

REQUIREMENT ANALYSIS DOCUMENT

RAD v01.01



October 31, 2016

OPTIMUM HAIR FINDER

Contents

1 Introduction…………………………………………………..2

* 1. Purpose of the system……………………………………..2
  2. Scope of the system……………………………………….2
  3. Objectives and success criteria of the project……………..2
  4. Definitions, acronyms, and abbreviations………………...2
  5. References………………………………………………...2
  6. Overview………………………………………………….2

1. Current System…………………………………………….2
2. Proposed System…………………………………………..3

3.1

v

# **1 Introduction**

* 1. **Purpose of the system**

It is often difficult to find out where salons are located if you are in an unfamiliar location. It would also be an inconvenience for the user to go shop for shop to ask which hairstyles specific hair salons offer and/or at what price range specific hair styles are on.

Our proposed solution is to come up with a web base system that will help a person locate a salon even in an unfamiliar place without struggles. The user will be able to search hairstyle categories and check hairstyle price range.

* 1. **Scope of the System**
  2. **Core system functionalities**
  3. **Objectives and success criteria of the project**

The success of the application depends upon meeting the following core set of objectives:

* 1. **Definitions, Acronyms, and Abbreviations**

Duration :- the length of time required to for the project.

UCP :- Use Case Point; the measure of the software size.

PF :- Productivity Factor

UUCP :- Unadjusted UCP; The UUCP gives the unadjusted size of the overall system, unadjusted because it does not account for the nonfunctional requirements (TCFs) and the environmental factors (ECFs).

* 1. **References**

"Software Engineering book", Ivan Marsic

# <http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf>

"Share Dialog", Facebook

<https://developers.facebook.com/docs/plugins/share-button>

1. **Current System**
2. **Proposed System**
   1. **Overview**

The proposed system will be web based. The application will allow the user to locate salons in the given location, find out which hairstyles those salons offer and the price ranges.

* 1. **Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Identifier** | **Priority\*** | **System requirement** |
| REQ1 | 5 | The system shall allow the user to create an account and sign in via Facebook, or manually. |
| REQ2 | 5 | If the user isn’t logged in, the system shall allow the user to be able to search for salons by hair category. |
| REQ3 | 5 | The system shall display the pictures of the hairstyles of hair category the user searched for. |
| REQ4 | 5 | If the user is not a registered user, then upon clicking a picture, the system should allow the user to register via either Facebook or manually. |
| REQ5 | 5 | If the user is a registered user, the system should allow the user to search salons by location and hair category. |
| REQ6 | 5 | The system should allow a registered user to book an appointment at the salon of their choice for the hair category they searched for. |
| REQ7 | 2 | The system should allow the user to share a hairstyle on Facebook. |
| REQ8 | 3 | The system should allow salons to update information about their businesses. For example, Salon name, Category, Tags, Address, Contact Information, Website, Rating, Trading hours. |
| REQ9 | 5 | The system should allow the user to view salon location on a map. |
| REQ10 | 5 | The system should allow the user to search by the price range. |
| REQ11 | 3 | The system should allow the user to change password. |

# \*Rating goes from 1 being of lowest priority to 5 being of the highest priority.

* 1. **Non-Functional Requirements**
     1. **Usability**
     2. **Reliability**

### Components of the project code will be tested alongside the implementation phase to ensure that they are functional.

* + 1. **Performance**

The system’s response time will vary, based on the web browser and the user’s network bandwidth. The optimum response time of the system is under two seconds. The web server being utilized for this system, Apache, can host a maximum of 256 concurrent connections, therefore the system will be able to accommodate 256 concurrent connections. The worst latency that the system will experience is 3.7 sec, based on the web server’s bandwidth.

* + 1. **Supportability**

The system will be made into a mobile application, allowing the find-the-shortest-route feature on the system to be more dynamic. The system will be maintained by a team of programmers and software engineer.

* + 1. **Implementation**
    2. **Availability**

Since the system is web based, there is no need for installation. The system will run on any web browser, therefore it is not limited to any operating system or platform.

* + 1. **Hardware**

The system will require any platform that has some sort of connection to the internet. Every function that this system utilizes depends on connecting to the database which the webserver hosts, from searching for hair salons by name to finding the shortest route to the salon.

