

**BeautéPanda: Android Application for Beauty Enthusiasts**

**BIT304 FINAL YEAR PROJECT I**

**ITERATION PLAN**

(E1500352) Ayu Chrisniyanti

(E1500367) Lovely

Submitted to the

FACULTY OF COMPUTING AND DIGITAL TECHNOLOGY

(SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY)

In partial fulfilment of the requirements

for the degree of

Bachelor of Information Technology (HONS)

HELP UNIVERSITY

SEPTEMBER - DECEMBER 2018

**Chapter 4: Iterations**

* 1. **Introduction**

In developing of the project, iteration is very important part to manage all the planning of project especially for the project that use Agile software development as the methodology. With the iteration plan, all the requirement that are needed to develop the project have the result as expected. Besides, the iteration in Agile methodology is a repeated interval work items that are fully developed and tested. Also, the iteration is needed to complete the project.

* 1. **Iteration Plan**

In this project, we develop our iteration plan to assure that our project is still on the track with our goal. In accordance with our project goals, we have to do many improvements and some changes in each phase to reach our goals. Therefore, we have to analyse the potential risk in each phase so we can eliminate and avoid the risks that may happen while developing this project and to assure all requirements in every phase is still in accordance with the requirements of this project.

* + 1. **Feasibility Study**

## In feasibility study, we as developers have to determine whether the project is worth to do and to ensure it can actually be executed. To analyze the feasibility of the project, we have done some researches and ask potential customers what they think about the application. This is to make sure the customer is actually interested with our application and this project can be carried out. From this, we can plan the iterations which is suitable to customers’ needs.

## We have evaluated that our project is feasible to be done, as we have the resources to do it and we have learnt the programming language that is used in the project. Furthermore, we have made project schedule and see that this project is feasible from time point-of-view, which means we have enough time to develop this project.

* + 1. **Detailed Analysis**

In detailed analysis phase, we defined between the functional and non – functional requirements that are needed to ensure the stability of application. In this analysis phase, the iteration plan will be conducted if the functional and non- functional requirements cannot fulfil the requirements that are needed in developing the project.

The iteration in this phase cannot be done until the functional and non – functional requirements can meet all the requirements that are needed for this project.

* + 1. **Designed Analysis**

In the designed analysis phase, we illustrate the design of the system, interface design, and the workflow of the application. In this phase includes with high level and expanded use case, class diagram, sequence diagram, database design, and user interface design.

The iteration plan in this phase will be conducted if the design of the system is too complicated, for example the user interface is not friendly or there is something wrong with database. The iteration plan can be done by evaluating and checking all of the system design such as use case, class diagram, sequence diagram, database design, and user interface design until it can meet with all of the requirements that are needed.

* + 1. **Construction**

The main focus of this phase is to make the application that correspond to the detailed design that has been illustrated. If there are some errors or there are functions that do not work properly, the iteration in construction phase will be conducted.

To solve these problems, the developers have to evaluate every prototype code until it works properly and match with the detailed design and detailed analysis that has been illustrated.

* + 1. **Testing**

In this phase, we will check each phase in the system to make sure that the outcome of the system can meet all requirements and match with detailed design and detailed analysis. It also to check the system if there is no error. The iteration in testing phase will be used if there are still errors found in the system.

This phase uses validation and verification to ensure that the system correspond with the requirements that are needed for the project. We use validation to make sure that all requirements are met each other. Verification is also use to check each function.

* + 1. **System Delivery**

The main focus of system delivery phase is to check the system whether it is already complete and meet all the requirements that are needed or not. This phase will be conducted, if there is any additional requirement need to be implemented in the system to make the system quality better.

* + 1. **Operation & Maintenance**

This iteration plan will be used if there are some bugs that will interrupt the user while accessing the application. This iteration is needed to remove the bugs and make sure it will disappear and will not appear anymore in the future so the user will use the application comfortably.

This is the list of all use cases that are needed and it will be used for BeautéPanda application. It includes the schedule of iteration plan. In this Final Year Project 1, we have done the iteration of two use cases such as Register and Login.

|  |  |  |  |
| --- | --- | --- | --- |
| Iteration | Use Cases | Proposed Start Date | Proposed End Date |
| 1 | Register | 26th November 2018 | 30th November 2018 |
| 2 | Login | 3rd December 2018 | 6th December 2018 |
| 3 | Edit Profile | 4th February 2019 | 8th February 2019 |
| 4 | Edit Portfolio | 11th February 2019 | 15th February 2019 |
| 5 | View Portfolio | 11th February 2019 | 15th February 2019 |
| 6 | Search Makeup Artist or Beauty Salon | 18th February 2019 | 22nd February 2019 |
| 7 | Send Message | 25th February 2019 | 1st March 2019 |
| 8 | Reply Message | 25th February 2019 | 1st March 2019 |
| 9 | Make an Appointment | 4th March 2019 | 11th March 2019 |
| 10 | Confirm an Appointment | 4th March 2019 | 11th March 2019 |
| 11 | Give Feedback | 12th March 2019 | 15th March 2019 |
| 12 | Reply Feedback | 12th March 2019 | 15th March 2019 |
| 13 | Give Rating | 18th March 2019 | 22nd March 2019 |
| 14 | View Beauty News | 25th March 2019 | 29th March 2019 |
| 15 | Edit Beauty News | 25th March 2019 | 29th March 2019 |
| 16 | View Announcements | 1st April 2019 | 5th April 2019 |
| 17 | Report Feedback | 1st April 2019 | 5th April 2019 |
| 18 | Edit Announcements | 8th April 2019 | 12th April 2019 |
| 19 | Remove Feedback | 8th April 2019 | 12th April 2019 |
| 20 | Logout | 15th April 2019 | 19th April 2019 |

* 1. **Iteration 1**
     1. **Introduction**

In this Final Year Project 1, we have completed two prototypes of our application called BeautéPanda which are register page and login page. After we finished developing the prototypes, we started to check the prototypes by testing the prototypes. The schedule for this iteration plan began on November 26, 2018 to December 6, 2018 which has certain aims and goals. The aims and goals are follows:

* To ensure the function of register and login page are work properly and without any bugs or error that might be interrupted the user while using the application.
* To ensure the register and login page are working in accordance with detailed analysis
* To ensure the interface of register and login page met with detailed design which are user friendly and easy to use. Therefore, everyone can use this application properly.
  + 1. **Purpose**

Iteration in developing a project is very important phase. Iteration itself contributes in overall development of this project. In feasibility study, iteration contributed to ensure the project met the required resources that are needed to build BeautéPanda application and also to ensure that this application can meet all the user requirements.

Besides, in detailed analysis and detailed design, iteration plan contributed to ensure the analysis and the design of the system can meet the requirements that are needed in developing the project and also to ensure that the user interface of the application is user friendly and easy to use by the user. For the construction, the iteration contributed to ensure that the application that we developed is in accordance with detailed analysis that has been illustrated.

In testing of the system, iteration contributed to ensure that the prototypes of the application can give the great outcomes as expected in detailed design and detailed analysis, also to ensure that the prototypes did not processed errors. While in system delivery, iteration contributed to ensure that the prototypes of the application is complete and resemble the real system.

In operation and maintenance, it contributed in checking the bugs to ensure that there is no interruption in the system. Nevertheless, we have not implemented the system delivery and operation and maintenance stage yet because it stage will be implemented and finished in Final Year Project 2 later.

* + 1. **Context**

This iteration plan can meet with all the requirements of the project that are needed. Also, this iteration plan can speed up the process of developing this project and to ensure that our project is in accordance with the project goal that we expected. The first two use cases that will be developed in this iteration are register and login. The iteration for register use case is we want to make sure that we have done the analysis to ensure that the register page meets all the requirements that are needed and run properly. Besides, for login use case we make sure that can login page can work and run properly without any errors and interference. We also ensure that the overall user interface is user – friendly and user can use the application comfortably.

