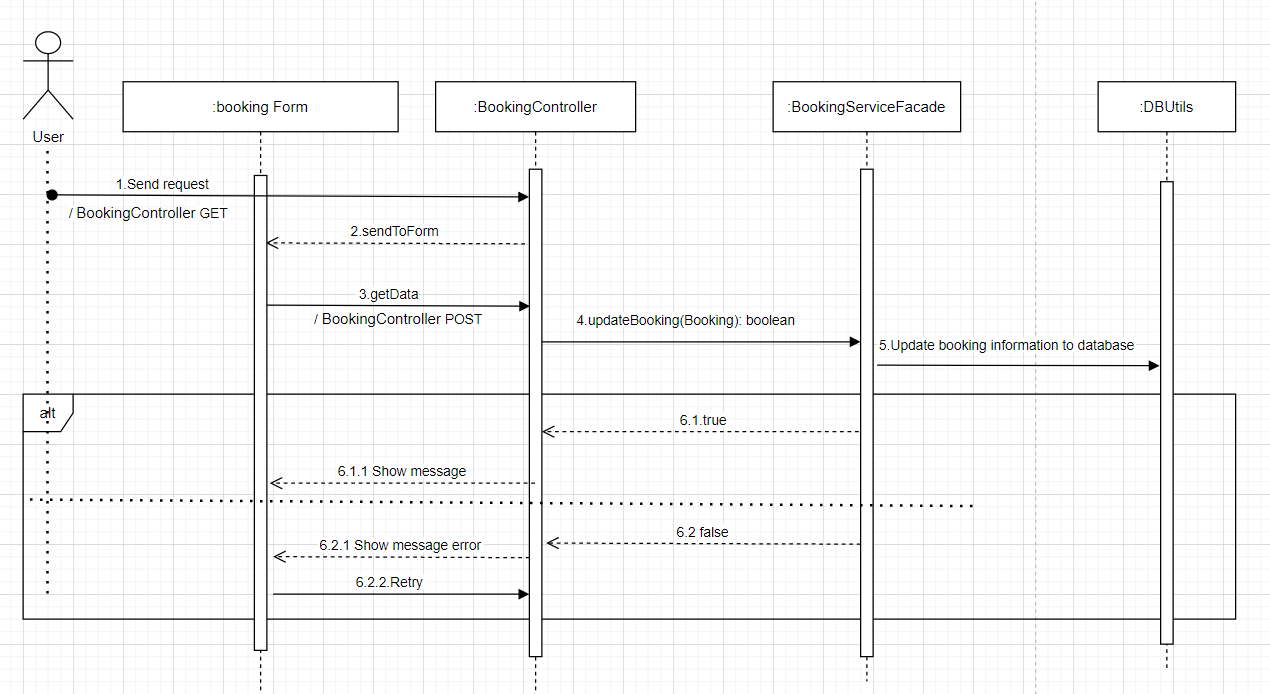


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



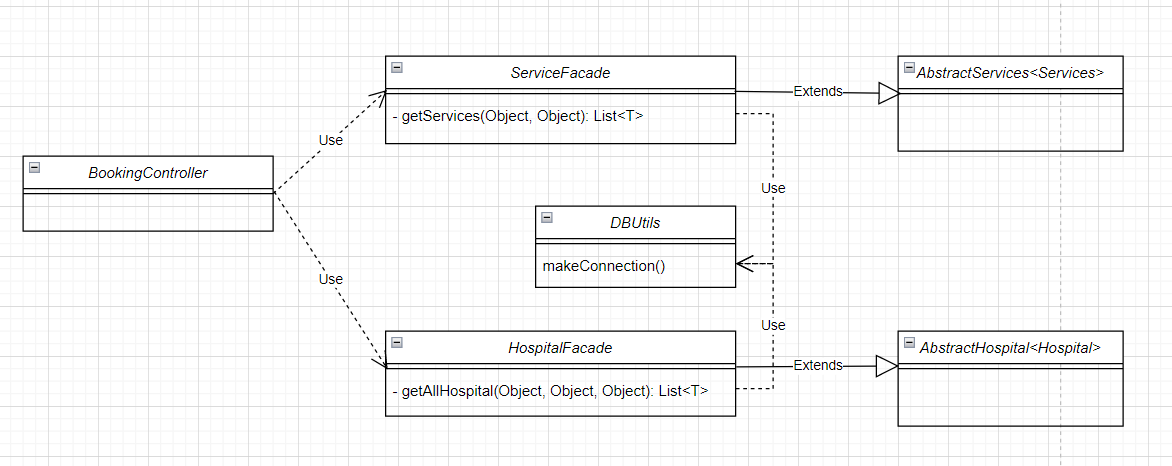
### d. Database queries

boolean updateBooking(Booking)

+"UPDATE Booking SET BookingStatus = ? WHERE BookingID = ?";

## 21. Booking Management / View History Booking Info (chưa có bảng)

### a. Class Diagram

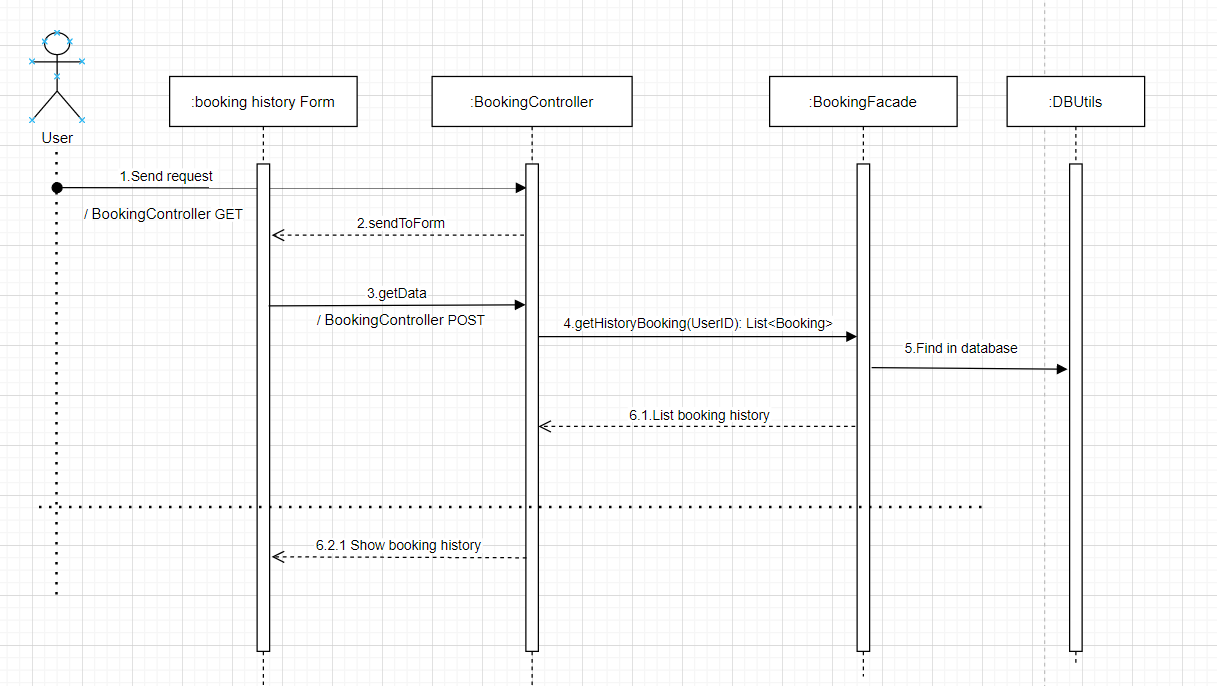


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



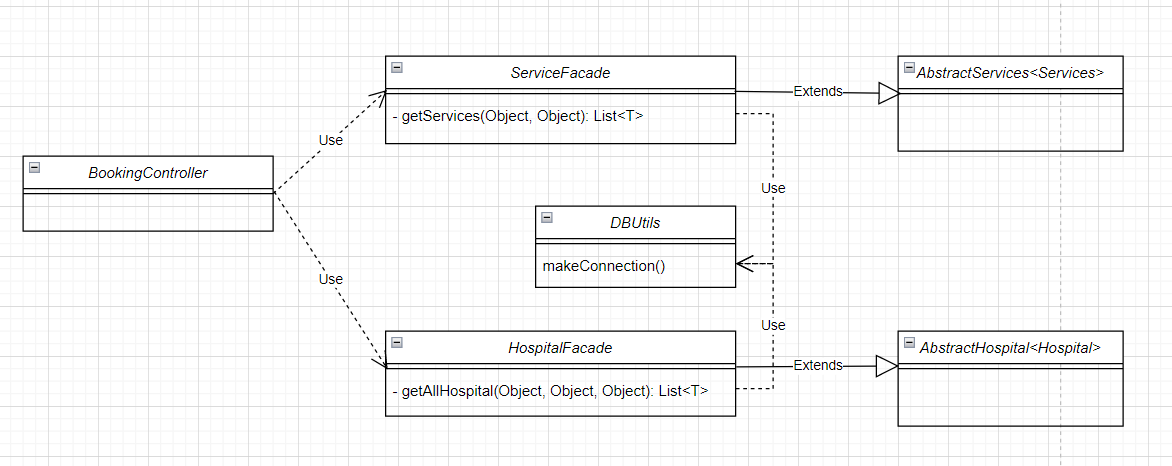
### d. Database queries

getHistoryBooking(UserID): List<Booking>:

## +"SELECT TOP 5 \* FROM Booking WHERE UserID = ?";

## 22. Booking Management / Search History Booking

### a. Class Diagram

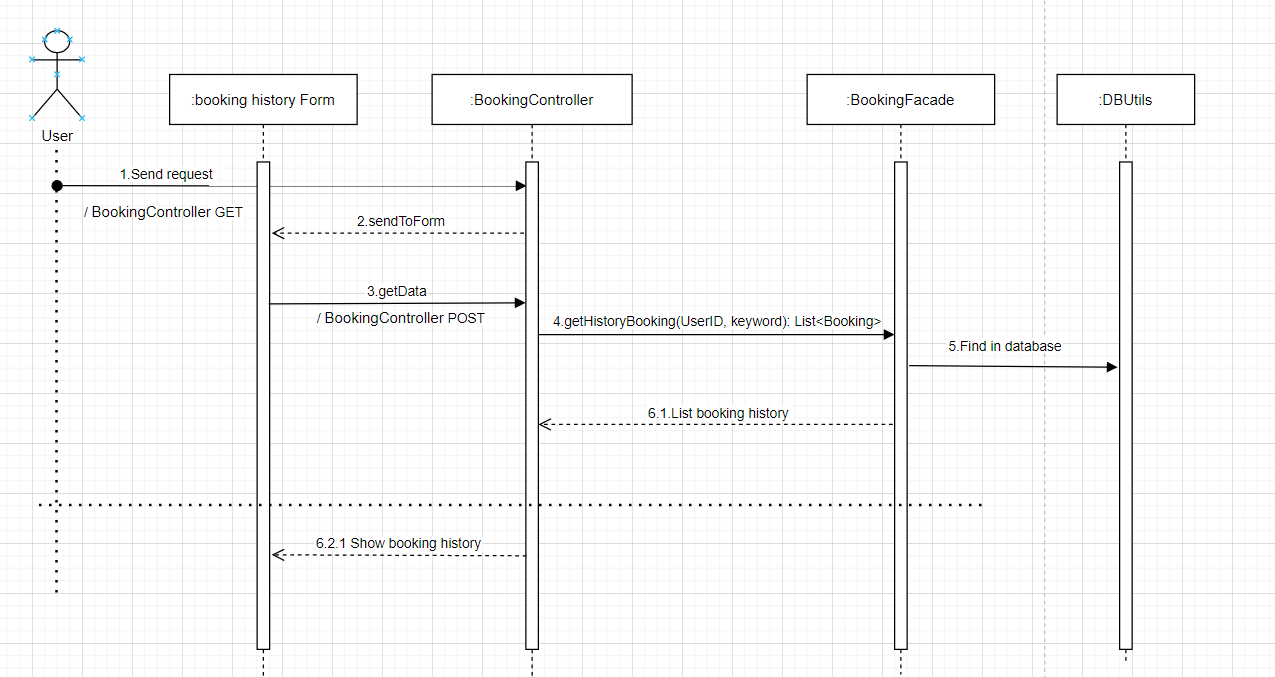


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



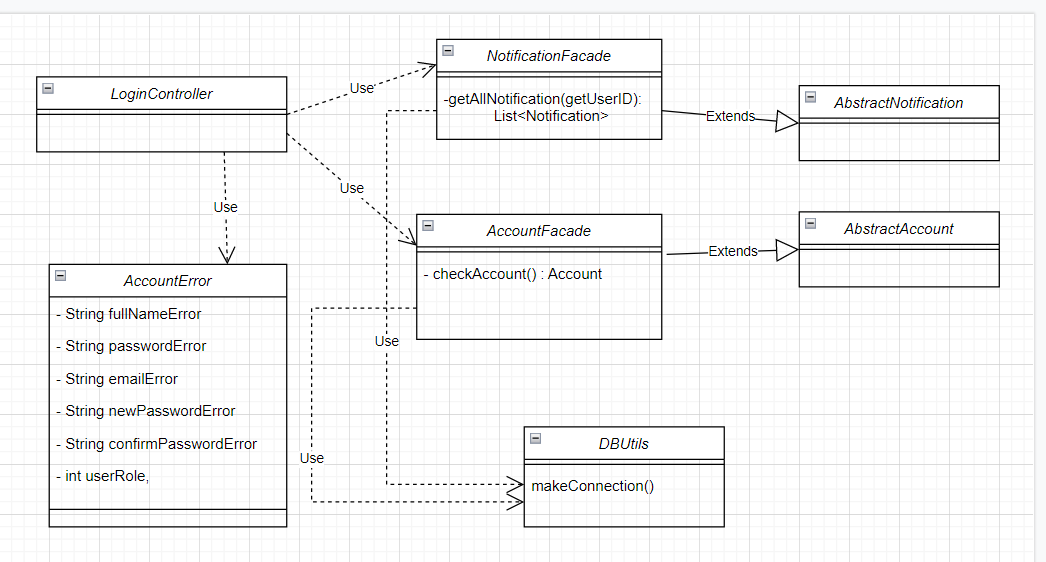
### d. Database queries

getHistoryBooking(UserID,keyword): List<Booking>:

## +"SELECT TOP 5 \* FROM Booking WHERE UserID = ? and NameOfService like ? ";

## 23. Announcement / View Notification (chưa có bảng)

### a. Class Diagram

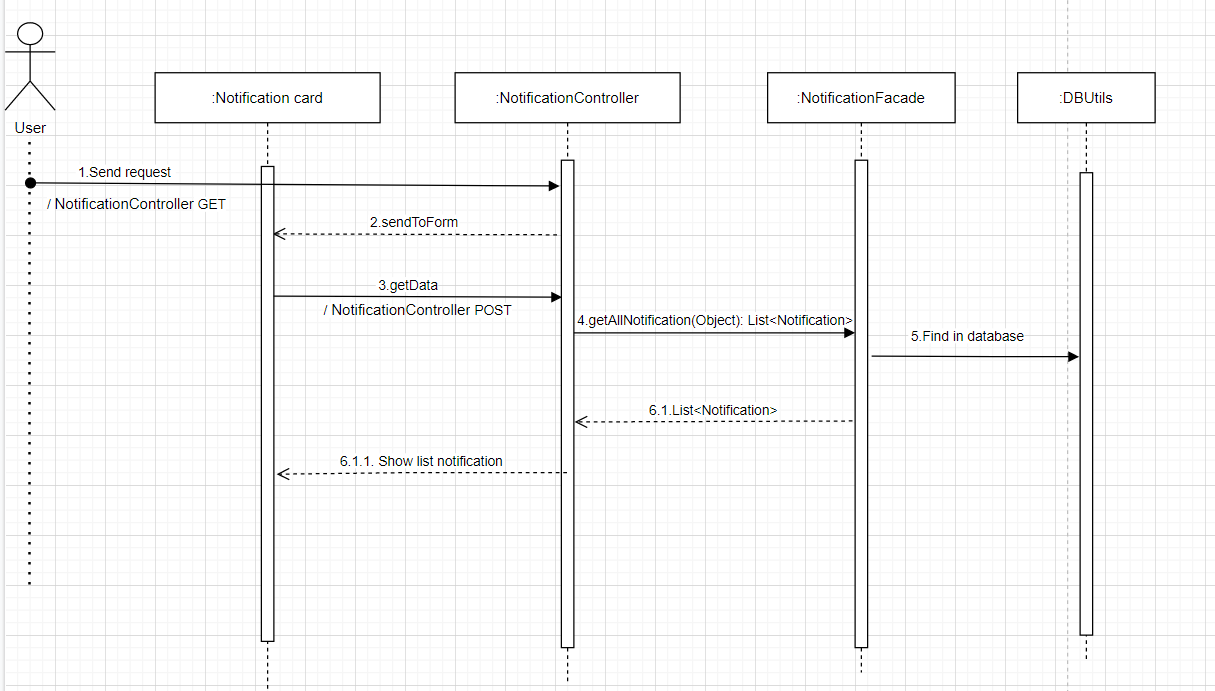


### b. Class Specifications

#### NotificationFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoNotificationFromSQL* | *get information of employee, input (ResultSet resultSet), output (Notification)* |
| *02* | *getAllNotification* | *get all Notification, input (Connection connection, Object object, Object action), output (List<Notification>)* |
| *03* | *addNotification* | *add Notification, input (Connection connection ,Notification notification), output (boolean)* |
| *04* | *updateNotification* | *update Notification, input (Connection connection, Notification notification), output (boolean)* |
| *05* | *countNotification* | *count number of Notification, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



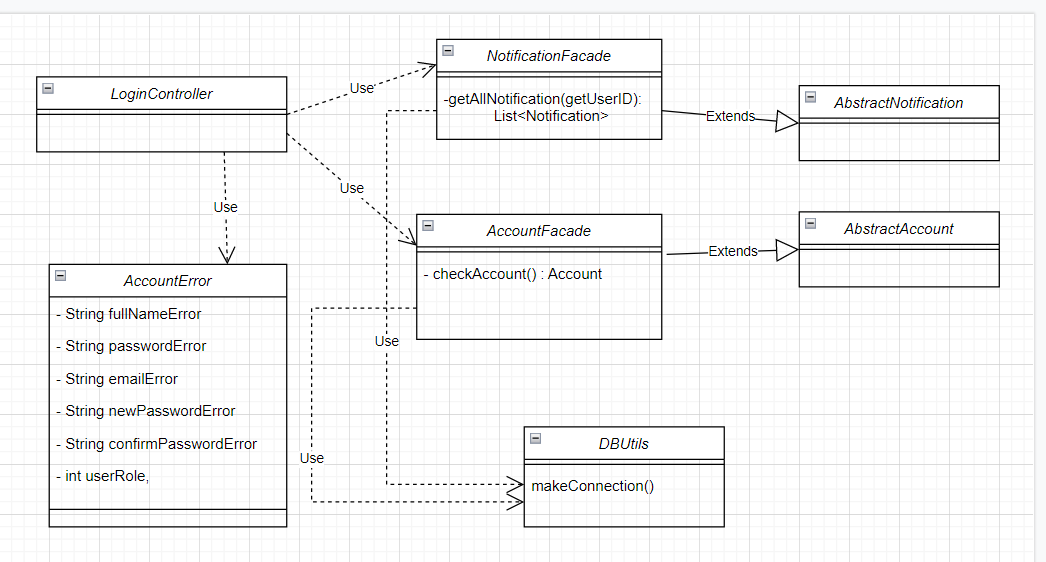
### d. Database queries

getAllNotification(): List<Notification>

+"SELECT \* FROM Notify WHERE UserID = ?";

## 24. Announcement / Delete Notification

### a. Class Diagram

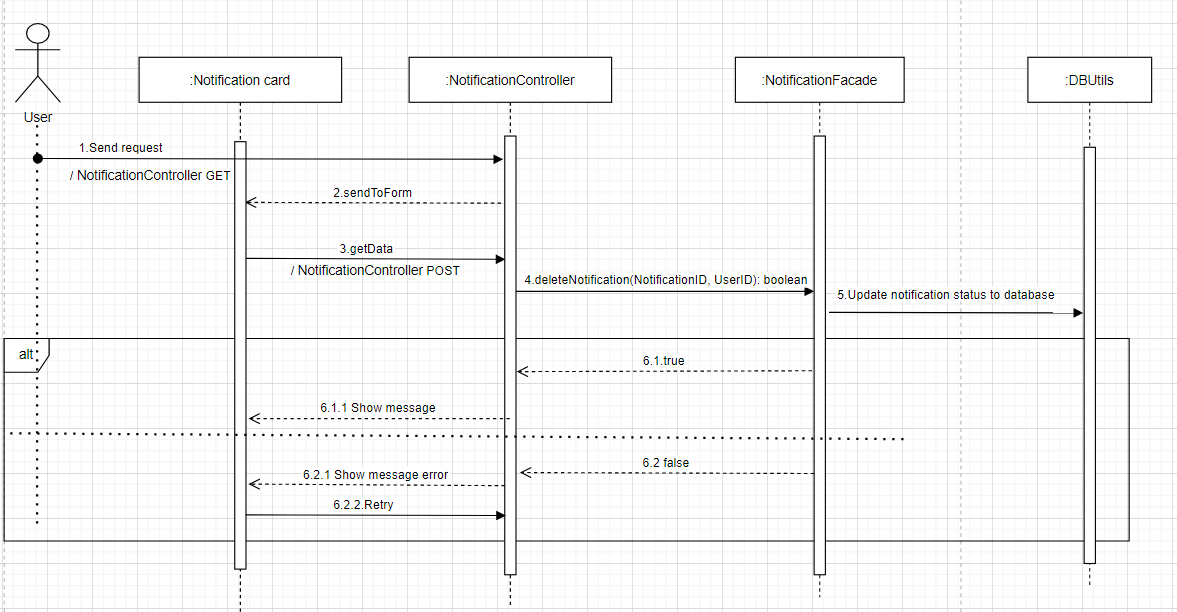


### b. Class Specifications

#### NotificationFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoNotificationFromSQL* | *get information of employee, input (ResultSet resultSet), output (Notification)* |
| *02* | *getAllNotification* | *get all Notification, input (Connection connection, Object object, Object action), output (List<Notification>)* |
| *03* | *addNotification* | *add Notification, input (Connection connection ,Notification notification), output (boolean)* |
| *04* | *updateNotification* | *update Notification, input (Connection connection, Notification notification), output (boolean)* |
| *05* | *countNotification* | *count number of Notification, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



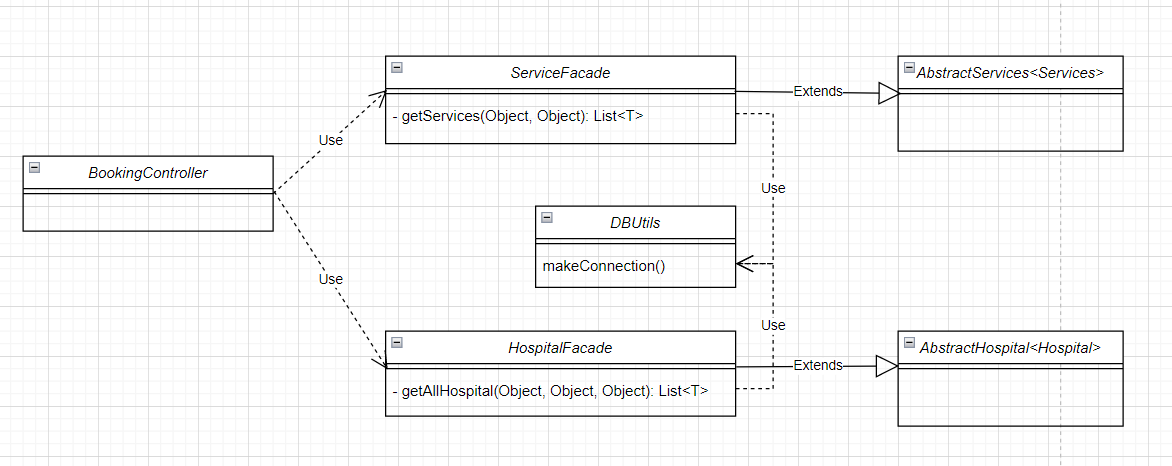
### d. Database queries

updateNotification(): boolean

+"UPDATE Notify SET NotifyStatus = 1 WHERE NotifyID = ? AND UserID = ?";

## 25. Booking Management / Check Booking Info

### a. Class Diagram

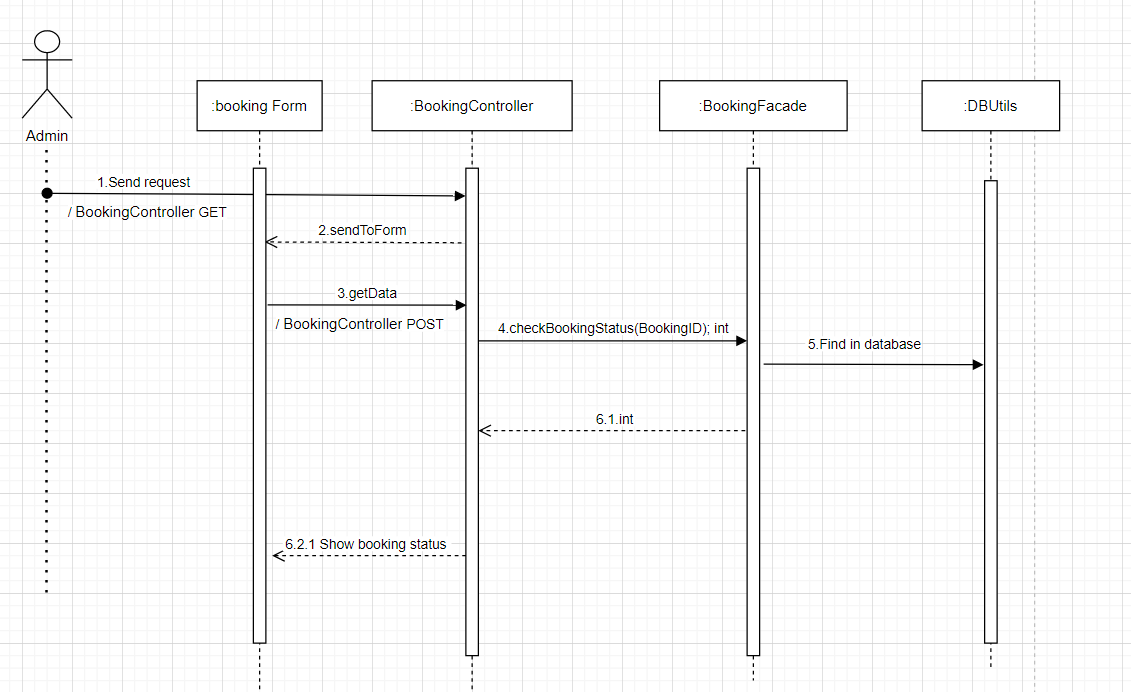


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



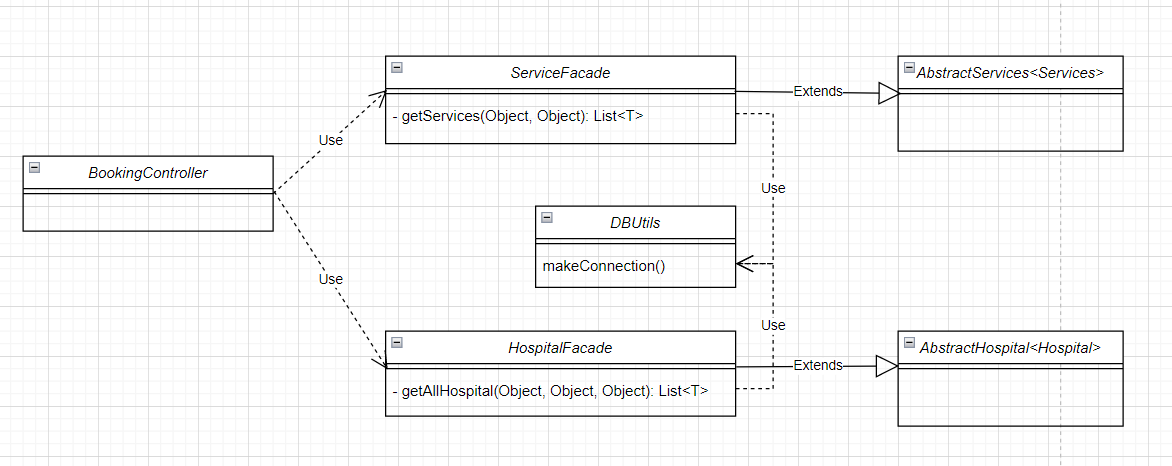
### d. Database queries

checkBookingStatus(BookingID): int:

+"SELECT BookingStatus FROM Booking WHERE BookingID = ?";