* + 1. **Software**

For the system to be operational, one must have installed the most recent version of the web browser. Since this is a web-based application, the web browser must be able to accommodate JavaScript code, some functions, such as the log in window, are written in JavaScript.

* + 1. **Security**

Most of our security will be focused of the login and registration of the user. All of the information the use inputs will be encrypted before being sent to our server.

Once the user is registered, the system will allow the user three attempts at logging in. Failing to give a correct password in these three attempts will result in the system asking the user to reset the password by sending a link to the email they entered when registering.

* + 1. **Interface**

Since the system is web based, the system will interact with the user’s web browser and the web server hosting the system. The data between the user’s inputs and the system’s results will be transferred via the network the user is connected to. The system will be able to run on the any version of web browser.

* + 1. **Assumptions / Constrains**

To make use of any of the functions the system offers, the system must have registered the potential user. If the system can’t identify the user then the system must prompt the potential user to register.

* 1. **System Models**
     1. **Stakeholders**

**Client:** They would like to use the system to know where they can find the nearest salons and the salons of their choice.

* + 1. **Actors and Goals**

|  |  |  |
| --- | --- | --- |
| **ACTOR** | **ACTOR’S GOALS** | **USE CASE NAME** |
| Client | To create an Account. | UC-1 Sign Up |
| Client | To Sign In to the system. | UC-2 Sign In |
| Client | To change password. | UC-3 Change Password |
| Client | To search for salons by location. | UC-4 Search By Location |
| Client | To search for salons by hair category | UC-5 Search By Hair Category |
| Client | To be able to see where the salon is situated on Google maps and how far is it from the client’s location. | UC-6 View Salon On Map |
| Client | To be able to get route from where the client is to the salon on Google maps. | UC-7 Get Salon Route |
| Client | To be able to get salon address on Google maps salon position. | UC-8 Get Salon Address |
| Client | To book an appointment with the salon. | UC-9 Book |
| Client | To search for salons by price range. | UC-10 Price Range |
| Client | To be able to view hairstyle pictures. | UC-11 Photographs |
| Client | To be able to like and share a hairstyle on Facebook | UC-12 Share |

* + 1. **Use Cases**
       1. **Casual Description**

**UC-1 Sign Up**

The client want to sign up and create an account so that she can have access to some other functionalities of the system.

**UC-2 Sign In**

The client signs in to the system.

**UC-3 Change Password**

The clients can change their password.

**UC-4 Search By Location**

The client can search for salons by their location.

**UC-5 Search By Category**

The client can search for salons by the treatment they offer.

**UC-6 View Salon On Map**

The client can view salon on the Google Map and get the distance from the clients location to the salon.

**UC-7 Get Salon Route**

The client can get the route from its current location to the salon on Google maps.

**UC-8 Get Salon Address**

The client willbe able to get salon address on Google maps salon position.

**UC-9 Booking**

The client can be able to book an appointment with the salon.

**UC-10 Price Range**

The client can be able to search for a salon by the price range.

**UC-11 Photographs**

The client can view the hairstyle pictures.

**UC-12 Share**

The client can like and share a post on Facebook.