* + 1. **Schedule of Iteration Workflows**

In implementing this iteration, there are no significant problems that might be interrupted the schedule of our project. Therefore, we can implement the iteration accordance to the project schedule so there is no delay in our schedule for iteration.

|  |  |  |  |
| --- | --- | --- | --- |
| Workflow | Start Date | End Date | Duration (days) |
| **Use Case 1: Register** |  |  |  |
| Analysis | 26th November 2018 | 30th November 2018 | 5 Days |
| Design | 26th November 2018 | 30th November 2018 | 5 Days |
| Implementation | 26th November 2018 | 30th November 2018 | 5 Days |
| Testing | 26th November 2018 | 30th November 2018 | 5 Days |
| **Use Case 2: Login** |  |  |  |
| Analysis | 3rd December 2018 | 6th December 2018 | 4 Days |
| Design | 3rd December 2018 | 6th December 2018 | 4 Days |
| Implementation | 3rd December 2018 | 6th December 2018 | 4 Days |
| Testing | 3rd December 2018 | 6th December 2018 | 4 Days |

* + 1. **Iteration Schedule Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| Task Name | Start | Finish | Assigned To |
| Use Case 1 (Register) | 26th November 2018 | 30th November2018 | Lovely |
| **1.1 Analysis** |  |  |  |
| 1.1.1 Analysing of register requirement | 26th November 2018 | 30th November2018 | Lovely & Niya |
| * 1. **Design** |  |  |  |
| 1.1.2 Designing of the registration form and interface | 26th November 2018 | 30th November2018 | Niya |
| **1.3 Implementation** |  |  |  |
| 1.1.3 Implementation of registration | 26th November 2018 | 30th November2018 | Lovely |
| **1.4 Testing** |  |  |  |
| 1.1.4 Testing register | 26th November 2018 | 30th November2018 | Lovely & Niya |

|  |  |  |  |
| --- | --- | --- | --- |
| Task Name | Start | Finish | Assigned To |
| Use Case 2 (Login) | April 12, 2018 | April 13, 2018 | Niya |
| 1.1 Analysis |  |  |  |
| 1.1.1 Analysis of login requirement | April 12, 2018 | April 13, 2018 | Lovely & Niya |
| 1.2 Design |  |  |  |
| * 1. 2 Design of the login form and interface | April 12, 2018 | April 13, 2018 | Niya |
| 1.3 Implementation |  |  |  |
| 1.1.3 Implementation of login | April 12, 2018 | April 13, 2018 | Lovely |
| 1.4 Testing |  |  |  |
| 1.1.4 Testing login | April 12, 2018 | April 13, 2018 | Lovely & Niya |

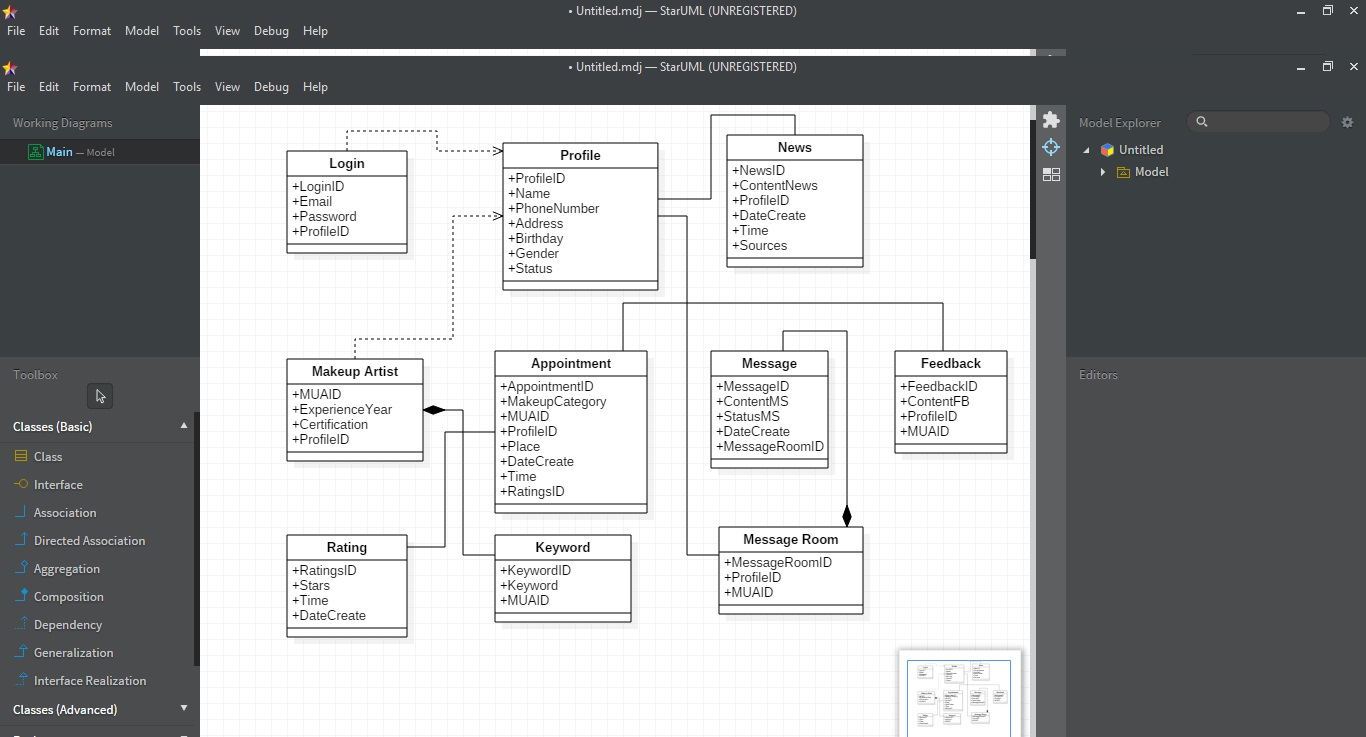
* + 1. **Resource Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Resources | Usage | Quantity | Cost |
| **Hardware** | | | |  |
| 1 | Computer  ASUS laptop X450J  (Windows 10) | Project development of android application and testing with emulator | 1 computer | Rp 0  (Existing computer) |
| 2 | Samsung Galaxy A5  (Android Nougat 7.0) | Project testing to ensure the application can run properly | 1 smartphone | Rp 0  (Existing smartphone) |
| 3 | Computer  Apple MacBook Pro Mid 2012 | Project development of android application and testing with emulator | 1 computer | Rp 0  (Existing computer) |
| **Software** | | | |  |
| 4 | IntelliJ IDEA Community Edition 2018.2.4 | Android application development to develop the front – end and back – end of BeautéPanda | 2 platforms | Rp 0  (Open source) |
| **Human Resources** | | | |  |
| 5 | Team member | Developing and testing the prototype of the system | 2 members | Rp 0 |