## 26. Booking Management / Confirm Booking

### a. Class Diagram

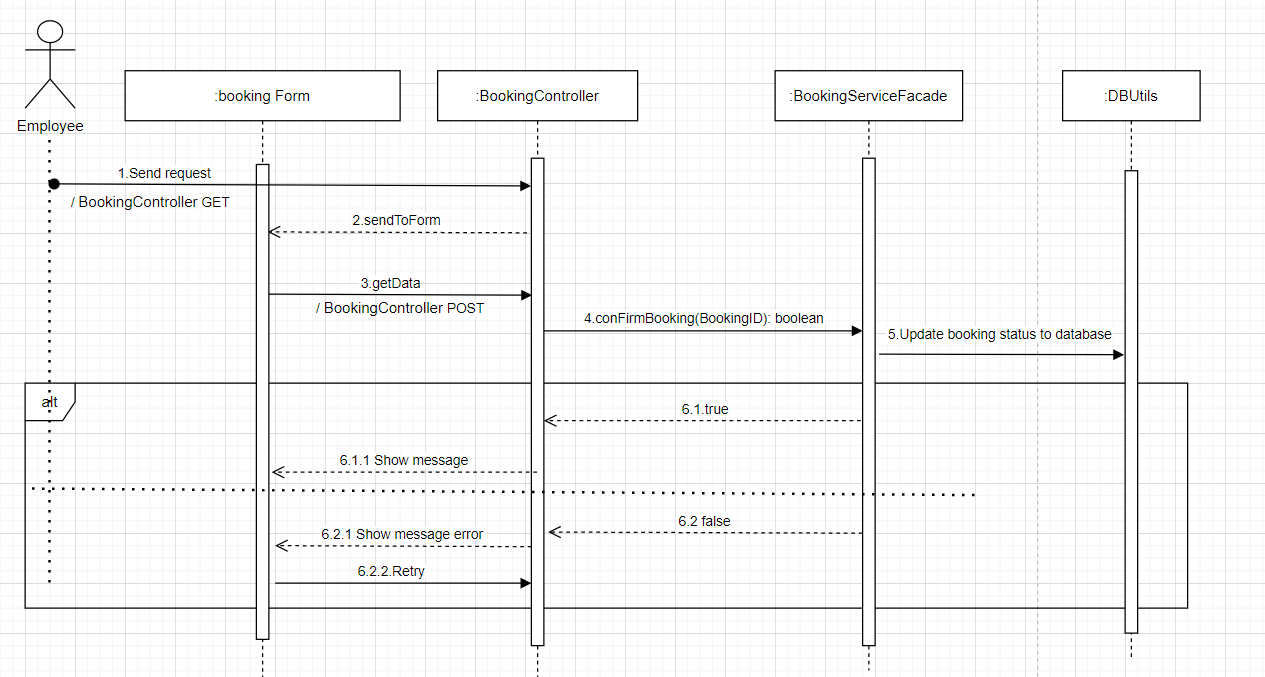


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



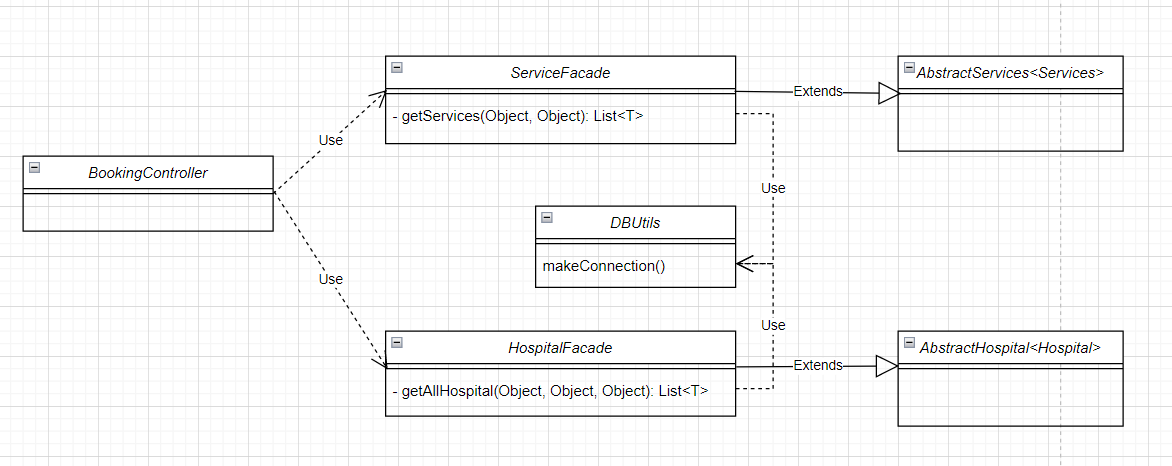
### d. Database queries

confirmBooking (BookingID): Boolean:

+UPDATE BookingStatus FROM Booking WHERE BookingID = ?";

## 27. Booking Management / Export Booking Info

### a. Class Diagram

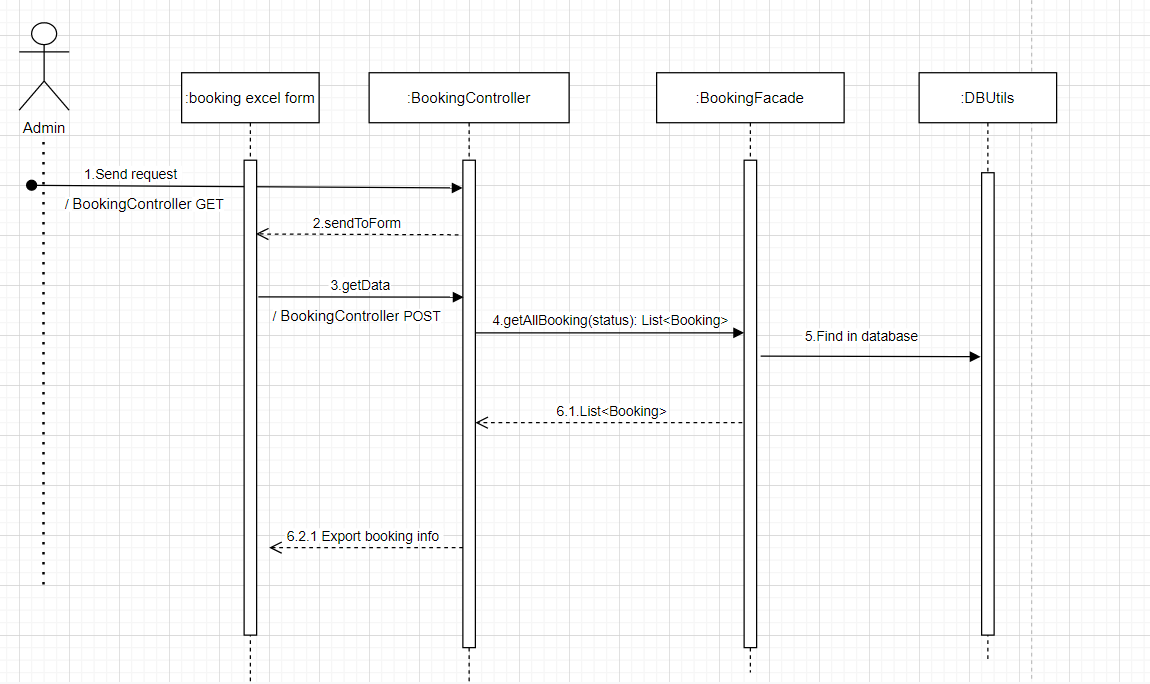


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



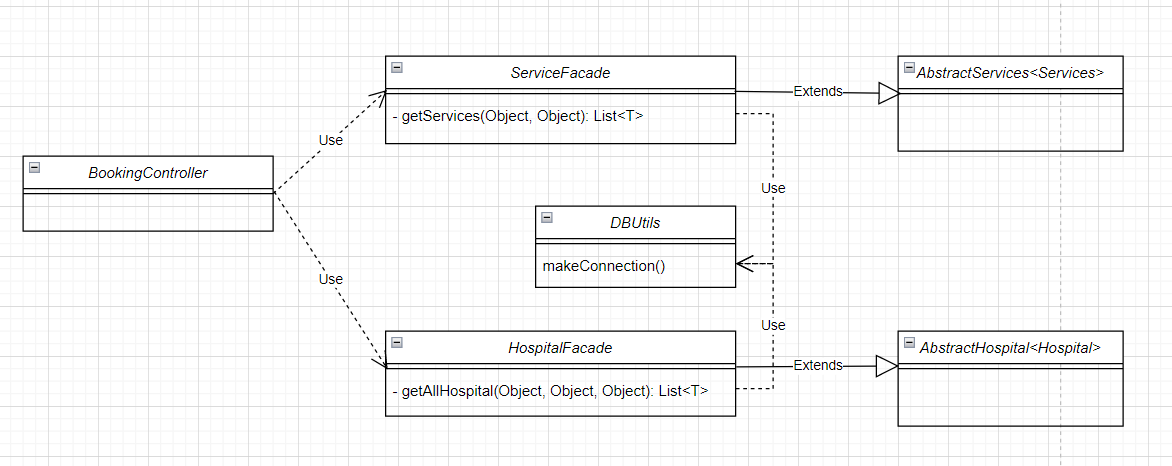
### d. Database queries

getAllBooking(): List<Booking>:

## +"SELECT \* FROM Booking WHERE BookingStatus = 1";

## 28. Booking Management / Export Booking Info

### a. Class Diagram

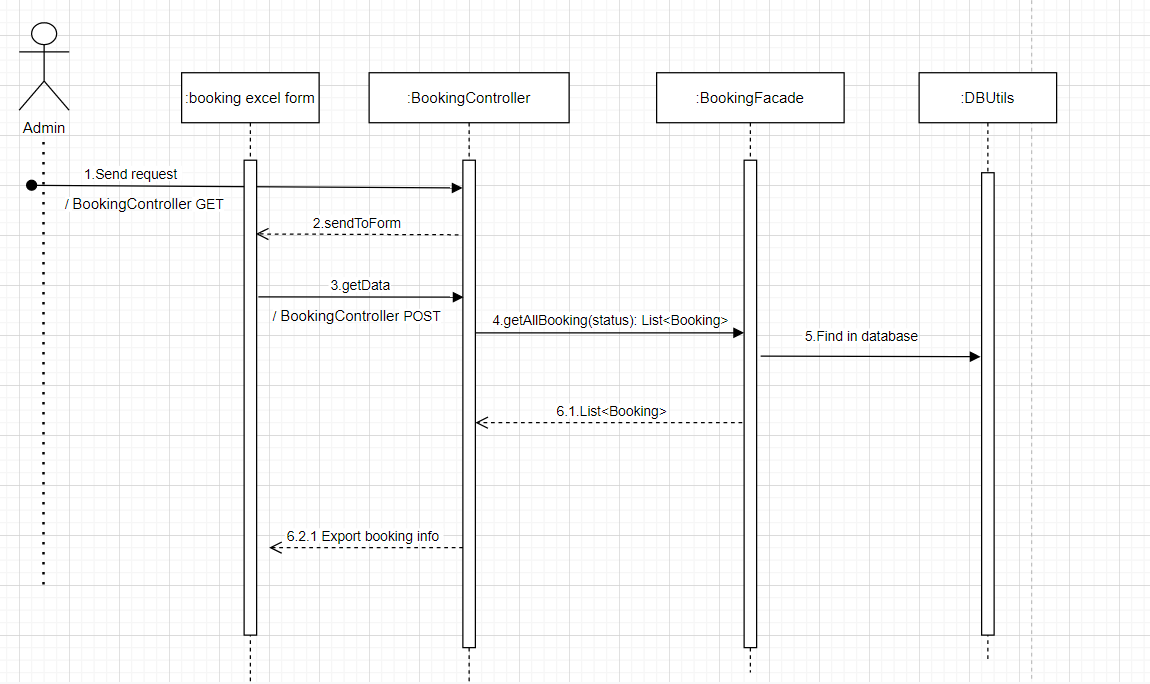


### b. Class Specifications

#### BookingFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoBookingFromSQL* | *get booking information, input (ResultSet resultSet),*  *output (Booking)* |
| *02* | *getAllBooking* | *get all booking, input (Connection connection, Object action, Object status), output (List<Booking>)* |
| *03* | *addBooking* | *add new Booking, input (Connection connection, Booking booking), output (boolean)* |
| *04* | *updateBooking* | *update booking, input (Connection connection,Booking booking), output (boolean)* |
| *05* | *countBooking* | *count number of booking, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



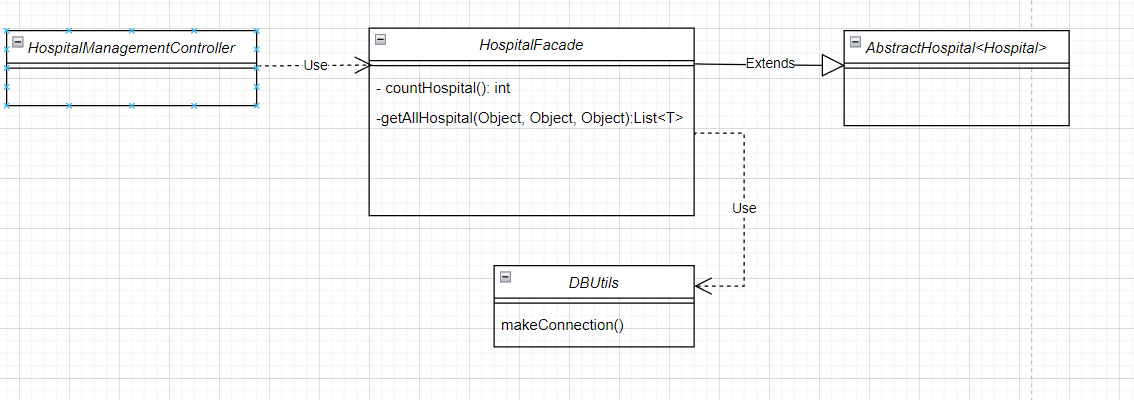
### d. Database queries

getAllBooking(): List<Booking>:

## +"SELECT \* FROM Booking WHERE BookingStatus = 1";

## 29. Clinic Management / Add Clinic

### a. Class Diagram

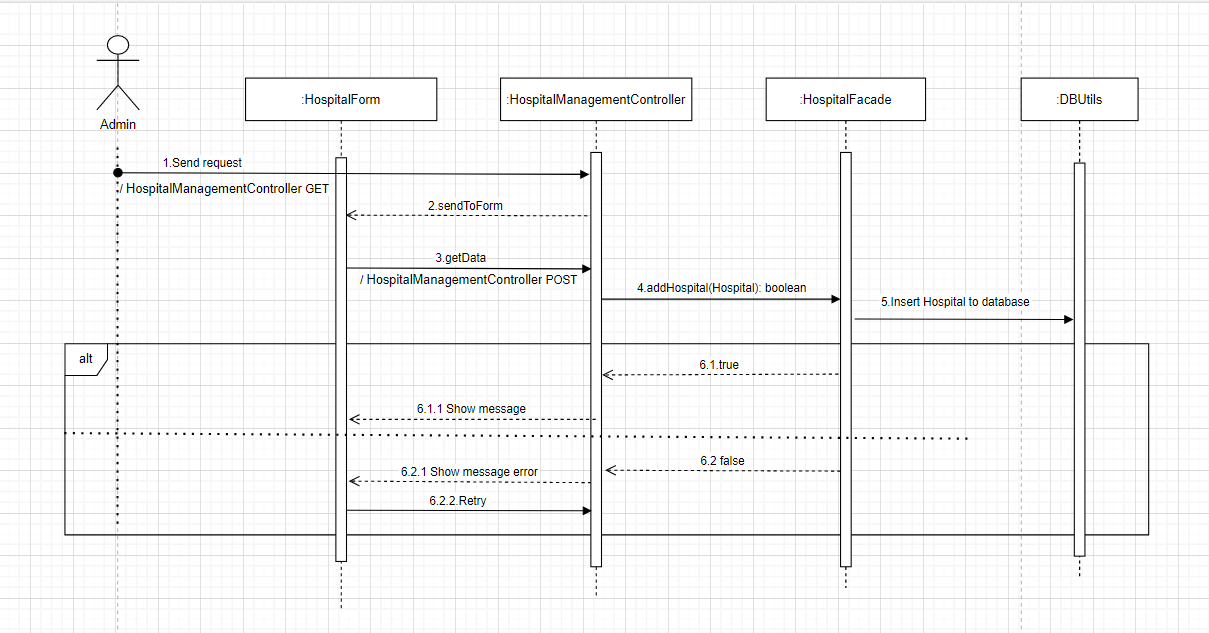


### b. Class Specifications

#### HospitalFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoHospitalFromSQL* | *get Hospital information, input (ResultSet resultSet),*  *output (Hospital)* |
| *02* | *getAllHospital* | *get all Hospital information, input (Connection connection, Object object, Object action, Object status), output (List<Hospital>)* |
| *03* | *addHospital* | *add new Hospital , input (Connection connection, Hospital hospital), output (boolean)* |
| *04* | *updateHospital* | *update Hospital information, input (Connection connection, Object hospitalID), output (boolean)* |
| *05* | *countHospital* | *count Hospital , input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