* + - 1. **Use Case Diagram**

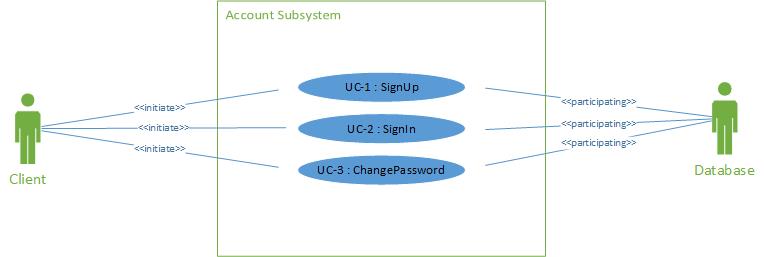
****

Figure 1 Use Case Diagram for Account Subsystem

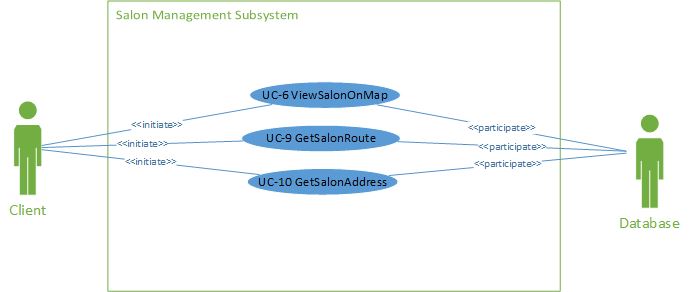
****

Figure 1 Use Case Diagram for Salon Management Subsystem

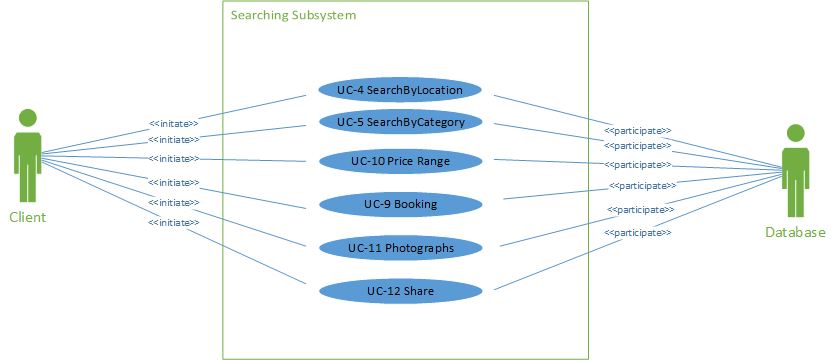
****

Figure 1 Use Case Diagram for Searches Subsystem

**3.4.3.3 Traceability Matrix**

The Traceability Matrix allows the reader to cross the functional and non-functional requirements described earlier with the use cases. This demonstrates which use cases fulfill each requirement, and the total priority weight of each use case will determine which cases are the most important. If an X is present at any point in the column for a Use Case, then the corresponding requirement’s priority weight must be added to the sum. The remaining Xs in the column are similarly considered, and the total priority weight for the Use Case is listed at the bottom of the column

.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Prior**  **ity**  **Weig-ht** | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** | **UC11** | **UC12** |
| **REQ1** | 5 | X | X |  |  |  |  |  |  |  |  |  |  |
| **REQ2** | 5 |  |  |  |  | X |  |  |  |  |  |  |  |
| **REQ3** | 5 |  |  |  |  |  |  |  |  |  |  | X |  |
| **REQ4** | 5 | X |  |  |  |  |  |  |  |  |  |  |  |
| **REQ5** | 5 |  |  |  | X | X |  |  |  |  |  |  |  |
| **REQ6** | 5 |  |  |  |  |  |  |  |  | X |  |  |  |
| **REQ7** | 2 |  |  |  |  |  |  |  |  |  |  |  | X |
| **REQ8** | 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| **REQ9** | 5 |  |  |  |  |  | X | X | X |  |  |  |  |
| **REQ**  **10** | 5 |  |  |  |  |  | X |  |  |  | X |  |  |
| **REQ**  **11** | 3 |  |  | X |  |  |  |  |  |  |  |  |  |
| **TOTAL WEIGHT** |  | 10 | 5 | 3 | 5 | 5 | 10 | 5 | 5 | 5 | 5 | 5 | 2 |

**Table 1 Traceability Matrix**

**3.4.3.4 Fully Dressed Descriptions**

|  |  |
| --- | --- |
| **Use Case UC-1** | **Sign Up** |
| Related Requirements: | Req1 |
| Initiating Actor: | Client |
| Actor’s Goal: | To create an account |
| Participating Actors: | Database |
| Preconditions: | None worth mentioning |
| Post conditions: | A new client is added into the database. |
| Flow of Events for Main Success Scenario:  1. Client goes to home page, clicks on the hyperlink “Sign Up”.  2. System prompts the client to enter their details.  3. The client enters required details and then submits.  4. The System records the new data into the database and signals completion. | |

Table 3.4.3.3.1 UC-1

|  |  |
| --- | --- |
| **Use Case UC-2** | **Sign In** |
| Related Requirements: | Req1 |
| Initiating Actor: | Client |
| Actor’s Goal: | To sign in to the system |
| Participating Actors: | Database |
| Preconditions: | None worth mentioning |
| Post conditions: | A Client is signed in to the system. |
| Flow of Events for Main Success Scenario:  1. Client goes to home page, clicks on the hyperlink “Sign in”.  2. System prompts the client to enter their details.  3. The client enters required details and then submits.  4. System prepares a database query to verify whether the details provided are valid. | |