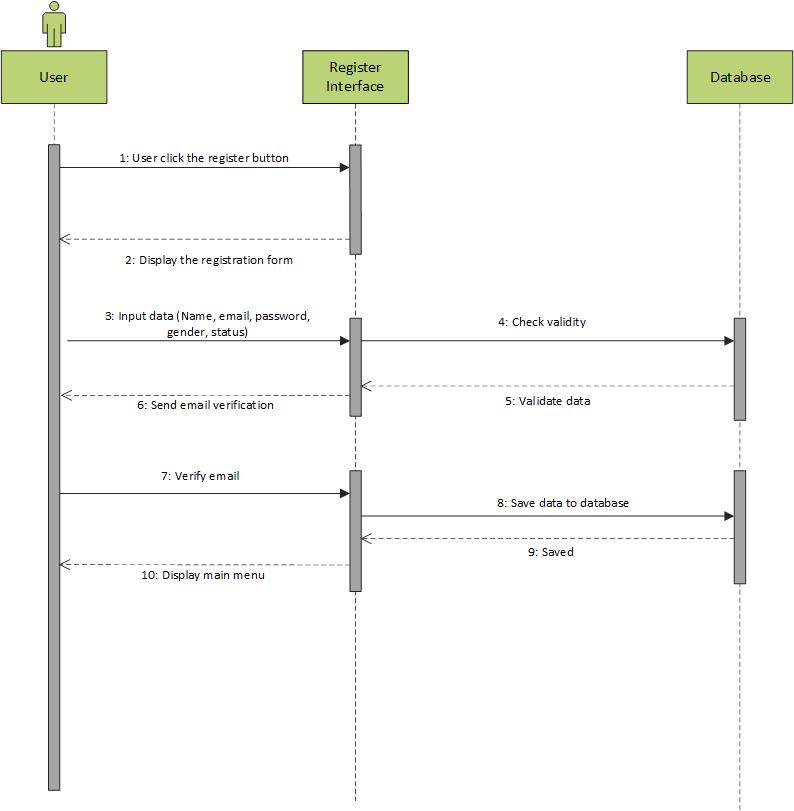
* + 1. **Evaluation Criteria**

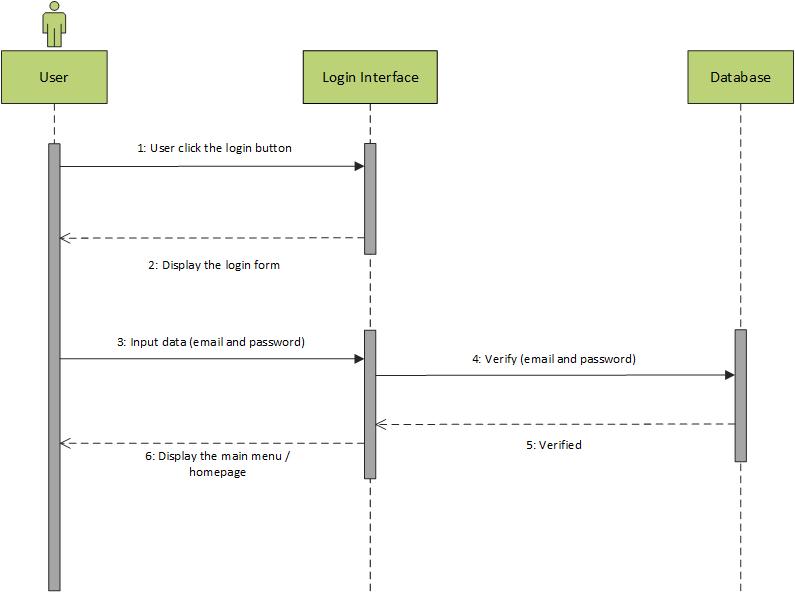
We have successfully finished the iteration plan. We got more understanding with the application that we develop after we did the evaluation of the user requirements for this application. We have done this chapter 4 (Iteration) in time and no delays in our schedule that we made before. We also got more understanding about to use and operate the software that we used during develop the application.

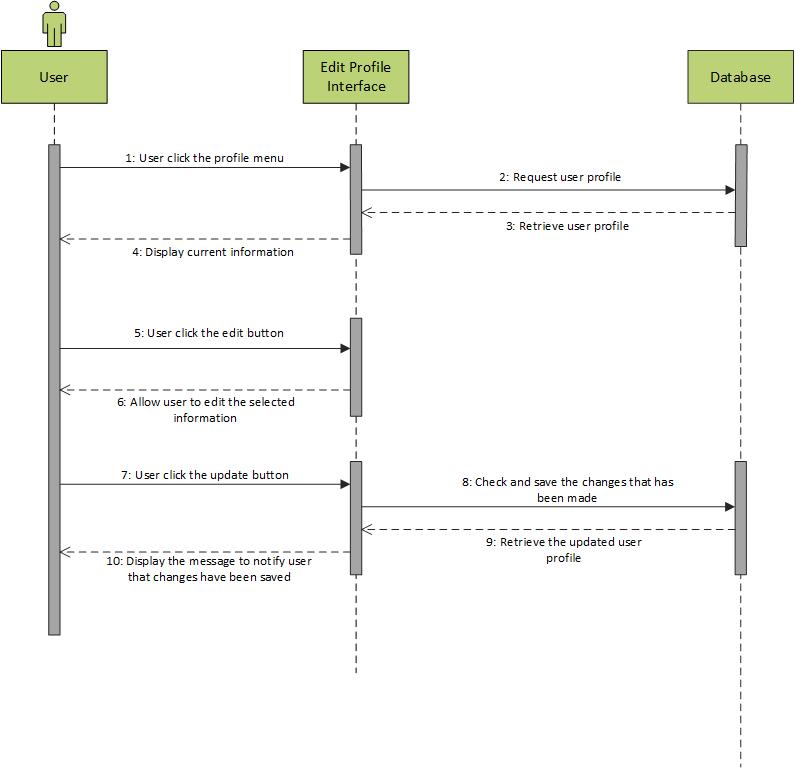
* + 1. **Analysis and Design Artefact**
       1. **Analysis Artefact**
          1. **Analysis Class Diagram**

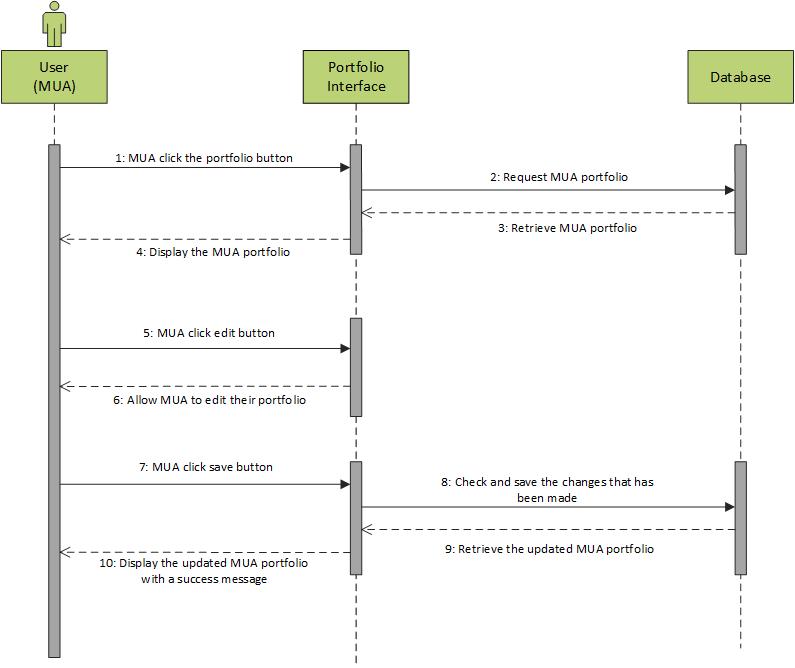
****

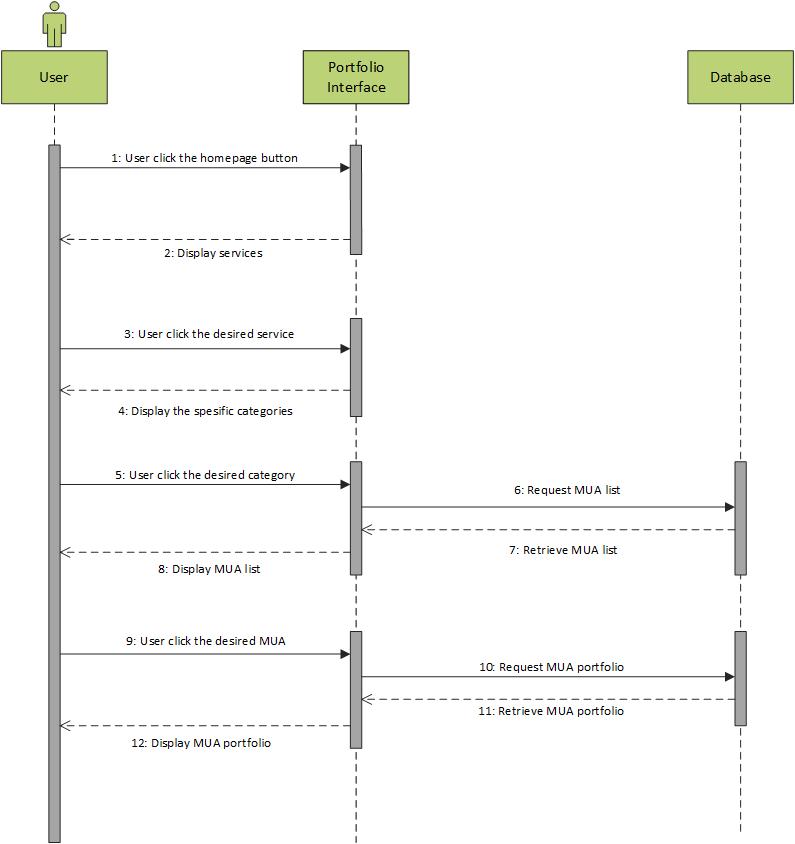
* + - 1. **Design Artefact**
         1. **Sequence Diagram**