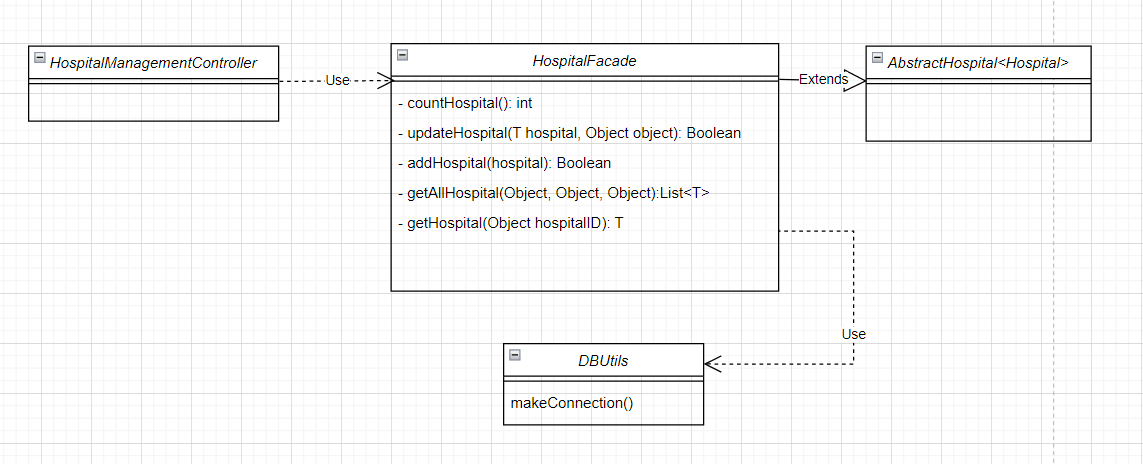
### d. Database queries

addHospital (Hospital): boolean:

+“INSERT INTO Hospital(HospitalID, HospitalName, HospitalPhone, HospitalAddress) VALUES(?, ?, ?, ?)";

## 30. Clinic Management / Update Clinic

### a. Class Diagram

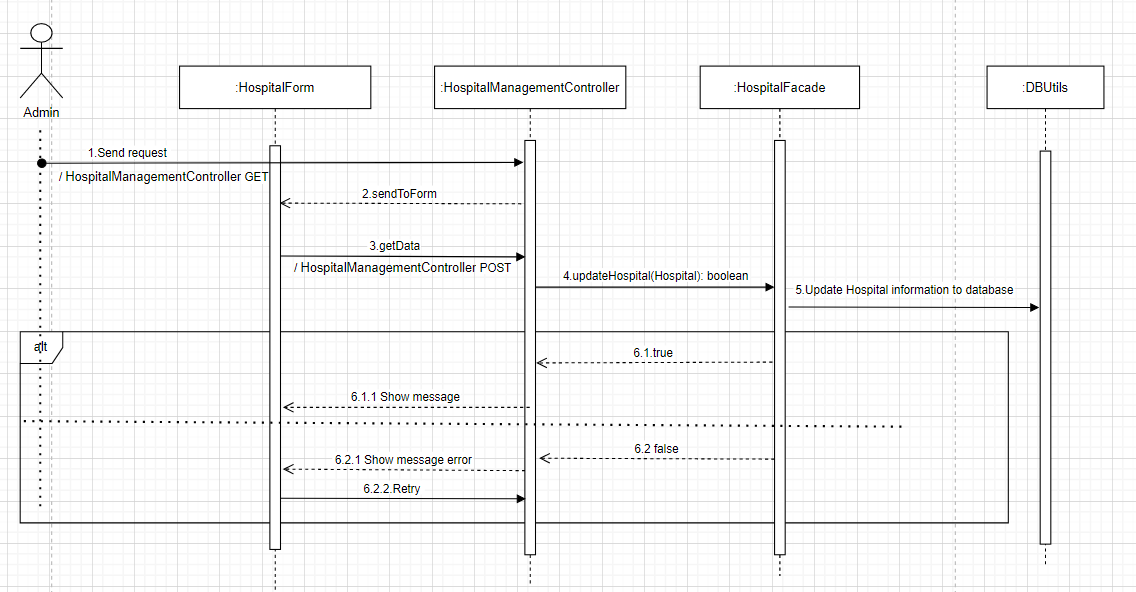


### b. Class Specifications

#### HospitalFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoHospitalFromSQL* | *get Hospital information, input (ResultSet resultSet),*  *output (Hospital)* |
| *02* | *getAllHospital* | *get all Hospital information, input (Connection connection, Object object, Object action, Object status), output (List<Hospital>)* |
| *03* | *addHospital* | *add new Hospital , input (Connection connection, Hospital hospital), output (boolean)* |
| *04* | *updateHospital* | *update Hospital information, input (Connection connection, Object hospitalID), output (boolean)* |
| *05* | *countHospital* | *count Hospital , input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



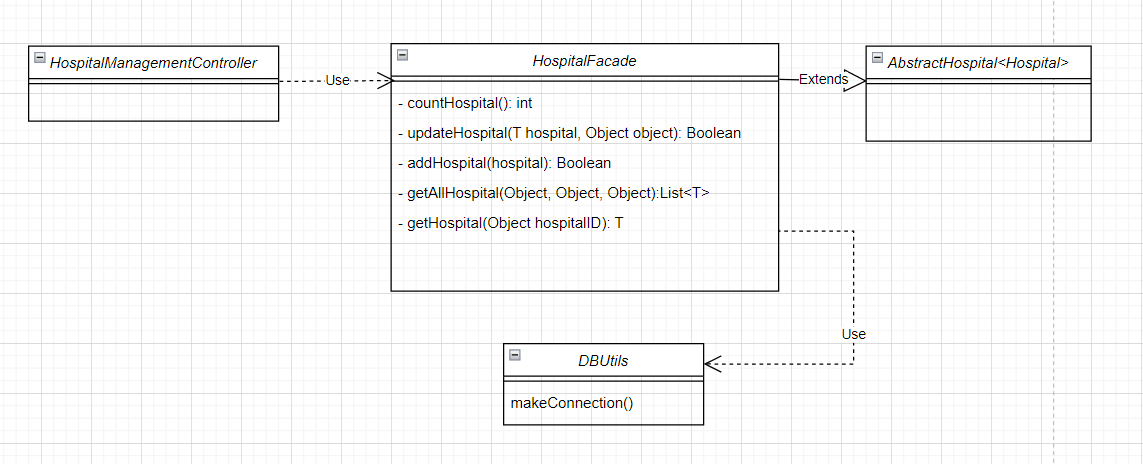
### d. Database queries

updateHospital (Hospital): boolean:

+"UPDATE Hospital SET HospitalName = ?, HospitalPhone = ?, HospitalAddress = ? WHERE HospitalID = ?";

## 31. Clinic Management / Delete Clinic

### a. Class Diagram

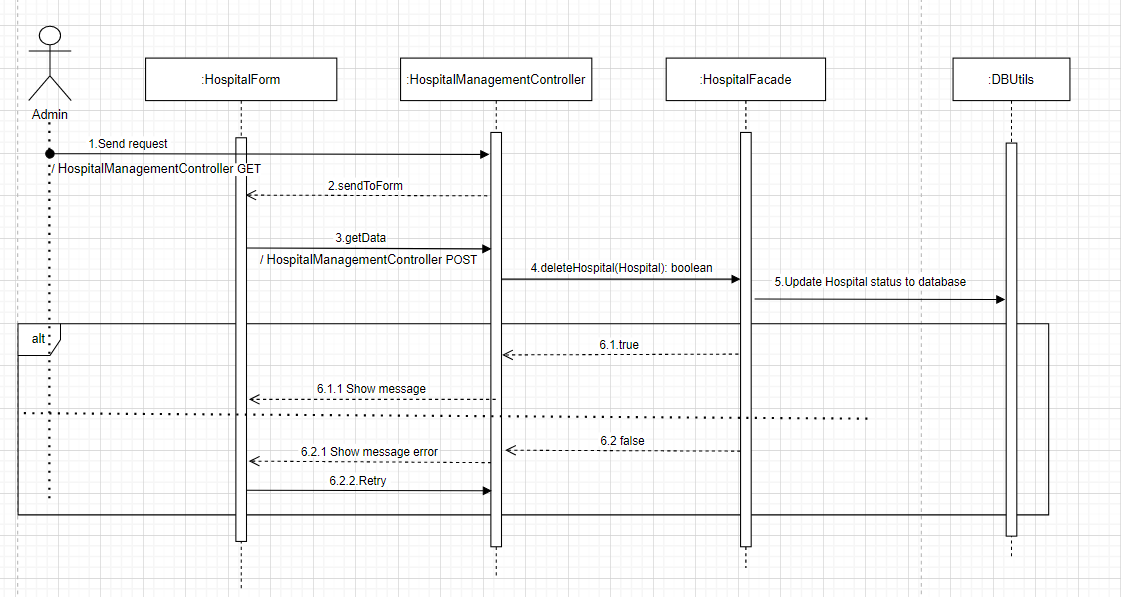


### b. Class Specifications

#### HospitalFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoHospitalFromSQL* | *get Hospital information, input (ResultSet resultSet),*  *output (Hospital)* |
| *02* | *getAllHospital* | *get all Hospital information, input (Connection connection, Object object, Object action, Object status), output (List<Hospital>)* |
| *03* | *addHospital* | *add new Hospital , input (Connection connection, Hospital hospital), output (boolean)* |
| *04* | *updateHospital* | *update Hospital information, input (Connection connection, Object hospitalID), output (boolean)* |
| *05* | *countHospital* | *count Hospital , input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



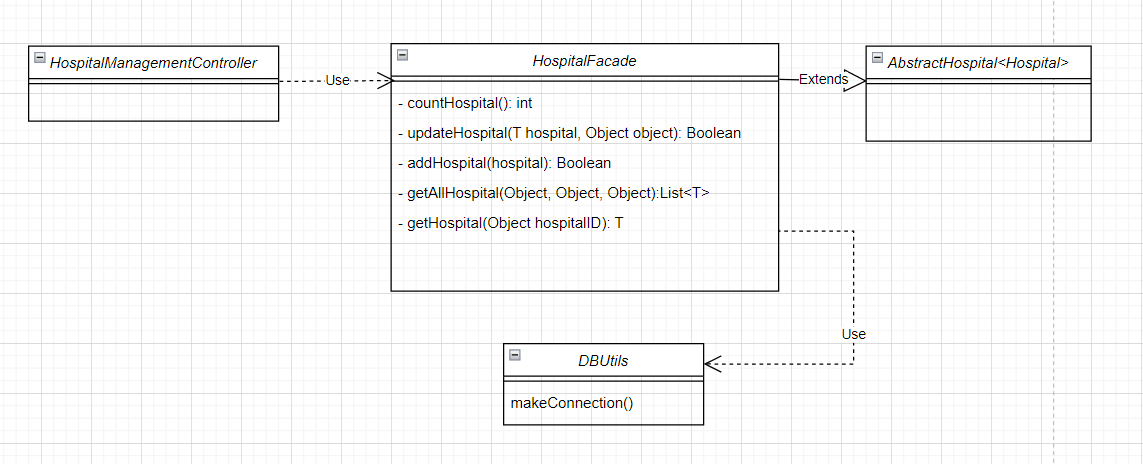
### d. Database queries

updateHospital (Hospital): boolean:

## +"UPDATE Hospital SET HospitalStatus = ? WHERE HospitalID = ?";

## 32. Clinic Management / Search Clinic

### a. Class Diagram

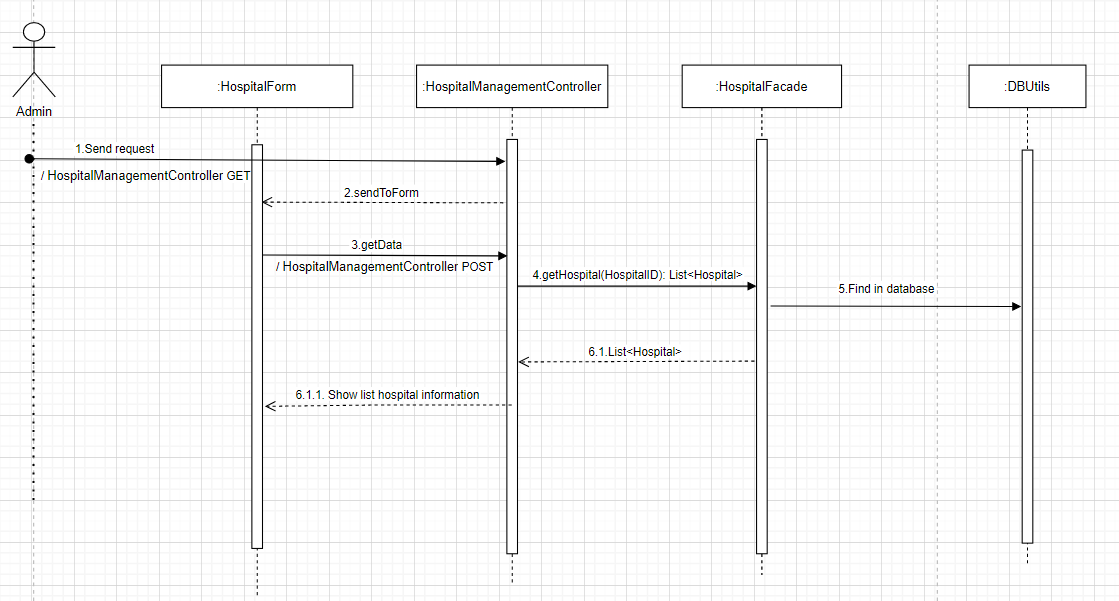


### b. Class Specifications

#### HospitalFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoHospitalFromSQL* | *get Hospital information, input (ResultSet resultSet),*  *output (Hospital)* |
| *02* | *getAllHospital* | *get all Hospital information, input (Connection connection, Object object, Object action, Object status), output (List<Hospital>)* |
| *03* | *addHospital* | *add new Hospital , input (Connection connection, Hospital hospital), output (boolean)* |
| *04* | *updateHospital* | *update Hospital information, input (Connection connection, Object hospitalID), output (boolean)* |
| *05* | *countHospital* | *count Hospital , input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



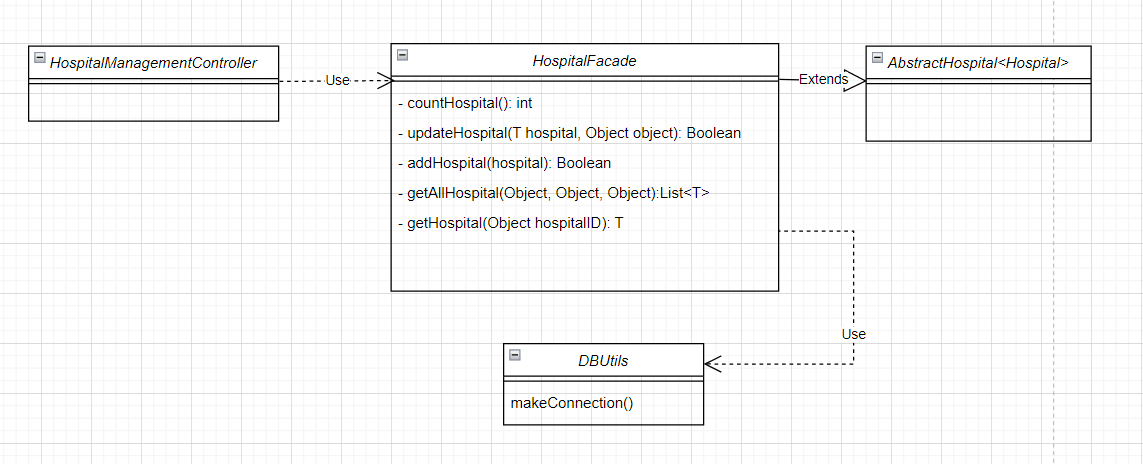
### d. Database queries

getHospital(HospitalID): Hospital:

+"SELECT \* FROM Hospital WHERE HospitalID = ?";