Table 3.4.3.3.2 UC-2

|  |  |
| --- | --- |
| **Use Case UC-3** | **Change Password** |
| Related Requirements: | Req11 |
| Initiating Actor: | Client |
| Actor’s Goal: | To change password |
| Participating Actors: | Database |
| Preconditions: | None worth mentioning |
| Postconditions: | System successfully updates the client’s password. |
| Flow of Events for Main Success Scenario:  1. The client clicks on sign in hyperlink.  2. The system displays the sign in form.  3. The client clicks on “Forgot Password” hyperlink.  4. System prepares a database query to verify whether the email address provided is in the database. If the email address is found in the database, then an email will be sent to the client with a link that when clicked will direct the client to a page where they can register a new password.  5. System prepares another database query to update the password.  6. Client will then login using the email address and the updated password. | |

Table 3.4.3.3.3 UC-3

|  |  |
| --- | --- |
| **Use Case UC-4** | **Search By Location** |
| Related Requirements: | Req10 |
| Initiating Actor: | Any Client |
| Actor’s Goal: | To search for salons by Location |
| Participating Actors: | Google Maps |
| Preconditions: | • The sets of salons and their locations are stored in the system database.  • The system displays the menu of available functions at the home page of the system, “search by hairstyle”, “search by location”, “search by price range” and “search by name”. |
| Postconditions: | A page with all salons in the location searched will be generated |
| Flow of Events for Main Success Scenario:  1. Client goes to home page search drop down menu, chooses option “search by Location”.  2. System allows client to enter the location.  3. System prepares a database query to search the database and returns all the salons in the location searched.  4. Clients can then select the salons of their choice. | |

Table 3.4.3.3.5 UC-4

|  |  |
| --- | --- |
| **Use Case UC-5** | **Search By Category** |
| Related Requirements: | Req2, Req5 |
| Initiating Actor: | Any Client |
| Actor’s Goal: | To search for salons by Treatment |
| Participating Actors: | Database |
| Preconditions: | The user clicks on a dropdown menu and selects a category to filter the results and clicks on the search button. |
| Postconditions: | The system returns pictures of the categories selected. |
| Flow of Events for Main Success Scenario:  1. System allows client to selects a hairstyle category and search thereby.  2. System prepares a database query to search the database and returns all the salons which offer the category searched for.  3. Clients can then select the salons of their choice with the category searched for. | |

Table 3.4.3.3.6 UC-5

|  |  |
| --- | --- |
| **Use Case UC-6** | **View Salon On Map** |
| Related Requirements: | Req9, Req10 |
| Initiating Actor: | Any client |
| Actor’s Goal: | To be able to see where salon is situated on Google maps. |
| Participating Actors: | Google Maps |
| Preconditions: | • The set of salons and IP addresses stored in the system database.  • The system displays available options at the salon page, one of which is view salon on map. |
| Postconditions: | Google map shows position of salon. |
| Flow of Events for Main Success Scenario:   1. Client goes to home page and searches for the hair category of their choice. 2. The system will then provide another search option where the user has to select a location of the salon convenient for them. 3. The user will then select the location. 4. The system will then prepares a database query to retrieve all the salons in the user’s desired location that offers the selected hair category. 5. Then the system will compile a list of all the retrieved salons with two clickable buttons for each salon; the view salon on map and the view salon gallery buttons. | |

Table 3.4.3.3.7 UC-6

|  |  |
| --- | --- |
| **Use Case UC-7** | **Get Salon Route** |
| Related Requirements: | Req9 |
| Initiating Actor: | Any client |
| Actor’s Goal: | To be able to get route from where the client is to the salon on Google maps. |
| Participating Actors: | Google Maps |
| Preconditions: | • The set of salons and IP addresses stored in the system database.  • The system displays available options at the salon page, one of which is get route. |
| Postconditions: | Google maps calculates route from current position to salon position and shows route on map. |
| Flow of Events for Main Success Scenario:   1. The system allows the user to search for salon whether by hair category, price range, and/or location.   2. System complies a list of salons as a result of the search.  3. Upon selecting a salon of your choice, there will be a button where by the user can get the salon route. | |