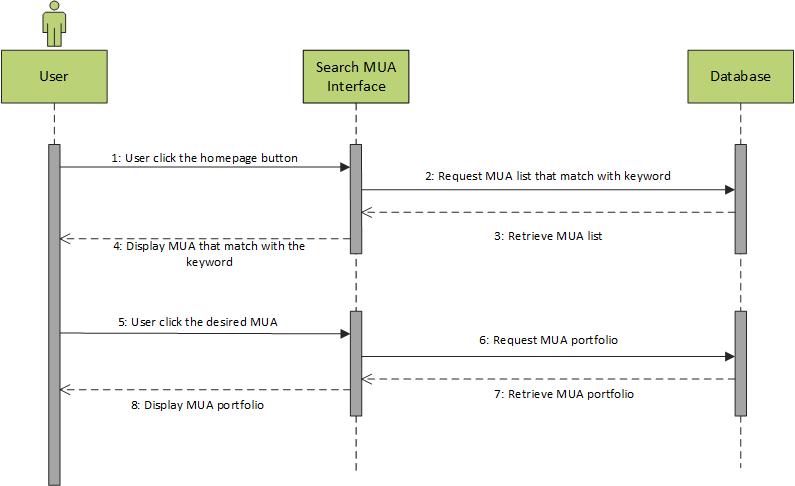
****

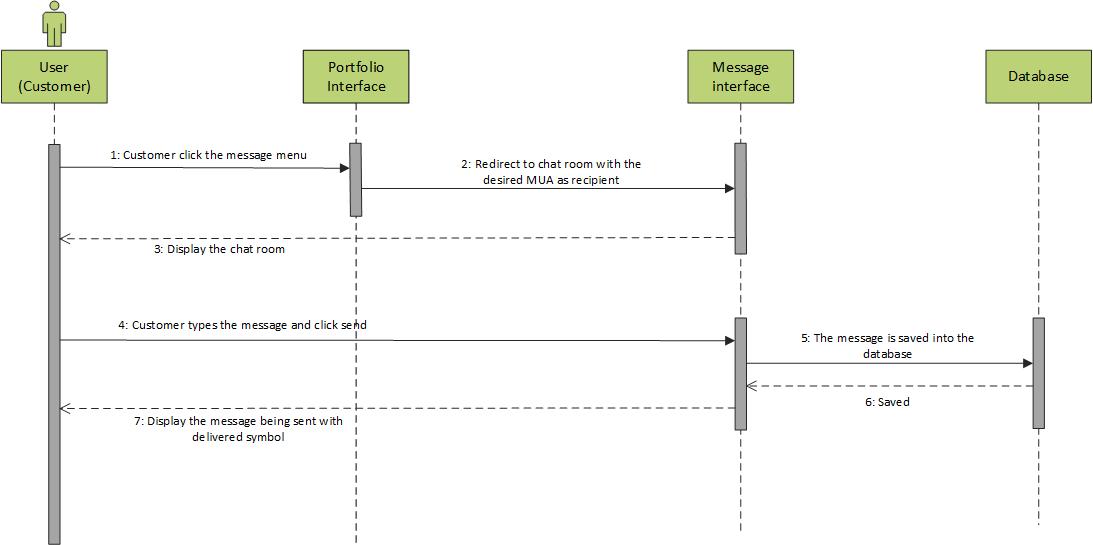


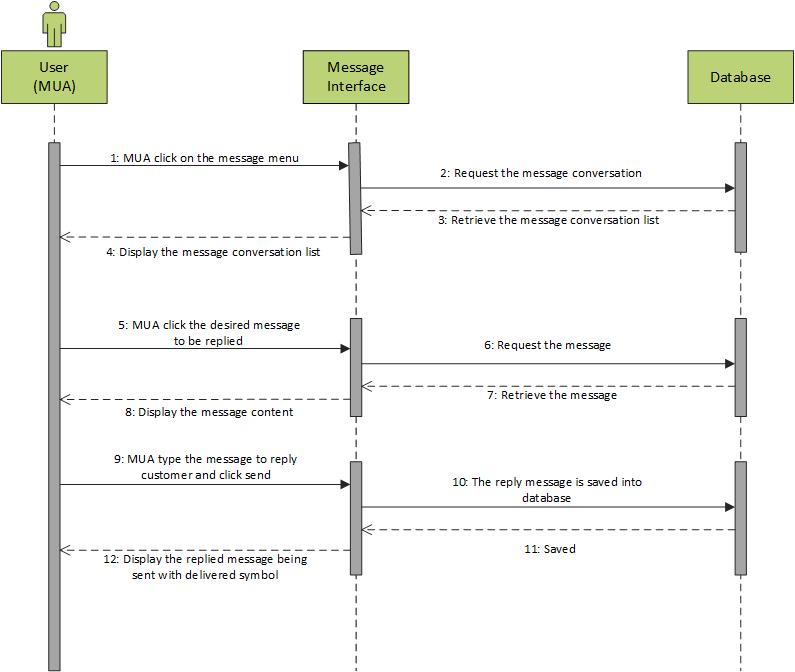


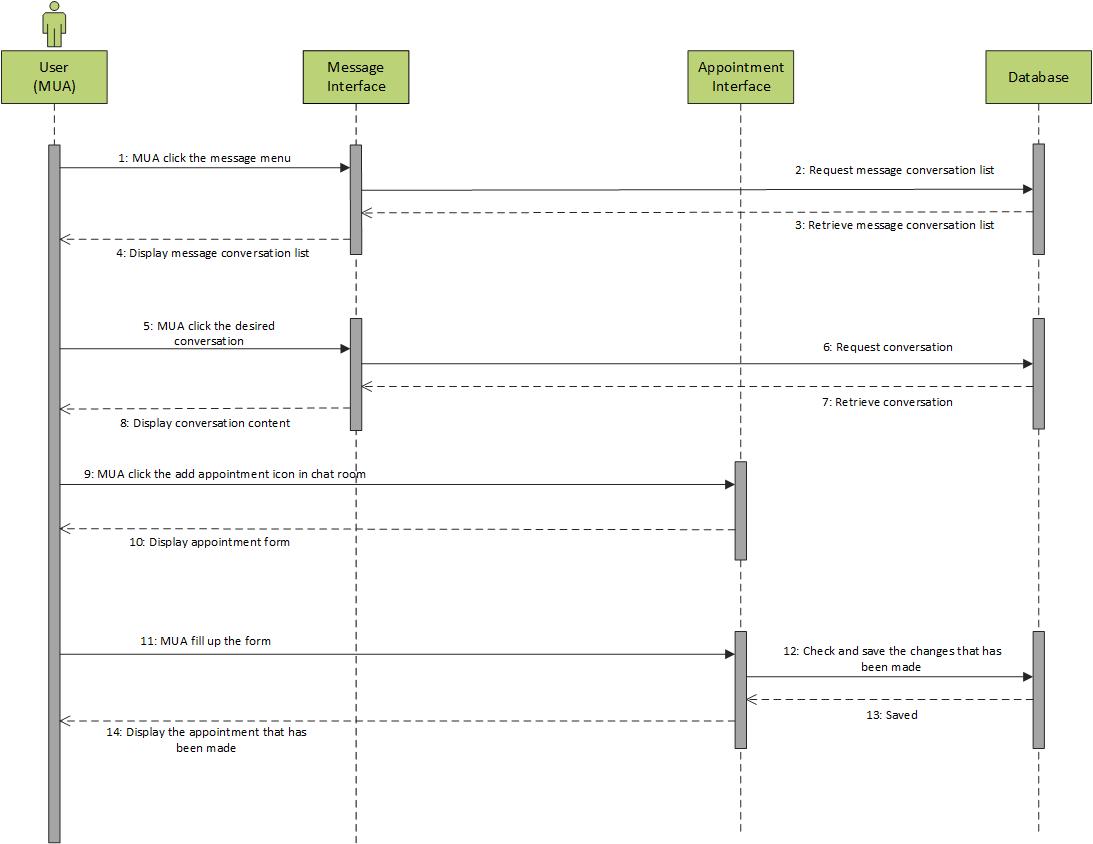


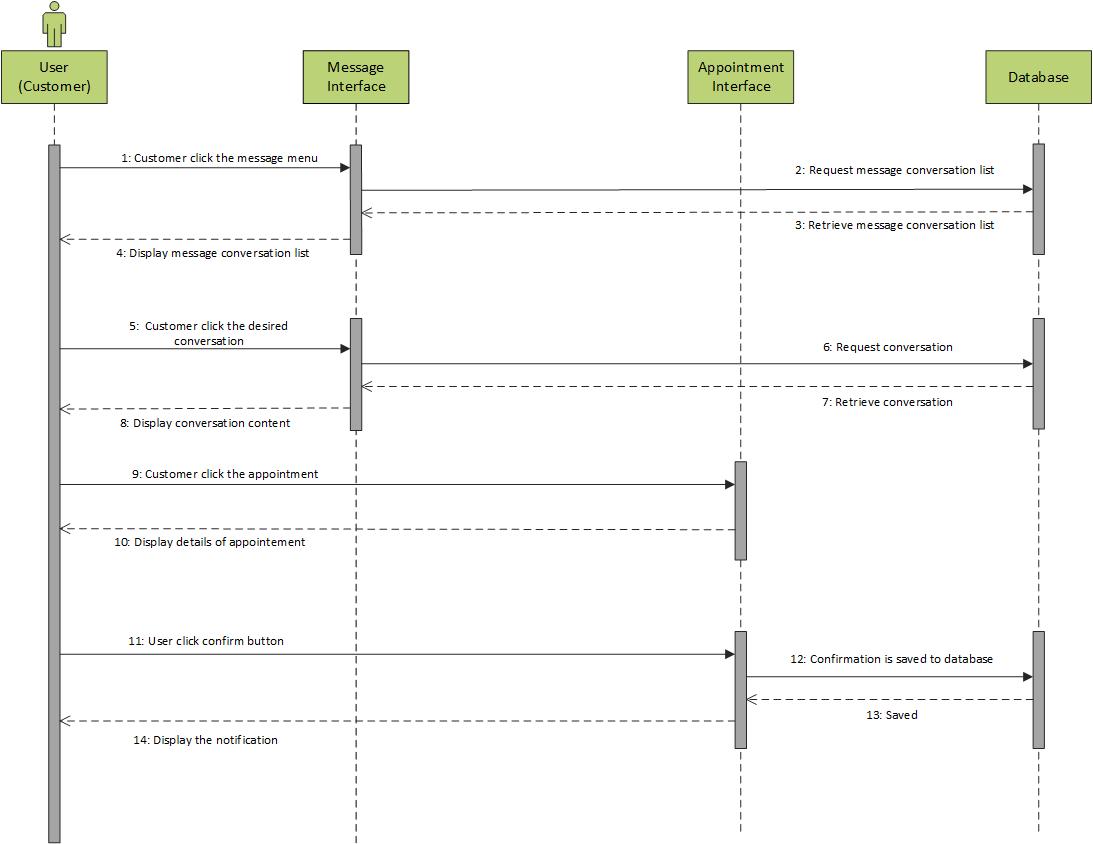


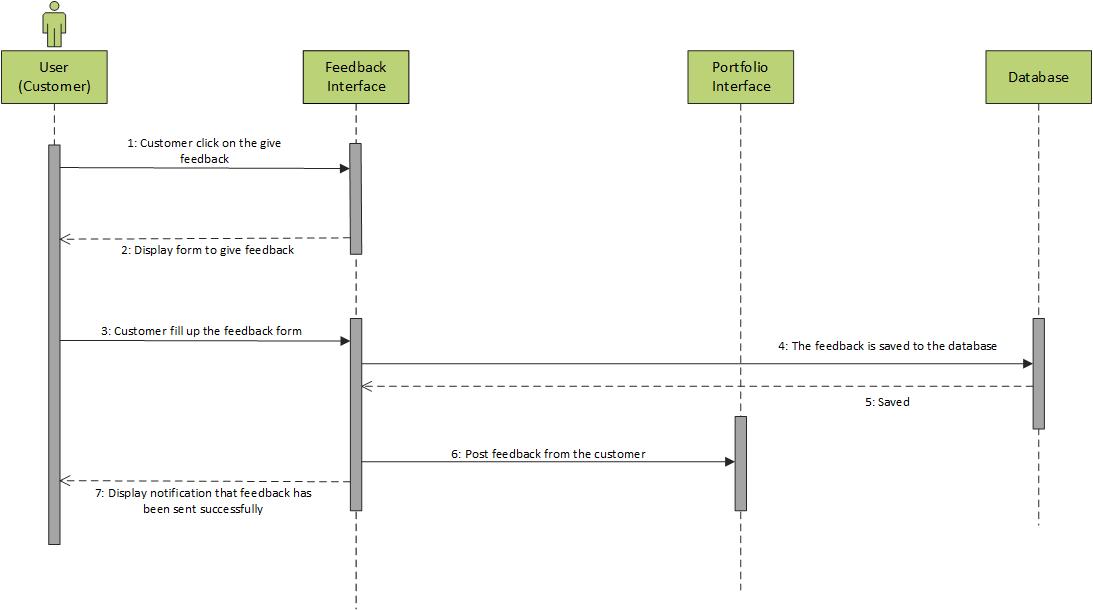


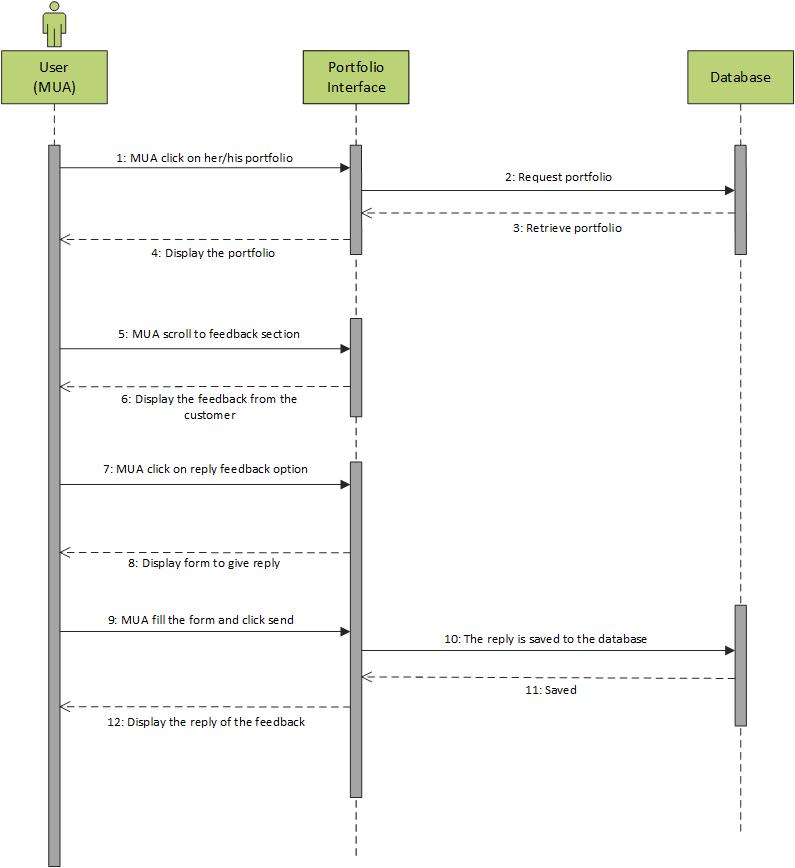


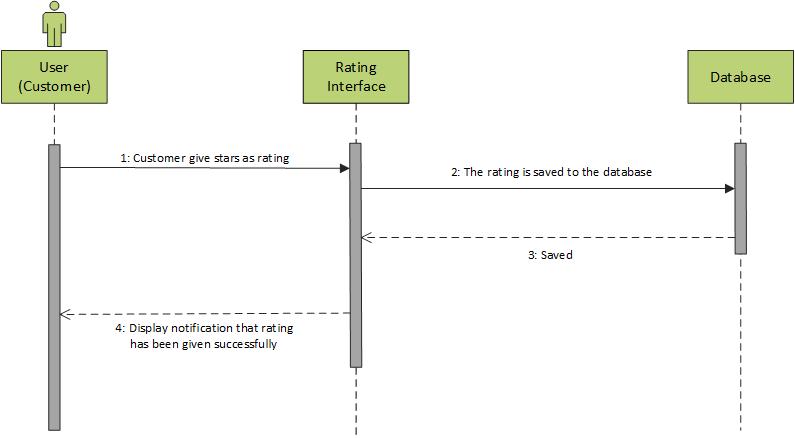


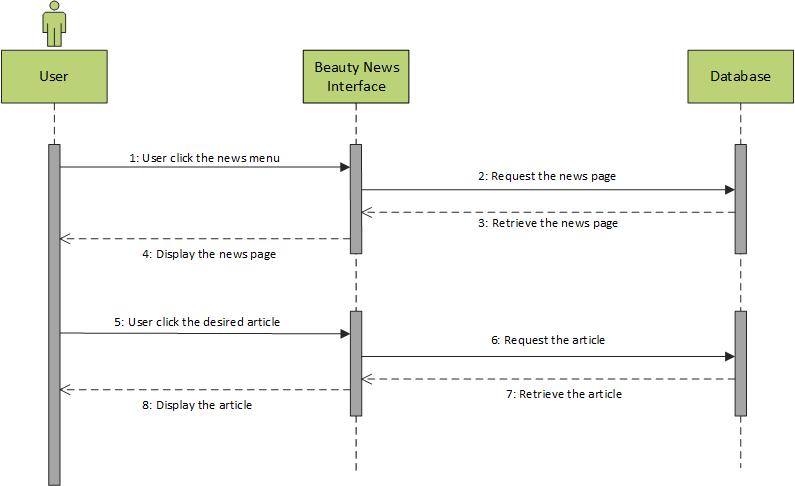


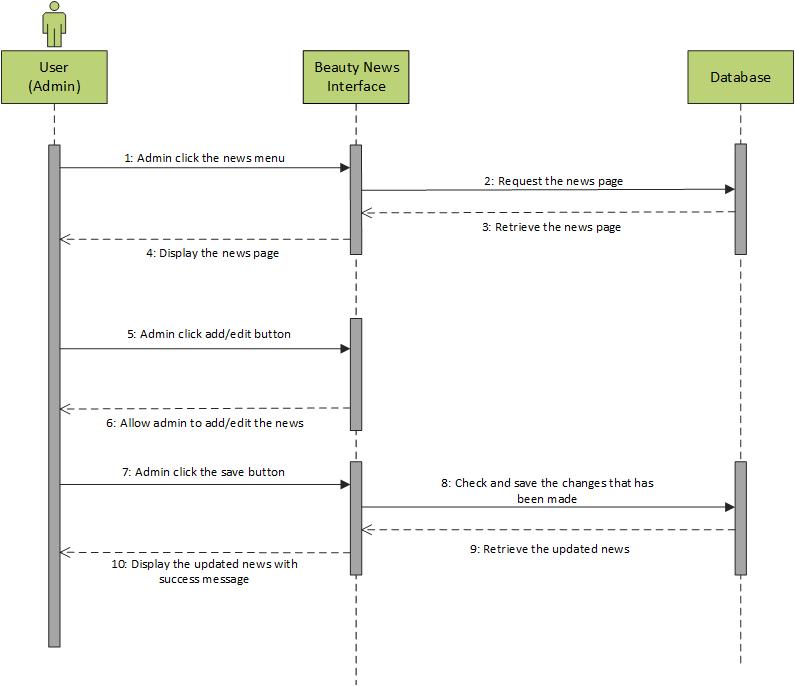


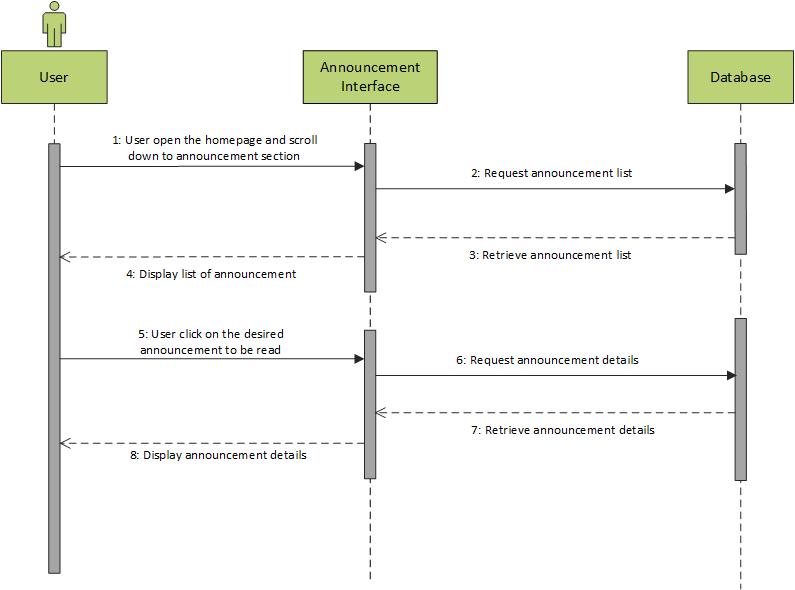
****

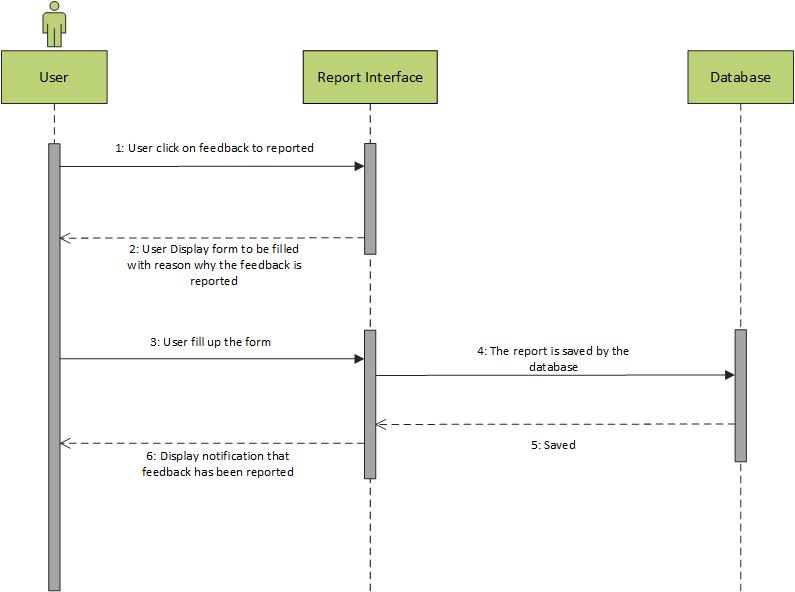
****

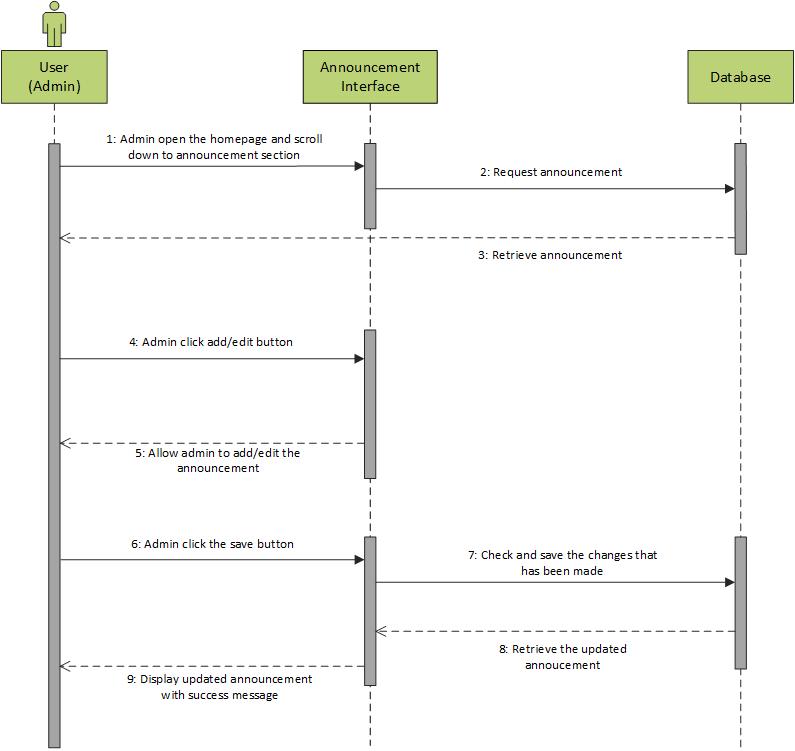
****

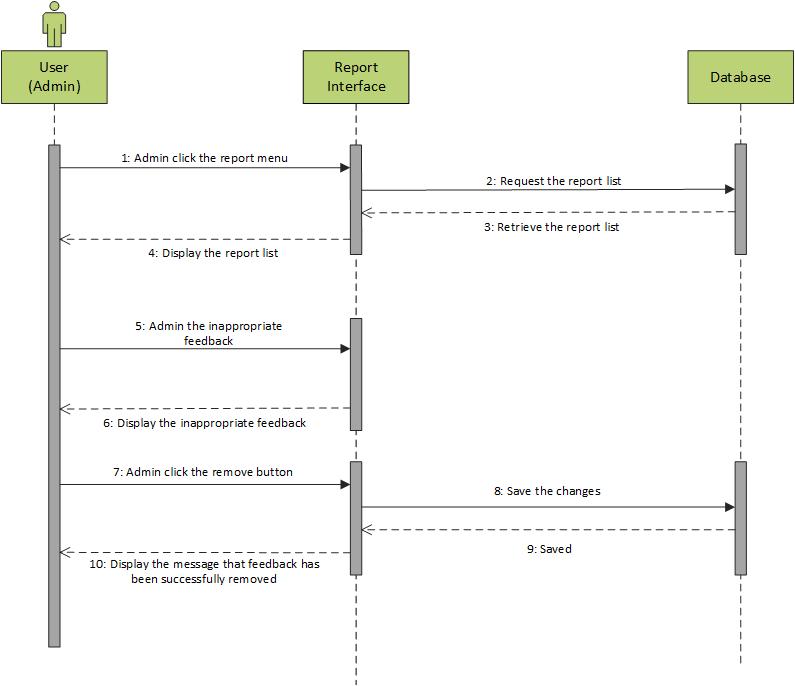
****

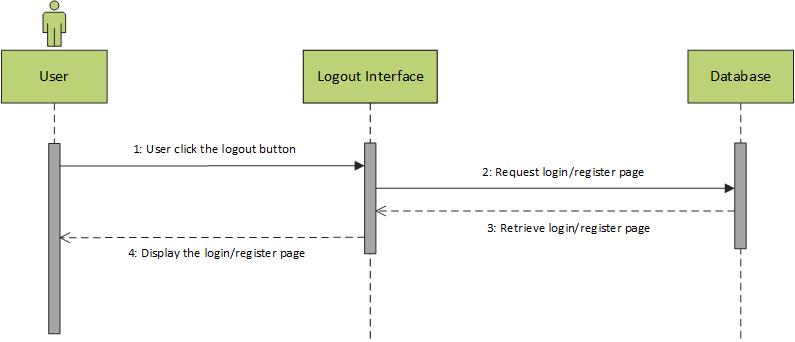


****

****

****

****

****

* + - * 1. **Database Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | LoginID | Int | Yes | 15 |
|  | Email | Varchar | No | 30 |
|  | Password | Varchar | No | 30 |
|  | ProfileID | Int | No | 30 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | KeywordID | Int | Yes | 15 |
|  | Keyword | Varchar | No | 255 |
|  | MUAID | Int | No | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | MUAID | Int | Yes | 15 |
|  | ExperienceYear | Year | No | - |
|  | Certification | Varchar | No | 255 |
|  | ProfileID | Int | No | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | AppointmentID | Int | Yes | 15 |
|  | MakeupCategory | Varchar | No | 255 |
|  | MUAID | Int | No | 15 |
|  | ProfileID | Int | No | 15 |
|  | Place | Varchar | No | 255 |
|  | DateCreate | Date | No | - |
|  | Time | Time | No | - |