## 33. Customer Service/ Write Feedback (Thiếu controller)

### a. Class Diagram

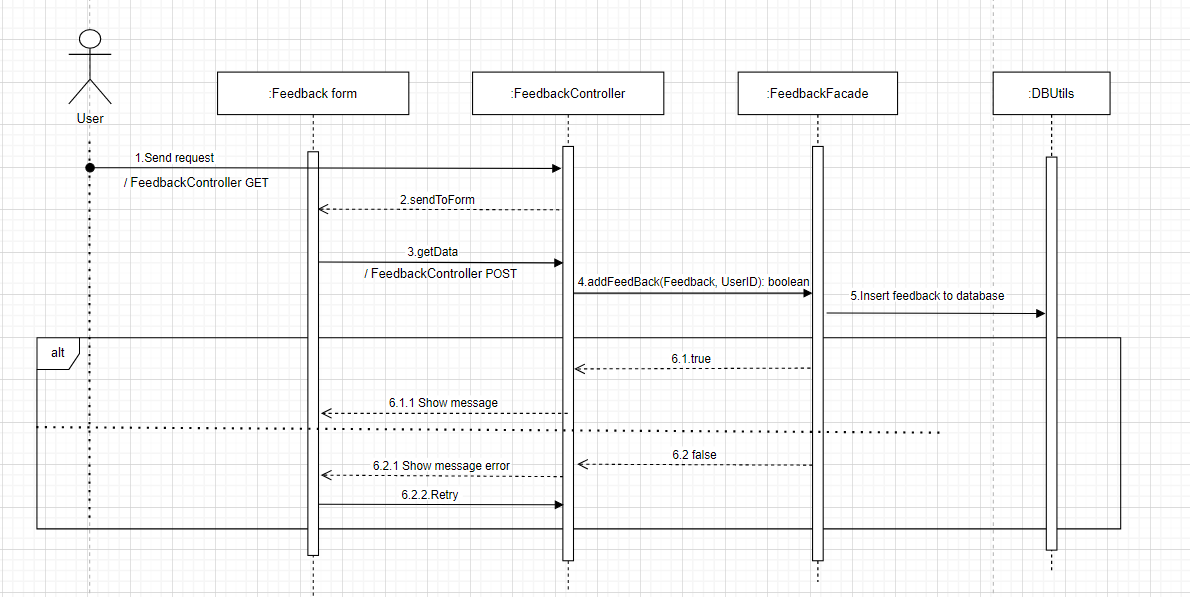


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoAccountFromSQL* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



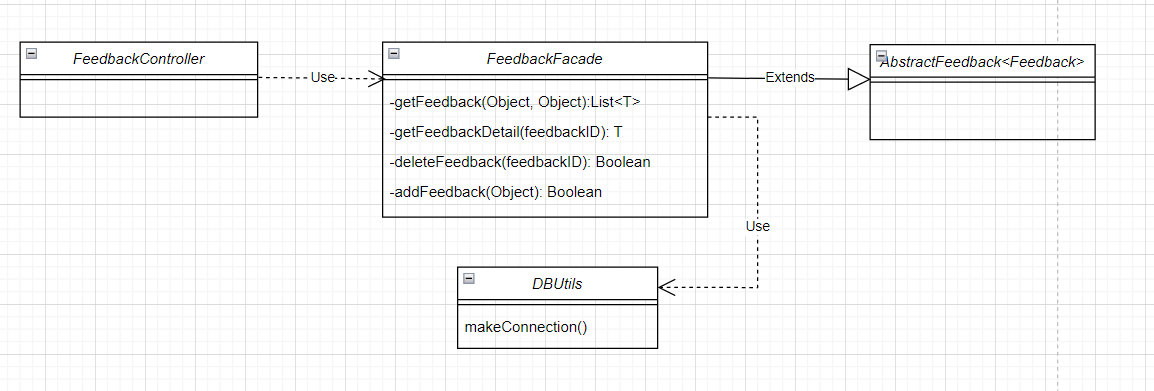
### d. Database queries

addFeedback (Feedback, UserID): boolean:

+??

## 34. Customer Service/ Update Feedback (Thiếu controller)

### a. Class Diagram

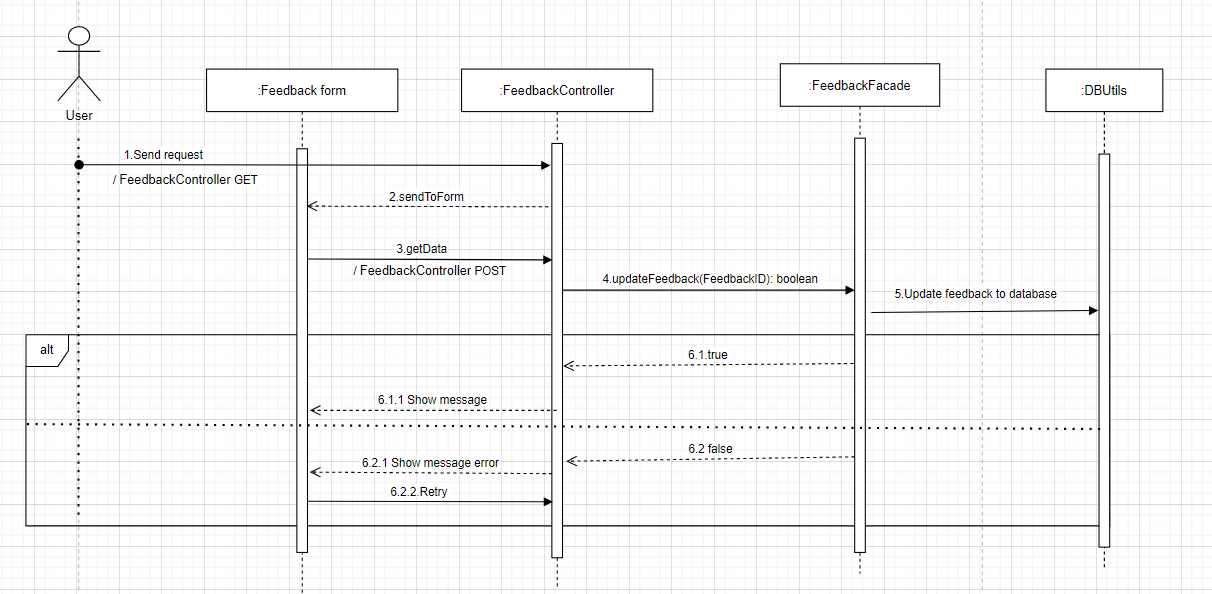


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedBack* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



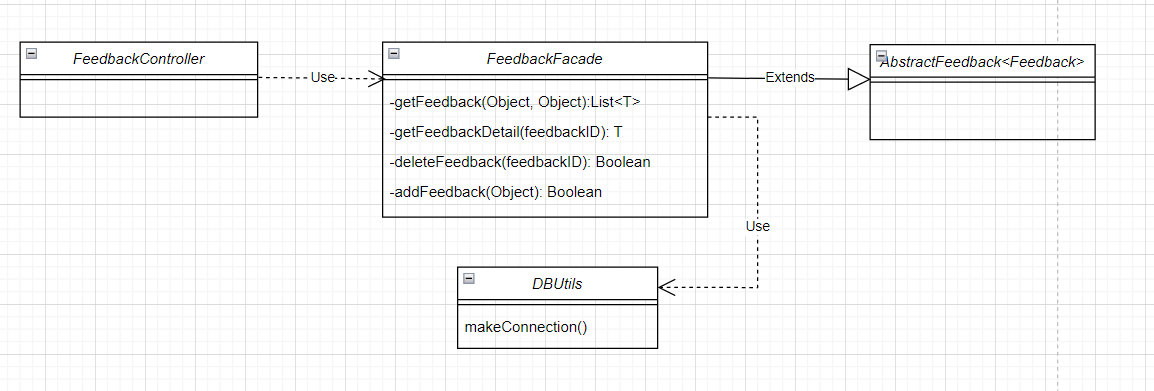
### d. Database queries

UpdateFeedback (FeedbackID): boolean:

+??

## 35. Customer Service/ Delete Feedback (Thiếu controller)

### a. Class Diagram

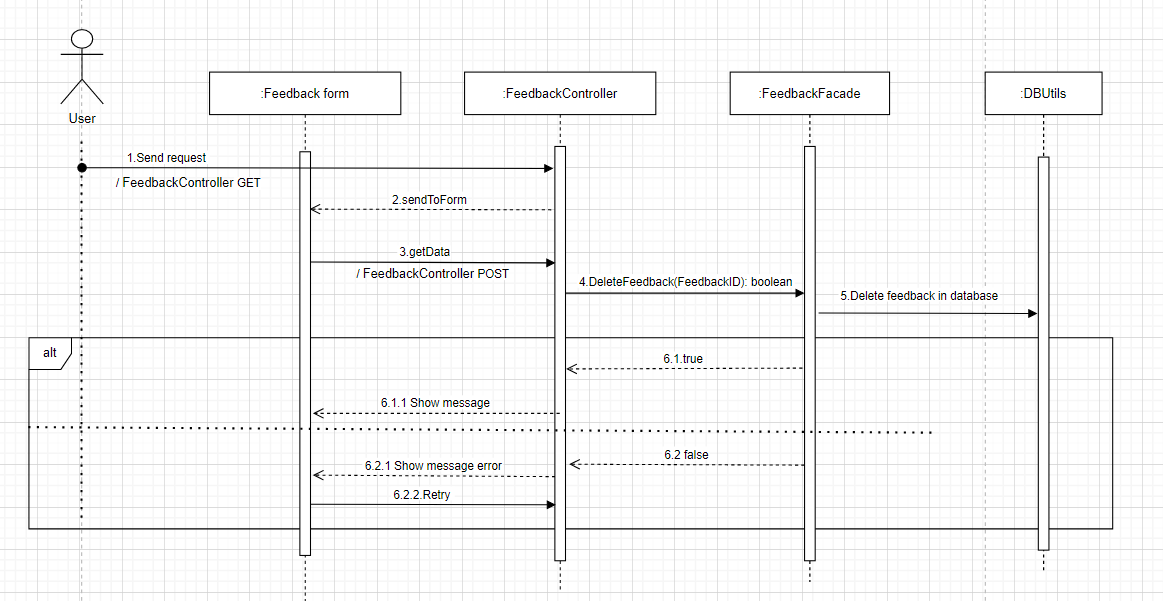


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedBack* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



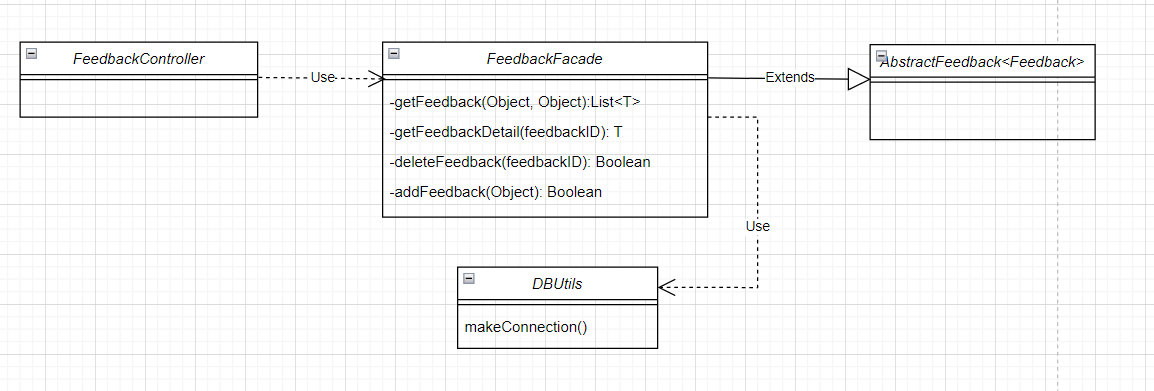
### d. Database queries

deleteFeedback (FeedbackID): boolean:

+??

## 36. User Management/ View Feedback

### a. Class Diagram

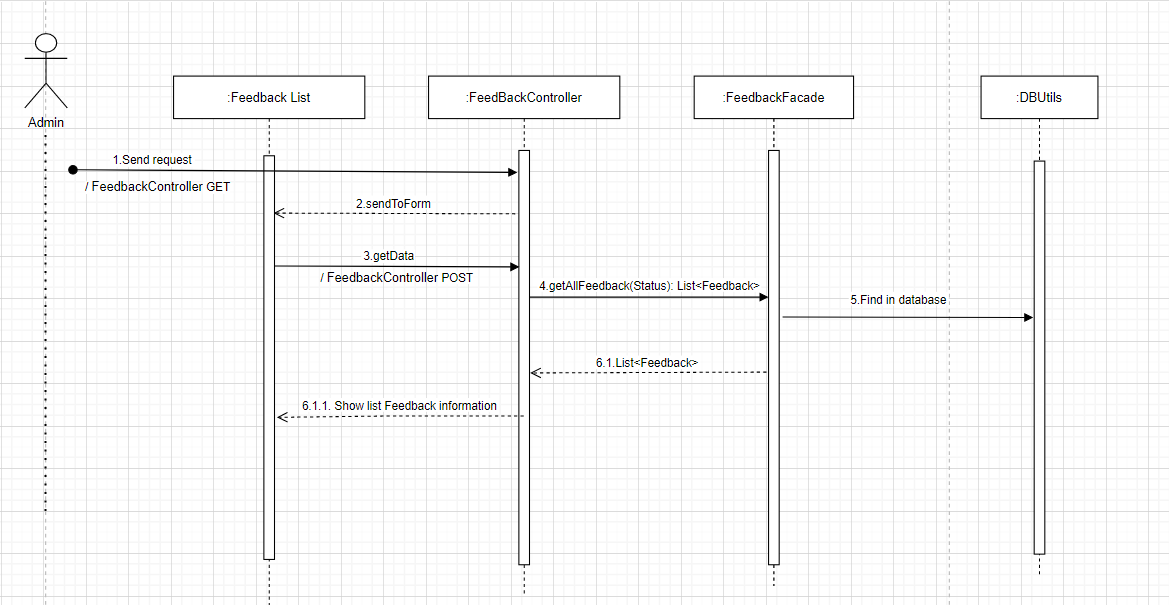


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedBack* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



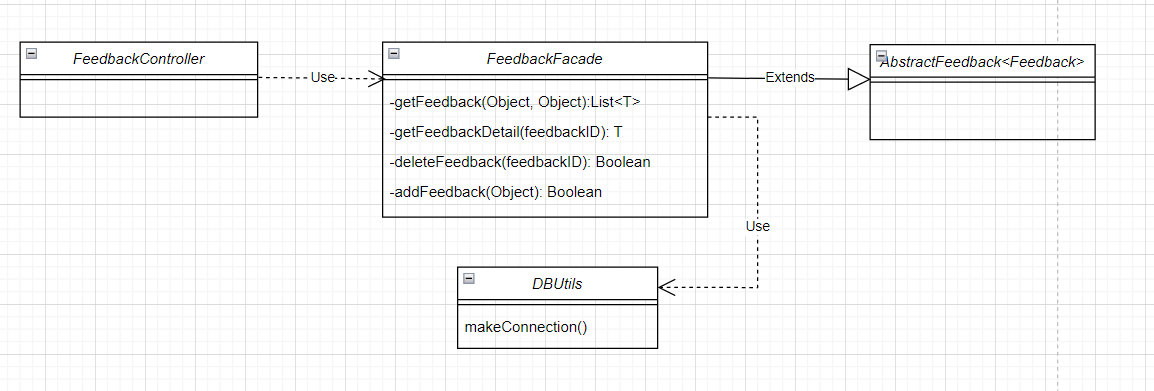
### d. Database queries

getAllFeedback():List<Feedback>:

+””