|  | RatingID | Int | No | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | ProfileID | Int | Yes | 15 |
|  | Name | Varchar | No | 255 |
|  | PhoneNumber | Varchar | No | 255 |
|  | Address | Varchar | No | 255 |
|  | Status | Int | No | 15 |
|  | Birthday | Date | No | - |
|  | Gender | Int | No | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | FeedbackID | Int | Yes | 15 |
|  | ContentFB | Text | No | - |
|  | ProfileID | Int | No | 15 |
|  | MUAID | Int | No | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | MessageID | Int | Yes | 15 |
|  | ContentMS | Text | No | - |
|  | StatusMS | Varchar | No | 255 |
|  | DateCreate | Date | No | - |
|  | MessageRoomID | Int | No | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | MessageRoomID | Int | Yes | 15 |
|  | ProfileID | Int | No | 15 |
|  | MUAID | Int | No | 15 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | NewsID | Int | Yes | 15 |
|  | ContentNews | Text | No | - |
|  | ProfileID | Int | No | 15 |
|  | DateCreate | Date | No | - |
|  | Time | Time | No | - |
|  | Sources | Varchar | No | 255 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Key | Column Name | Data Type | Allow Null | Length |
| PK | RatingsID | Int | Yes | 15 |
|  | Stars | Int | No | 5 |
|  | Time | Time | No | - |
|  | DateCreate | Date | No | - |

* + - * 1. **Structural Design**

****

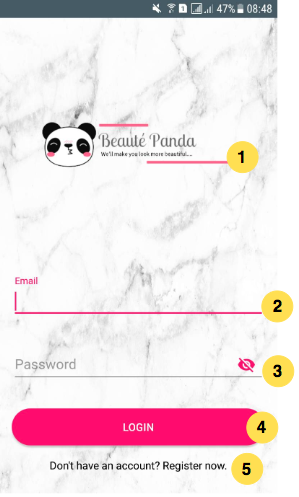
* + - * 1. **User Interface**

**4.3.8.2.4.1 Colour Used**

|  |  |  |
| --- | --- | --- |
| Color | Hex Color Code | Usage |
|  | #aaaaaa | Text, divider line |