## 37. User Management / Search Feedback (Thiếu controller)

### a. Class Diagram



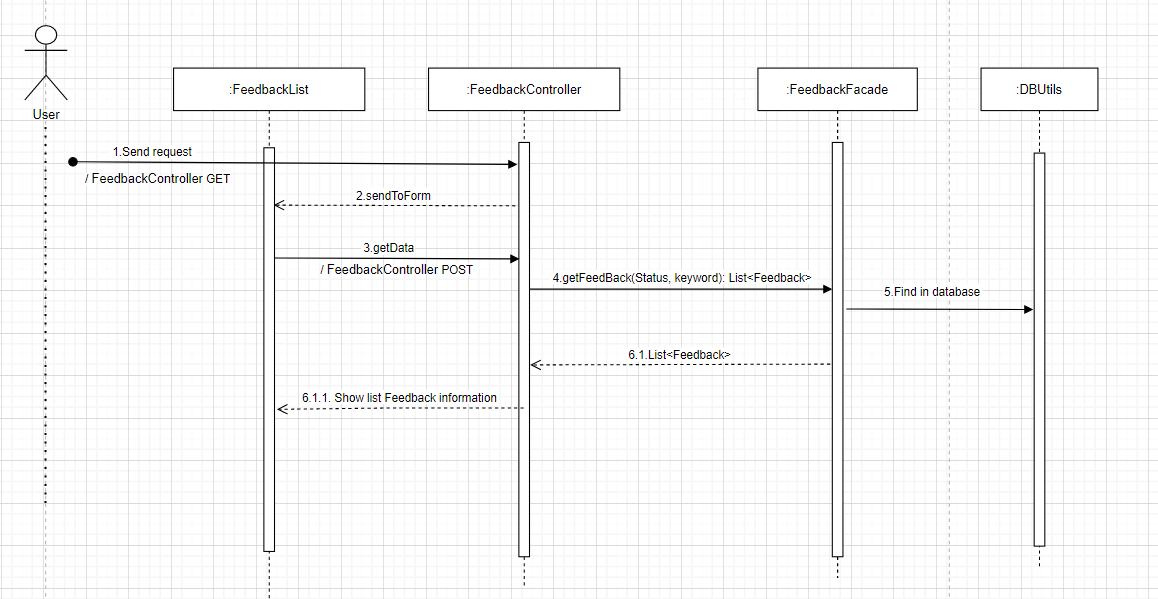
### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedBack* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### 

### c. Sequence Diagram(s)



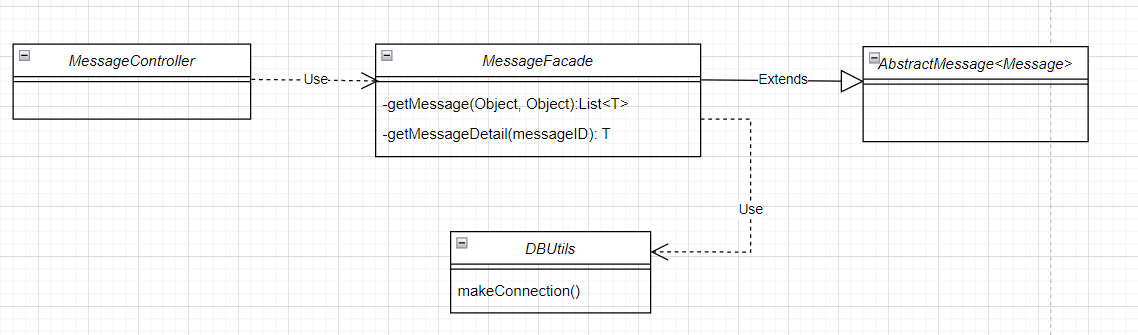
### d. Database queries

getFeedback():List<Feedback>:

+””

## 38. Customer Service/ Chat with staff

### a. Class Diagram

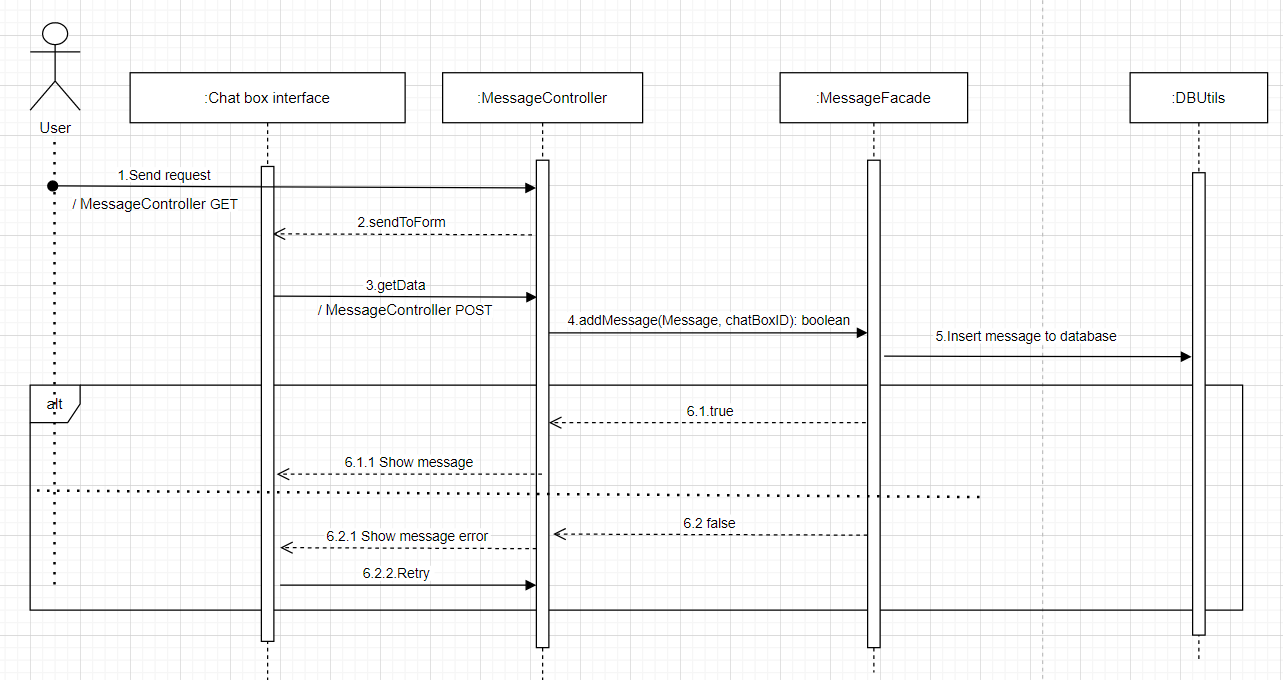


### b. Class Specifications

#### MessageFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getMessage* | *get Message information, input (ResultSet resultSet),*  *output (Message)* |
| *02* | *getAllMessage* | *get all Message information, input (Connection connection), output (List<Message>)* |
| *03* | *addMessage* | *add new Message, input (Connection connection), output (boolean)* |
| *04* | *updateMessage* | *update Message content, input (Connection connection, Message message), output (boolean)* |
| *05* | *countMessage* | *count Message, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



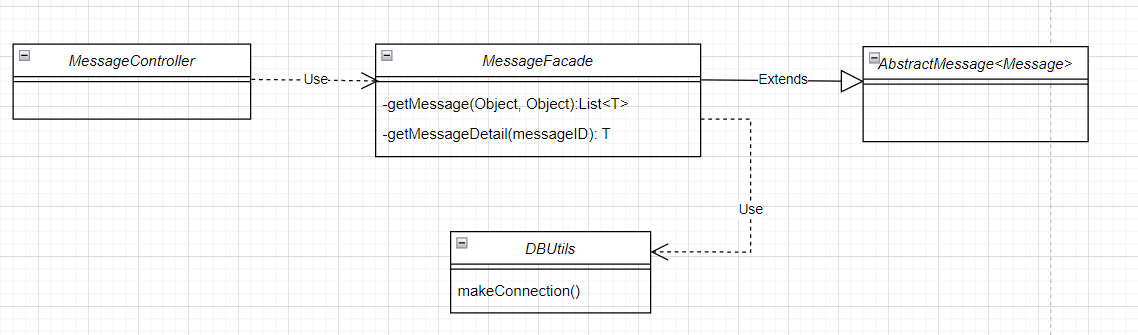
### d. Database queries

addMessage(chatBoxID, Message ): boolean

+””

## 39. Customer Service/ Reply customers

### a. Class Diagram

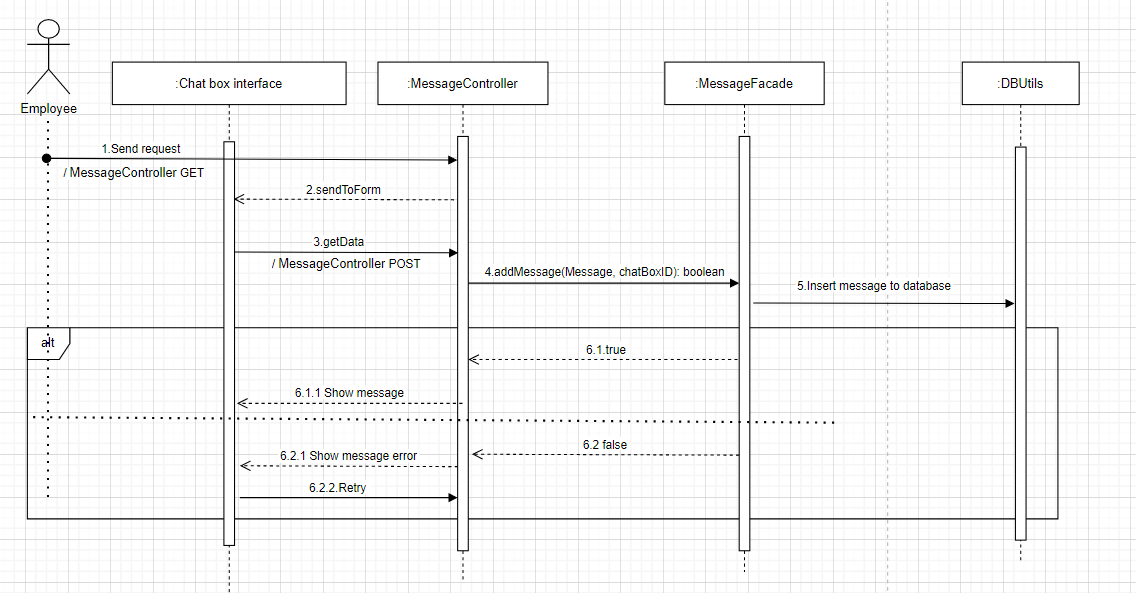


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getMessage* | *get Message information, input (ResultSet resultSet),*  *output (Message)* |
| *02* | *getAllMessage* | *get all Message information, input (Connection connection), output (List<Message>)* |
| *03* | *addMessage* | *add new Message, input (Connection connection), output (boolean)* |
| *04* | *updateMessage* | *update Message content, input (Connection connection, Message message), output (boolean)* |
| *05* | *countMessage* | *count Message, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



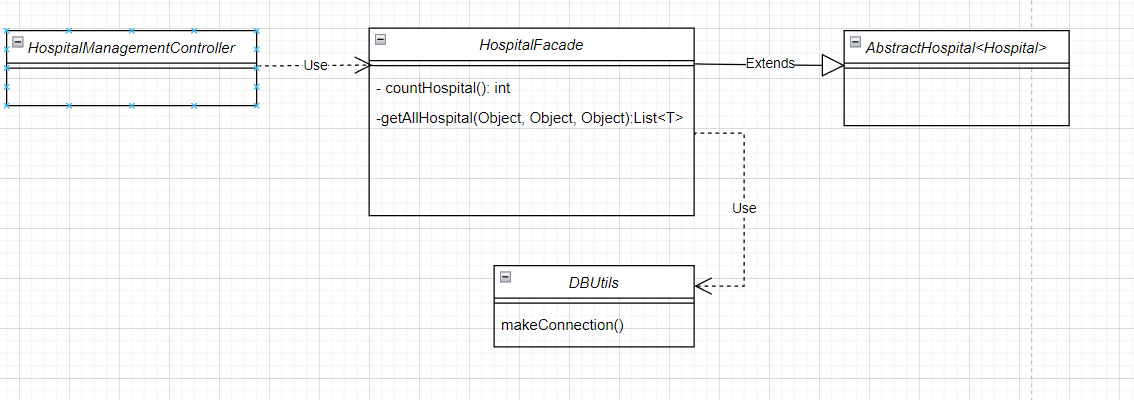
### d. Database queries

addMessage(chatBoxID, Message ): boolean

+””

## 40. User Management/ Block Account(Thiếu controller, bảng)

### a. Class Diagram

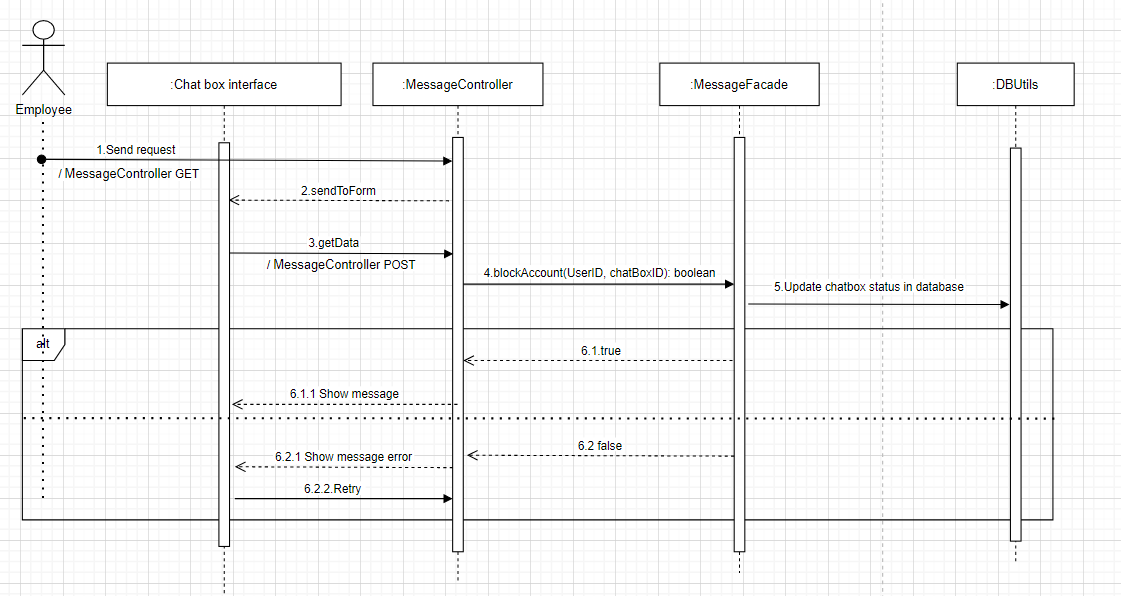


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedback* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



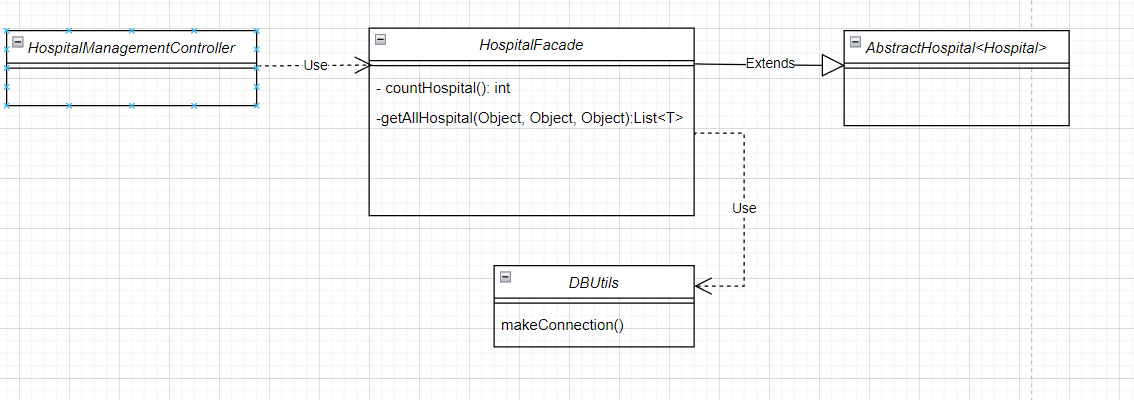
### d. Database queries

blockAccount(chatBoxID, UserID ): boolean

+””

## 41. User Management/ List of statistics (Thiếu controller, bảng)

### a. Class Diagram

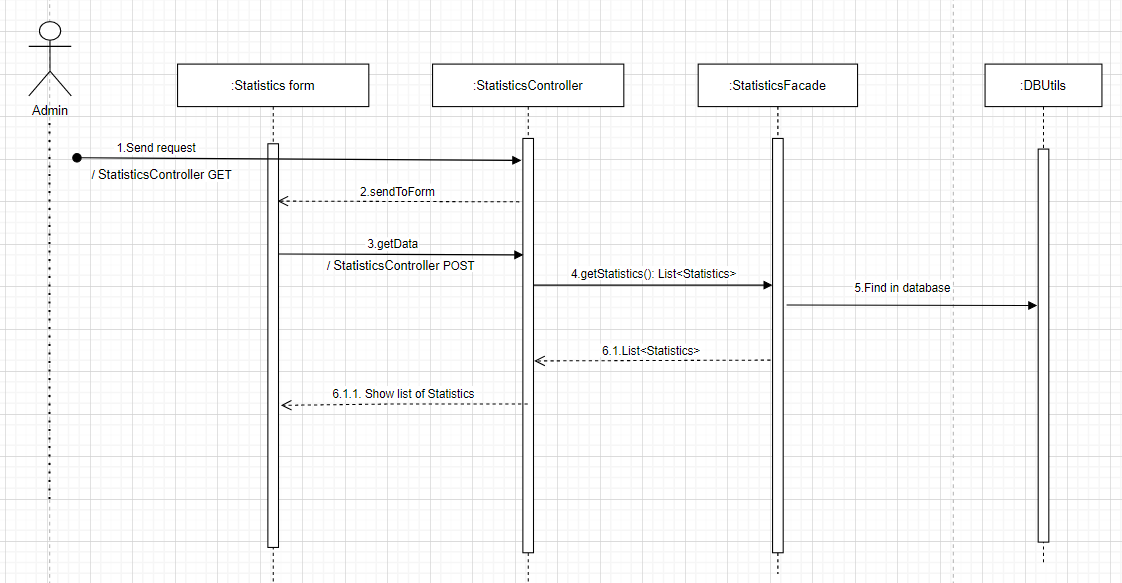


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedback* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



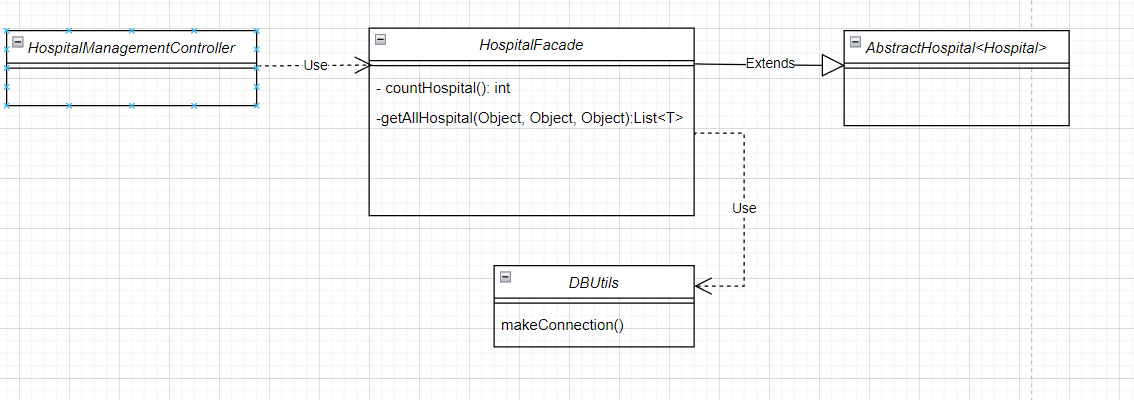
### d. Database queries

getStatistics():

+””

## 42. User Management/ List of user register (Thiếu controller, bảng)

### a. Class Diagram

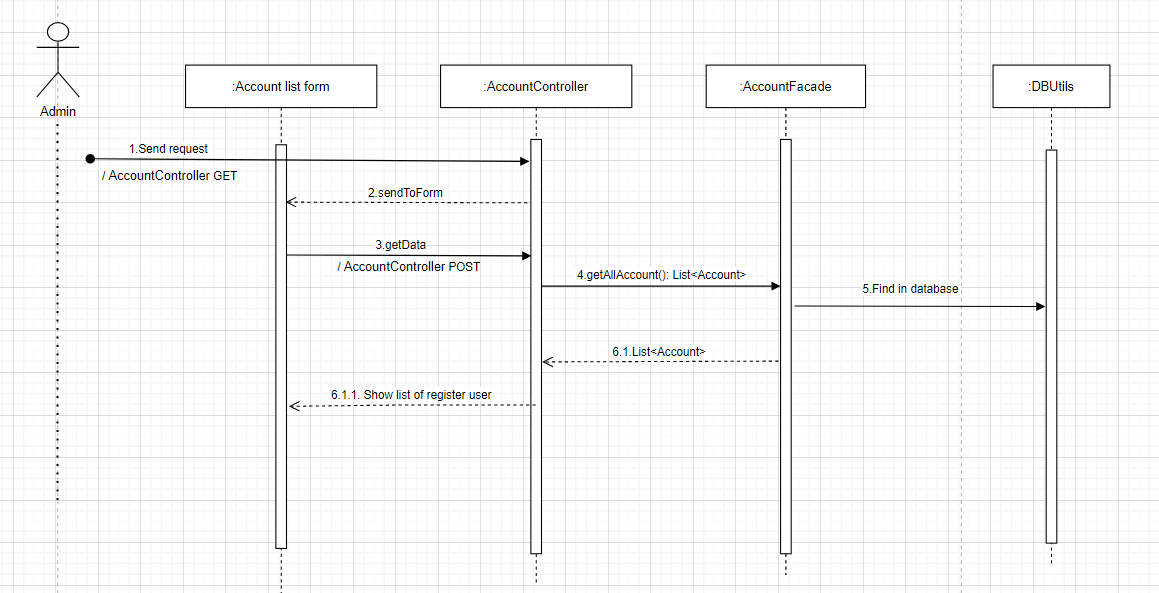


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedback* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



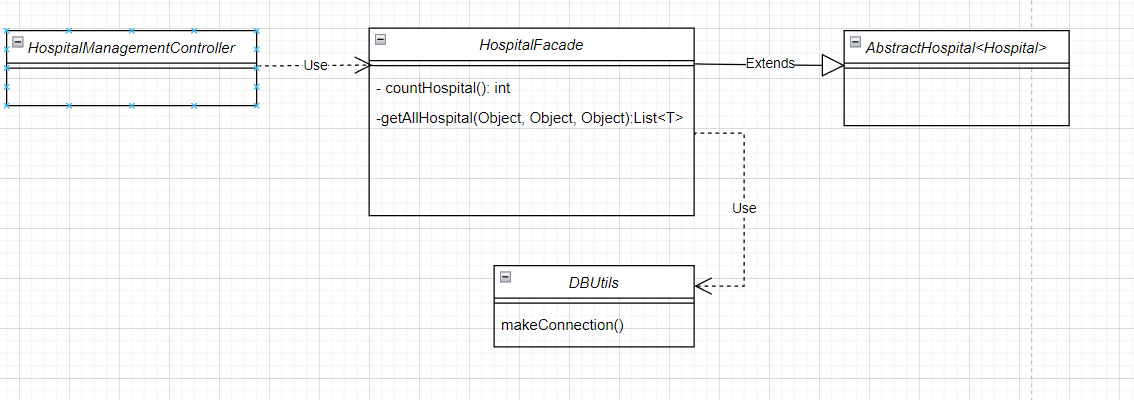
### d. Database queries

getAllAccount(): List<Account>:

+””

## 43. User Management/ Export file excel (Thiếu controller, bảng)

### a. Class Diagram

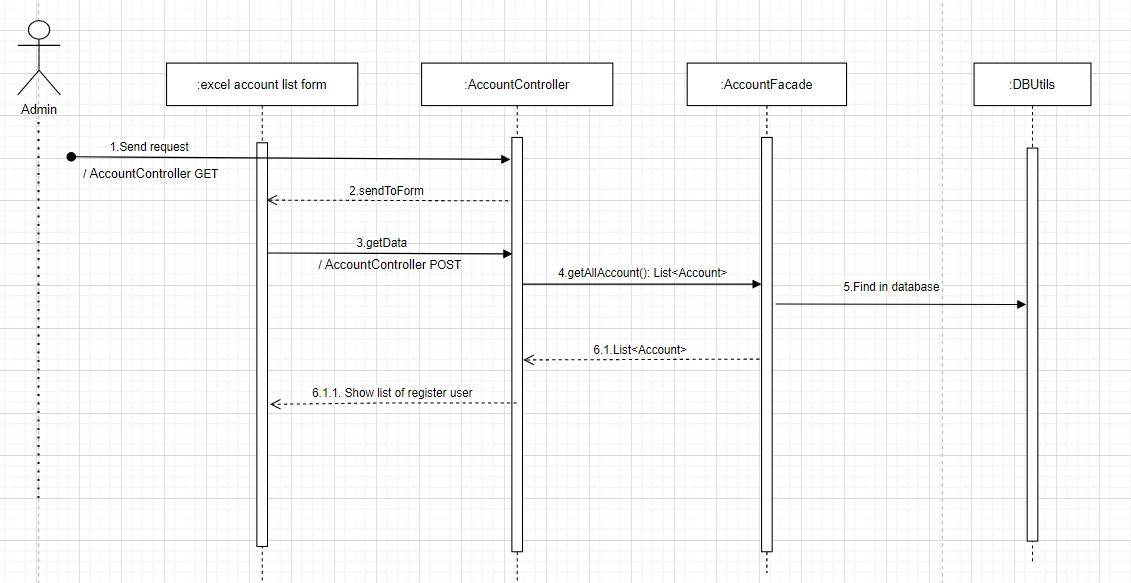


### b. Class Specifications

#### FeedBackFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getFeedback* | *get FeedBack information, input (ResultSet resultSet),*  *output (FeedBack)* |
| *02* | *getAllFeedBack* | *get all FeedBack information, input (Connection connection), output (List<FeedBack>)* |
| *03* | *addFeedBack* | *add new FeedBack, input (Connection connection), output (boolean)* |
| *04* | *updateFeedBack* | *update FeedBack information, input (Connection connection, FeedBack feedBack), output (boolean)* |
| *05* | *countFeedBack* | *count FeedBack, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



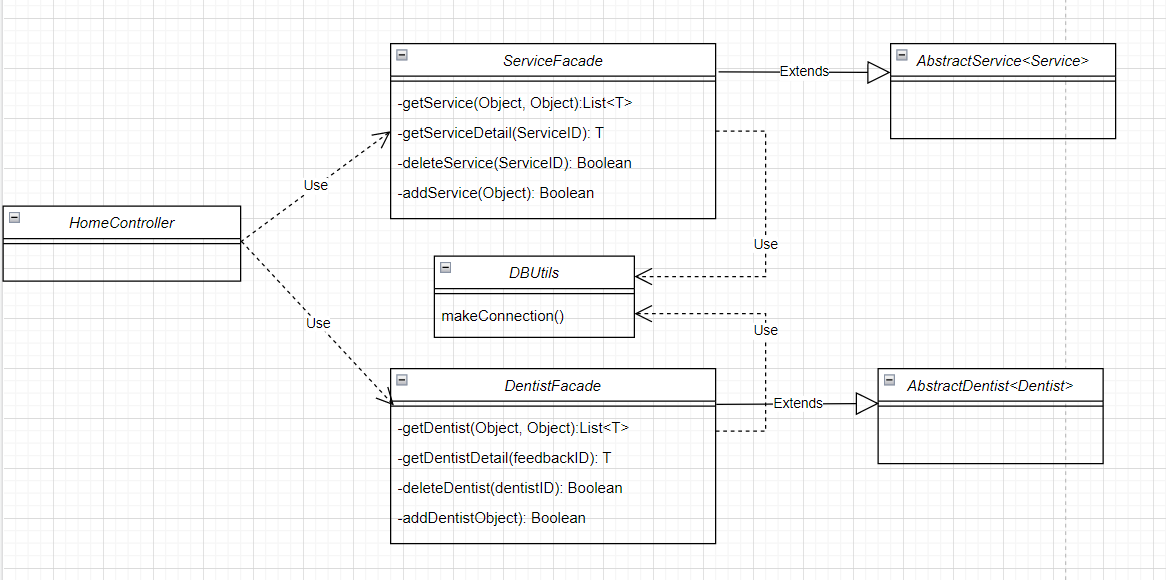
### d. Database queries

getAllAccount(): List<Account>:

+”

## 44. Overview Information/ View Introduction and overview information (Thiếu controller, bảng)

### a. Class Diagram



### b. Class Specifications

#### ServiceFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |

#### 

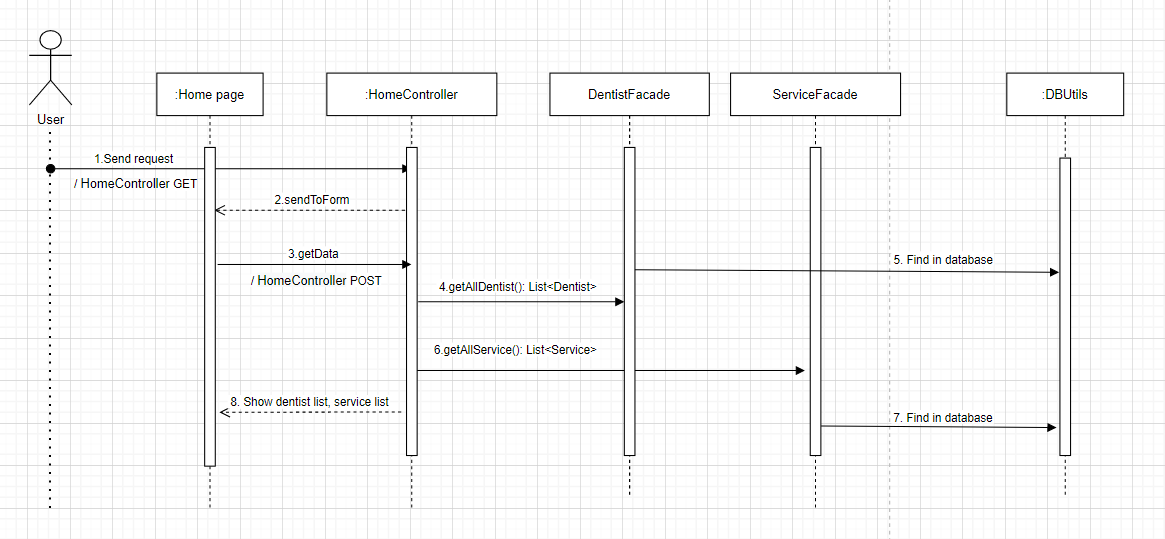
#### 

#### DentistFacade

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoDentistFromSQL* | *get dentist information, input (ResultSet resultSet),*  *output (Dentist)* |
| *02* | *getAllDentist* | *get all dentist information, input (Connection connection, Object object, Object action), output (List<Dentist>)* |
| *03* | *addDentist* | *add new dentist, input (Connection connection, Dentist dentist), output (boolean)* |
| *04* | *updateDentist* | *update dentist information, input (Connection connection, Dentist dentist), output (boolean)* |
| *05* | *countDentist* | *count dentist, input (Connection connection), output (int)* |