|  | #7a7a7a | Notification bar |
|  | #FFFFFF | Background, icon, text |
|  | #FF1470 | Icon background, button, icon, text, header, navigation drawer header |
|  | #000000 | Text |
|  | #BDBDBD | Divider line, text |
|  | #ff4500 | Warning text |

###### 4.3.8.2.4.2 User Interface Design

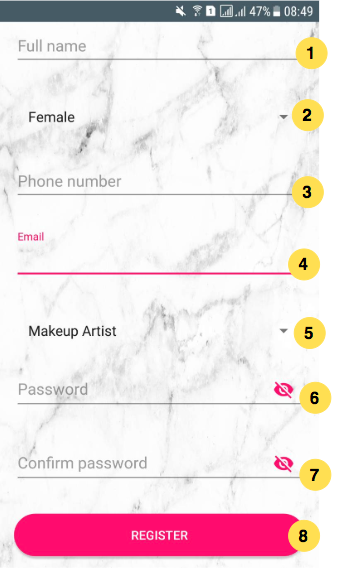
**Login Page**

****

**Explanation:**

1. **Logo:** Logo of the application is placed in the middle
2. **Email Textbox:** User can type their email address that they already registered
3. **Password Textbox:** User can input their password that they already registered and user can hide or show the password while typing.
4. **Login Button:** After user complete the form with correct data, user can tap the login button to go to the main menu of the application
5. **Link Text:** If the user do not have an account they can tap the link that placed under the login button to make an account and complete the form

**Registration Page**

****

**Explanation:**

1. **Full Name Textbox:** User can input their full name in this textbox to complete the registration form
2. **Gender Dropdown:** User can select their gender (Female or Male) by clicking the dropdown menu
3. **Phone Number Textbox:** User can input their phone number in this textbox
4. **Email Textbox:** User can input their email address that they want to use to make an account in this application
5. **Status Dropdown:** User can select their status (Makeup Artist or Customer) by clicking the dropdown menu
6. **Password Textbox:** User can input their password that they want to use and user can hide or show the password while typing.
7. **Confirm Password Textbox:** User have to re – type the password in this textbox to ensure the password is match
8. **Register Button:** After user complete the registration form, user can tap the register button.
   * 1. **Implementation and Testing** 
        1. **Implementation**

In our project, we are using agile methodology to develop this application, and combine it with prototyping. By doing using this methodology, it allow us as the developers to work closely with the customers. It also allows some iterations to be changed later if it is more suitable for the customers’ needs. We are using IntelliJ Idea to create the tests because our application is based on Android. We have learnt Java programming language before so it is familiar for us. Nevertheless, there were many aspects of the testing that we didn’t know and we should learn from tutorials for guidance.

* + - 1. **Testing**

Testing is very important part in developing an application. It needs to be done to ensure the quality of the application. This quality is not just the application can run well, but it also to check the the quality of user interface design and the code of the application. Testing also very important to ensure the application meets all the requirements that has been determined. In this Final Year Project 1, we tested our prototypes in two phases. The use case that we tested are the register and login.

* + - * 1. **Unit Testing**

| **Unit Test Plan** | | |
| --- | --- | --- |
| Module Name/ID: Register | Program Name/ID: BeautéPanda |
| **1. Module Overview** | | |
| Allow user to register for a new account. | | |
|  | | |
| **1.1 Inputs to Module** | | |
| * Full Name (Varchar) * Gender (Varchar) * Phone Number (Varchar) * Email (Varchar) * Status (Varchar) * Password (Varchar) * Confirm Password (Varchar) | | |
|  | | |
| **1.2 Outputs from Module** | | |
| New user has registered successfully and the information is stored in the database. System will show message to notify user that registration is successful. | | |
|  | | |
| **1.3 Logic Flow Diagram/Segment of Code** | | |
| **public class** RegisterActivityTest {   @Rule  **public** ActivityTestRule<RegisterActivity> **registerActivityTestRule** = **new** ActivityTestRule<RegisterActivity>(RegisterActivity.**class**);   @Before  **public void** setUp() **throws** Exception {  }   String **fullName** = **"Test fullname"**;  String **phoneNumber** = **"08123456788"**;  String **email** = **"test@example.com"**;  String **password** = **"password"**;  String **confirmPassword** = **"password"**;  String **wrongConfirmPassword** = **"wrongpassword"**;  String **successString** = **"User created successfully!"**;  String **failString** =**"Password does not match"**;   @Test  **public void** testUserRegister(){  *onView*(*withId*(R.id.***editTextFullName***)).perform(*typeText*(**fullName**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextPhoneNumber***)).perform(*typeText*(**phoneNumber**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextEmail***)).perform(*typeText*(**email**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextPassword***)).perform(*typeText*(**password**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextConfirmPassword***)).perform(*typeText*(**confirmPassword**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***buttonRegister***)).perform(*click*());  *onView*(*withText*(**successString**)).check(*matches*(*isDisplayed*()));  }   @Test  **public void** testUserRegisterFailed(){  *onView*(*withId*(R.id.***editTextFullName***)).perform(*typeText*(**fullName**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextPhoneNumber***)).perform(*typeText*(**phoneNumber**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextEmail***)).perform(*typeText*(**email**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextPassword***)).perform(*typeText*(**password**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextConfirmPassword***)).perform(*typeText*(**wrongConfirmPassword**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***buttonRegister***)).perform(*click*());  *onView*(*withText*(**failString**)).check(*matches*(*isDisplayed*()));  }   @After  **public void** tearDown() **throws** Exception {  } } | | |
|  | | |
| **2. Test Data** | | |
| 1. Valid Data  * Full Name: Test fullname * Gender: Female * Phone Number: 08123456788 * Email: [test@example.com](mailto:test@example.com) * Status: Makeup Artist * Password: password * Confirm Password: password  1. Invalid Data  * Full Name: Test fullname * Gender: Female * Phone Number: 08123456788 * Email: [test@example.com](mailto:test@example.com) * Status: Makeup Artist * Password: password * Confirm Password: wrongpassword (password confirmation does not match) | | |
|  | | |
| **2.1 Positive Test Cases** | | |
| * Full Name: Test fullname * Gender: Female * Phone Number: 08123456788 * Email: [test@example.com](mailto:test@example.com) * Status: Makeup Artist * Password: password * Confirm Password: password   This is an example of valid input data. Data/information inputted in accordance with requested data type. All columns filled, no empty column. This valid data then will be stored in the database and system will display a massage that says “User created successfully”. | | |
|  | | |
| **2.2 Negative Test Cases** | | |
| * Full Name: Test fullname * Gender: Female * Phone Number: 08123456788 * Email: [test@example.com](mailto:test@example.com) * Status: Makeup Artist * Password: password   Confirm Password: wrongpassword (password confirmation does not match)  This is an example of invalid input data. This will happen if the data/information inputted in the column is not complete or does not fit with requested data (i.e., password confirmation does not match). This invalid data will not be stored in the database and system will display a massage that says “Password does not match”. | | |
|  | | |
| **3. Interface Modules** | | |
| 1. Positive Test Case 2. Negative Test Case | | |
|  | | |
| **4. Test Tools** | | |
| This system is tested on smartphones (Samsung Galaxy A5 2016). Android version used in this testing are Android 7.0 Nougat. | | |
| | **Unit Test Plan** | | | --- | --- | | Module Name/ID: Log In | Program Name/ID: BeautéPanda | | **1. Module Overview** | | | Allow registered user to log in | | |  | | | **1.1 Inputs to Module** | | | * Email (Varchar) * Password (Varchar) | | |  | | | **1.2 Outputs from Module** | | | User with registered account can log into the system. After log in, system will display message that log in is successful. | | |  | | | **1.3 Logic Flow Diagram/Segment of Code** | | | **public class** LoginActivityTest {   @Rule  **public** ActivityTestRule<LoginActivity> **loginActivityTestRule** = **new** ActivityTestRule<LoginActivity>(LoginActivity.**class**);   **private** String **email** = **"test@example.com"**;  **private** String **password** = **"password"**;  **private** String **wrongPassword** = **"wrongpassword"**;  **private** String **successString** = **"Successfully Logged in!"**;  **private** String **failString** = **"Failed to log in , please try again"**;   @Before  **public void** setUp() **throws** Exception {  }   @Test  **public void** testUserLogin(){  *onView*(*withId*(R.id.***editTextEmail***)).perform(*typeText*(**email**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextPassword***)).perform(*typeText*(**password**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***buttonLogin***)).perform(*click*());  *onView*(*withText*(**successString**)).check(*matches*(*isDisplayed*()));  }   @Test  **public void** testUserLoginFailed(){  *onView*(*withId*(R.id.***editTextEmail***)).perform(*typeText*(**email**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***editTextPassword***)).perform(*typeText*(**wrongPassword**), *closeSoftKeyboard*());  *onView*(*withId*(R.id.***buttonLogin***)).perform(*click*());  *onView*(*withText*(**failString**)).check(*matches*(*isDisplayed*()));  }   @After  **public void** tearDown() **throws** Exception {  } } | | |  | | | **2. Test Data** | | | 1. Valid Data  * Email: test@example.com * Password: password  1. Invalid Data  * Email: test@example.com * Password: wrongpassword (confirmation password does not match) | | |  | | | **2.1 Positive Test Cases** | | | * Email: test@example.com * Password: password   This is an example of valid input data. All columns filled by user. System will check if email exists in the database and whether the password match. If the email and password match, system will display a massage that says “Successfully Logged in”. | | |  | | | **2.2 Negative Test Cases** | | | * Email: test@example.com * Password: wrongpassword   This is an example of invalid input data. This will happen if the data/information inputted in the column is not complete or incorrect (e.g., password is empty or password does not match with the account data in the database). If the password is wronf and is not found in the database, system will display a message that says “Failed to log in, please try again”. | | |  | | | **3. Interface Modules** | | | 1. Positive Use Case      1. Negative Use Case | | |  | | | **4. Test Tools** | | | This system is tested on smartphones (Samsung Galaxy A5 2016). Android version used in this testing are Android 7.0 Nougat. | | |  | | | | |

* + - * 1. **System Testing**

Testing 1: Register use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Tested:** | | Register | | |
| **Test Description:** | | Testing the registration process for new user. This is to ensure that new user to the system | | |
| **Pre-conditions** | | User should fill all the columns from the registration form displayed by the system. | | |
| **Post-conditions** | | User register successfully and can log in to the system. | | |
| **Notes:** | |  | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | Pass | | |
|  | **TEST STEP** | **EXPECTED TEST RESULTS** | **P** | **F** | |
|  | Register | User clicks the register now text and then the registration page opened | ✓ |  | |
|  | Unregistered user fills all fields from the registration form with valid data | User register a new account successfully to the system. | ✓ |  | |
|  | User does not input valid data or fill all the required data | A warning message will be displayed about which field is incorrectly filled. | ✓ |  | |

Testing 2: Log In use case

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Tested:** | | Log In | | |
| **Test Description:** | | Testing the login by the user. Ensure that user can log into the system | | |
| **Pre-conditions** | | User should have a registered account and fill the login form with valid email address and password. | | |
| **Post-conditions** | | User log in successfully. User enter the system and system displaying the home page. | | |
| **Notes:** | |  | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | Pass | | |
|  | **TEST STEP** | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | Log in | User presses the Log In button and then the login page opened | ✓ |  |
| 2. | User fills the form with valid email address and password | User logs in successfully to the system. | ✓ |  |
| 3. | User inputs invalid email address and password | System will display a warning message about what data the user did not fill correctly or no data inputted to the login form. | ✓ |  |

* + - 1. **Testing Phase Conclusion**

The testing stage is crucial in developing any software. It is to assure developers that functions of the system are properly working. We have done unit testing and system testing. We tested two use cases which are register and login. In unit testing, we test positive and negative case to ensure the behaviour of the system is what we expected. In system testing, we want to find if there is any difference when the use case is tested using certain test step. After the testing have been done, the results are as we expected. Register and log in use case has passed the unit testing and system testing.

* + 1. **Iteration Review and Conclusion**

All the iteration phases are finished as we have planned and it is in accordance with the project schedule. In developing this project, the iteration plan is very helpful in making the project on time and meet all requirements that are needed to develop the application. It also becomes a guidance in order to develop the project properly. Despite having some problems and difficulties while developing this project and testing the system, we managed to do it as we expected.