### 

### c. Sequence Diagram(s)



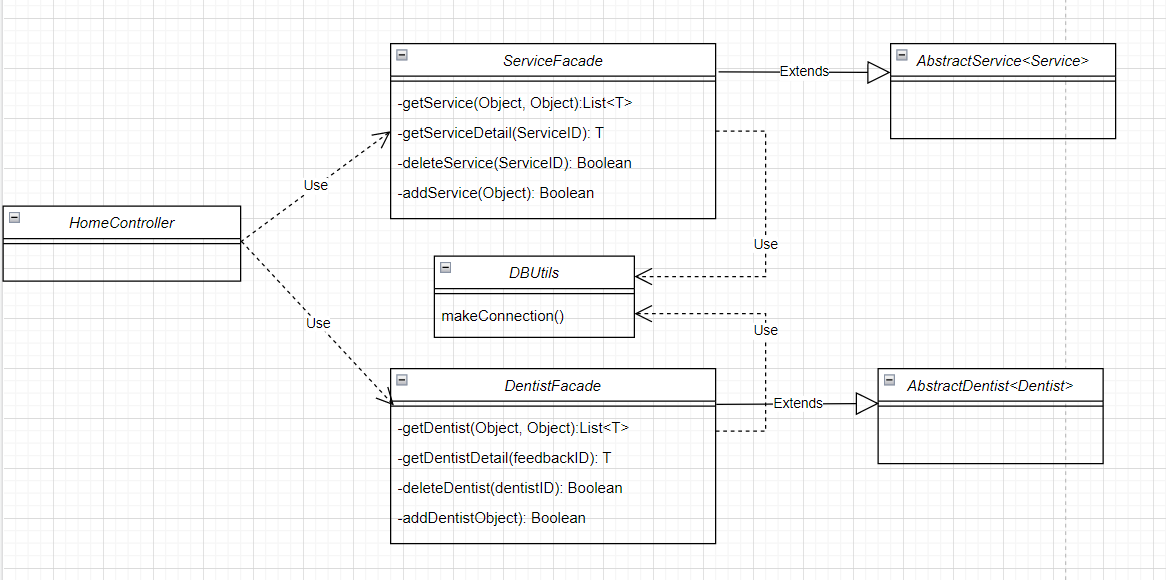
### d. Database queries

getAllAccount(): List<Account>:

+””

## 45. Overview Information/ Search Information (Thiếu controller, bảng)

### a. Class Diagram

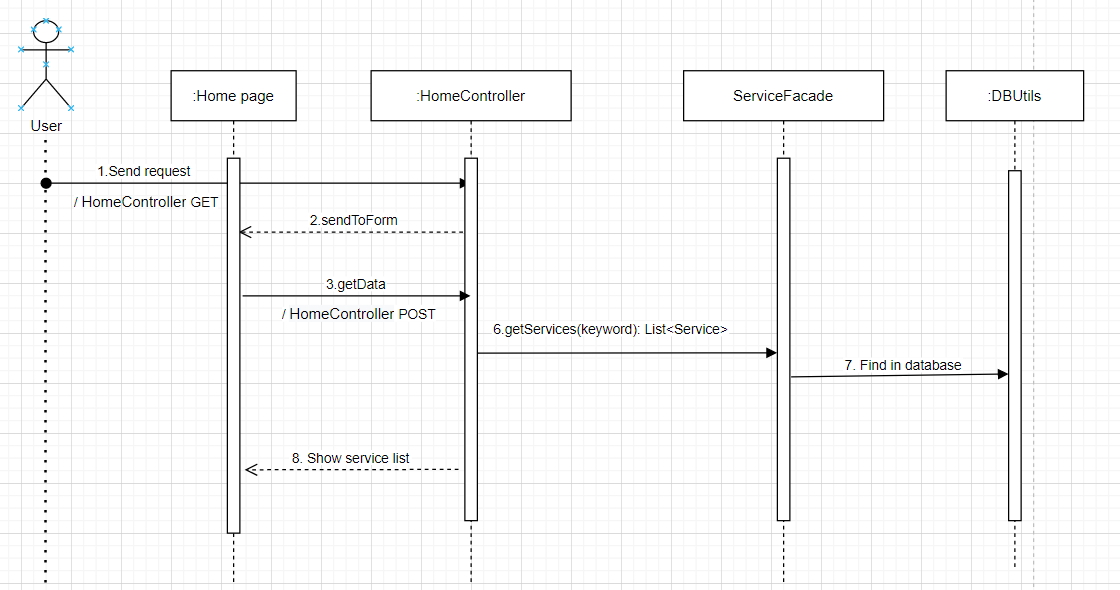


### b. Class Specifications

#### ServiceFacace

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getInfoServicesFromSQL* | *get service, input (ResultSet resultSet), output (Notification)* |
| *02* | *getServices* | *get all service, input (Connection connection, Object object, Object action), output (List<Services>)* |
| *03* | *addServices* | *add Services, input (Connection connection ,Services service), output (boolean)* |
| *04* | *updateServices* | *update Services, input (Connection connection, Services service), output (boolean)* |
| *05* | *countServices* | *count number of Services, input (Connection connection), output (int)* |

### c. Sequence Diagram(s)



### d. Database queries

getAllAccount(): List<Account>:

+””

## 2. <Feature/Function Name1>

*[Provide the detailed design for the function <Feature/Function Name1>. It include Class Diagram, Class Specifications, and Sequence Diagram(s)]*

### a. Class Diagram

*[This part presents the class diagram for the relevant feature]*



### b. Class Specifications

*[Provide the description for each class and the methods in each class, following the table format as below]*

#### XYZ Class

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

#### ABC Class

***Class Methods***

*[Provide the detailed description for the class methods]*

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *<method name>* | *<Description of the method, including the inputs, outputs & internal method processing>* |
|  |  |  |

### c. Sequence Diagram(s)

*[Provide the sequence diagram(s) for the feature, see the sample below]*



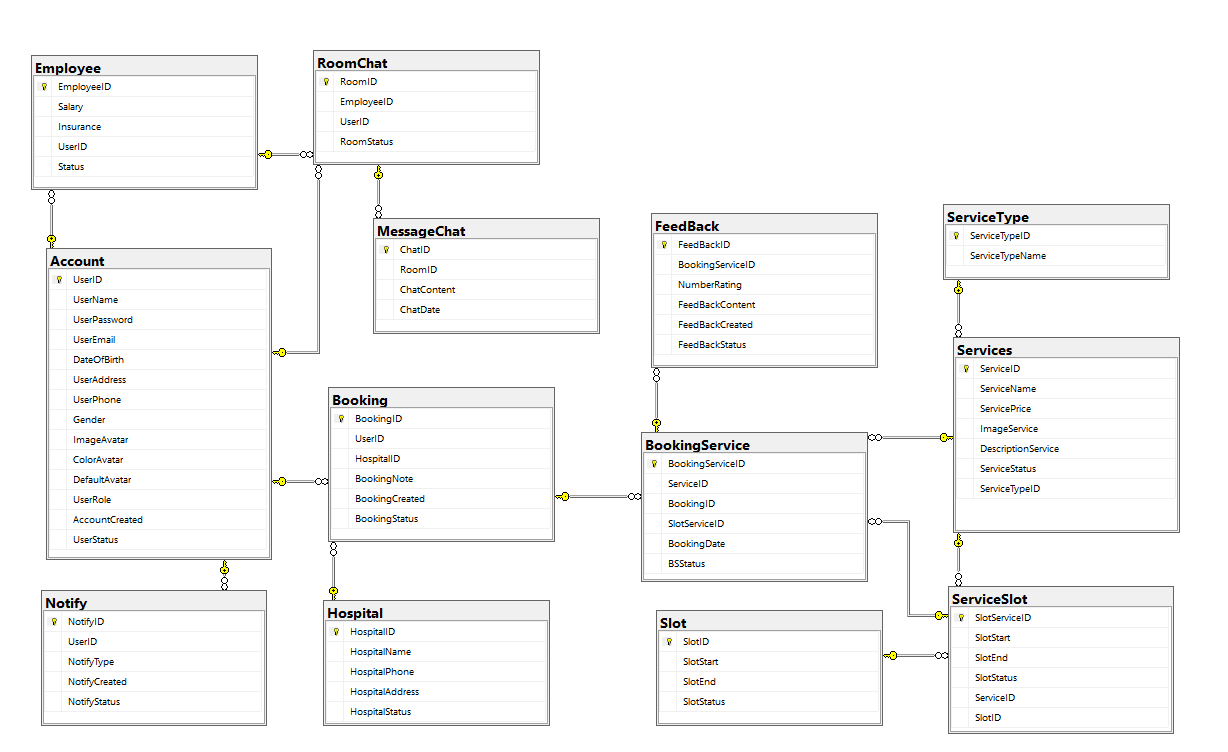
### d. Database queries

*[Provide the detailed SQL (select, insert, update...) which are used in implementing the function/screen]*

## 2. <Feature/Function Name2>

…

# III. Database Tables



## 1. Employee

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | EmployeeID | VARCHAR | 10 |  |  | PK |  |
| 2 | Salary | INT |  |  |  |  |  |
| 3 | Insurance | INT |  |  |  |  |  |
| 4 | UserID | VARCHAR | 10 | x |  | FK |  |

## 2. Account

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | UserID | VARCHAR | 10 |  |  | PK |  |
| 2 | FullName | NVARCHAR | 100 |  | x |  |  |
| 3 | UserPassword | VARCHAR | 50 |  | x |  |  |
| 4 | UserEmail | NVARCHAR | 50 |  | x |  |  |
| 5 | DateOfBirth | DATE |  |  |  |  |  |
| 6 | UserAddress | NVARCHAR | 150 |  |  |  |  |
| 7 | UserPhone | CHAR | 10 |  |  |  |  |
| 8 | Gender | CHAR | 1 |  |  |  |  |
| 9 | ImageAvatar | VARBINARY | MAX |  |  |  |  |
| 10 | ColorAvatar | VARCHAR | 10 |  |  |  |  |
| 11 | DefaultAvatar | CHAR | 1 |  |  |  |  |
| 12 | UserRole | INT |  |  |  |  | DEFAULT '0' |
| 13 | UserStatus | INT |  |  |  |  | DEFAULT '0' |
| 14 | AccountCreated | DATETIME |  |  |  |  | DEFAULT CURRENT\_TIMESTAMP |

## 3. RoomChat

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | RoomID | VARCHAR | 10 |  |  | PK |  |
| 2 | EmployeeID | VARCHAR | 10 |  |  | FK |  |
| 3 | UserID | VARCHAR | 10 |  |  | FK |  |
| 4 | RoomStatus | INT |  |  |  |  | DEFAULT '0' |

## 4. Feedback

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | FeedBackID | VARCHAR | 10 |  |  | PK |  |
| 2 | BookingServiceID | VARCHAR | 10 |  |  | FK |  |
| 3 | NumberRating | INT |  |  |  |  |  |
| 4 | FeedBackContent | NVARCHAR | Max |  |  |  |  |
| 5 | FeedBackStatus | INT |  |  |  |  | INT DEFAULT '0' |
| 6 | FeedBackCreated | DATETIME |  |  |  |  | DEFAULT CURRENT\_TIMESTAMP |

## 5. Booking

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | BookingID | VARCHAR | 10 |  |  | PK |  |
| 2 | UserID | VARCHAR | 10 |  |  | FK |  |
| 3 | HospitalID | VARCHAR | 10 |  |  | FK |  |
| 4 | BookingNote | NVARCHAR | Max |  |  |  |  |
| 5 | BookingCreated | DATETIME |  |  |  |  | DEFAULT CURRENT\_TIMESTAMP |
| 6 | BookingStatus | INT |  |  |  |  | DEFAULT '0' |

## 6. Notify

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | NotifyID | VARCHAR | 10 |  |  | PK |  |
| 2 | UserID | VARCHAR | 10 |  |  | FK |  |
| 3 | NotifyType | NVARCHAR | 10 |  | x |  |  |
| 4 | NotifyStatus | INT |  |  |  |  | DEFAULT '0' |
| 5 | NotifyCreated | DATETIME |  |  |  |  | DEFAULT CURRENT\_TIMESTAMP |

## 

## 7. MessageChat

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | ChatID | VARCHAR | 10 |  |  | PK |  |
| 2 | RoomID | VARCHAR | 10 |  |  | FK |  |
| 3 | ChatContent | NVARCHAR | MAX |  |  |  |  |
| 4 | ChatDate | DATETIME |  |  |  |  | DEFAULT CURRENT\_TIMESTAMP |

## 8. Hospital

| # | Field name | Type | Size | Unique | Not Null | PK/FK | Notes |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | HospitalID | VARCHAR | 10 |  |  | PK |  |
| 2 | HospitalName | NVARCHAR | 150 |  | x |  |  |
| 3 | HospitalPhone | CHAR | 10 |  | x |  |  |
| 4 | HospitalAddress | NVARCHAR | 150 |  | x |  |  |
| 5 | HospitalStatus | INT |  |  |  |  | DEFAULT '0' |

## 9. BookingService

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | BookingServiceID | VARCHAR | 10 |  |  | PK |  |
| 2 | ServiceID | VARCHAR | 10 |  |  | FK |  |
| 3 | BookingID | VARCHAR | 10 |  |  | FK |  |
| 4 | SlotServiceID | VARCHAR | 10 |  |  | FK |  |
| 5 | BookingDate | DATE |  |  |  |  |  |
| 6 | BSStatus | INT |  |  |  |  | DEFAULT '0' |

## 10. Services

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | ServiceID | VARCHAR | 10 |  |  | PK |  |
| 2 | ServiceName | NVARCHAR | 150 |  | x |  |  |
| 3 | ServicePrice | INT |  |  |  |  | DEFAULT '0' |
| 4 | ImageService | VARBINARY | MAX |  |  |  |  |
| 5 | DescriptionService | NTEXT |  |  |  |  |  |
| 6 | ServiceStatus | INT |  |  |  |  | DEFAULT '0' |
| 7 | ServiceTypeID | VARCHAR | 10 |  |  | FK |  |

## 

## 

## 

## 11. Slot

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | SlotID | VARCHAR | 10 |  |  | PK |  |
| 2 | SlotStart | TIME |  |  |  |  |  |
| 3 | SlotEnd | TIME |  |  |  |  |  |
| 4 | SlotStatus | INT |  |  |  |  | DEFAULT '0' |

## 12. ServiceType

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | ServiceTypeID | VARCHAR | 10 |  |  | PK |  |
| 2 | ServiceTypeName | NVARCHAR | 150 |  | X |  |  |

## 13. ServiceSlot

| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | SlotServiceID | VARCHAR | 10 |  |  | PK |  |
| 2 | SlotStart | TIME |  |  |  |  |  |
| 3 | SlotEnd | TIME |  |  |  |  |  |
| 4 | SSlotStatus | INT |  |  |  |  | DEFAULT '0' |
| 5 | ServiceID | VARCHAR | 10 |  |  | FK |  |
| 6 | SlotID | VARCHAR | 10 |  |  | FK |  |

## 