Table 3.4.3.3.8 UC-7

|  |  |
| --- | --- |
| **Use Case UC-8** | **Get Salon Address** |
| Related Requirements: | Req9 |
| Initiating Actor: | Client |
| Actor’s Goal: | To be able to get salon address on Google maps salon position. |
| Participating Actors: | Google Maps |
| Preconditions: | • The set of salons and IP addresses stored in the system database.  • The system displays available options at the salon page, one of which is view salon on map. |
| Postconditions: | Google map shows position of salon. |
| Flow of Events for Main Success Scenario:   1. Client goes to home page and searches for the hair category of their choice. 2. The system will then provide another search option where the user has to select a location of the salon convenient for them. 3. The user will then select the location. 4. The system will then prepares a database query to retrieve all the salons in the user’s desired location that offers the selected hair category. 5. Then the system will compile a list of all the retrieved salons with the salon’s location details. | |

Table 3.4.3.3.9 UC-8

|  |  |
| --- | --- |
| **Use Case UC-9** | **Booking** |
| Related Requirements: | Req6 |
| Initiating Actor: | Client |
| Actor’s Goal: | To book an appointment with a salon. |
| Participating Actors: | Database |
| Preconditions: | None worth mentioning |
| Post conditions: | A new salon’s details are added into the database. |
| Flow of Events for Main Success Scenario:  1. Client goes to booking page.  2. System prompts the client to fill in the booking form.  3. The client enters required details and then submits.  4. The System would then send an email to the salon with the booking details and send a copy to the user. | |

Table 3.4.3.3.10 UC-9

|  |  |
| --- | --- |
| Use Case UC-10 | Search By Price |
| Related Requirements: | Req10 |
| Initiating Actor: | Any Client |
| Actor’s Goal: | To Search for salons that can accommodate for the selected hairstyle chosen. |
| Participating Actors: | Database |
| Preconditions: | The system displays three drop down list. |
| Postconditions: | A list of salons should appear |
| Flow of Events for Main Success Scenario:   1. User selects one of the his/her preferred hairstyles 2. User then selects the minimum price she/he is willing to pay for the service 3. User then selects the maximum price she/he is willing to pay for the service 4. System then takes the selection given 5. Via SQL, the system will find the list of hair salons that meet the criteria | |

Table 3.4.3.3.11 UC-10

|  |  |
| --- | --- |
| **Use Case UC-12** | **Photographs** |
| Related Requirements: | Req3 |
| Initiating Actor: | Any Client |
| Actor’s Goal: | To be able to view hairstyle photographs. |
| Participating Actors: | Database |
| Preconditions: |  |
| Postconditions: |  |
| Flow of Events for Main Success Scenario:  1. Client goes to gallery page.  2. System prepares a database query to search the database and returns all the hairstyle pictures.  4. Clients can then select the picture to view it. | |

Table 3.4.3.3.12 UC-11

|  |  |
| --- | --- |
| **Use Case UC-11** | **Share** |
| Related Requirements: | Req7 |
| Initiating Actor: | Any Client |
| Actor’s Goal: | To be able to like and share posts on Facebook |
| Participating Actors: | Database |
| Preconditions: |  |
| Postconditions: |  |
| Flow of Events for Main Success Scenario:  1. Client goes to any share button and click it.  2. System prompts a Facebook menu to choose whether to share publicly or to the personal timeline. | |

Table 3.4.3.3.13 UC-12

**3.4.3.5 System Sequence Diagrams**

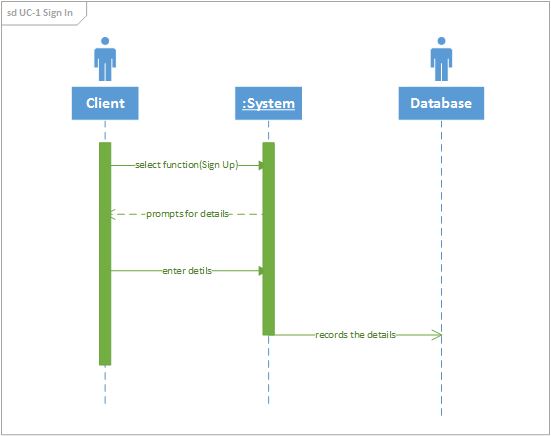


Figure 1 Sequence Diagram of Use Case 1

Figure 4-1 shows the sequence diagram of Use Case 1 which is “Sign Up”. It is used to create new client’s accounts. First the client clicks the “Sign Up” hyperlink. Next, the system prompts the client to enter their details. Then the client enters the required details. After that the system records the data into the database.

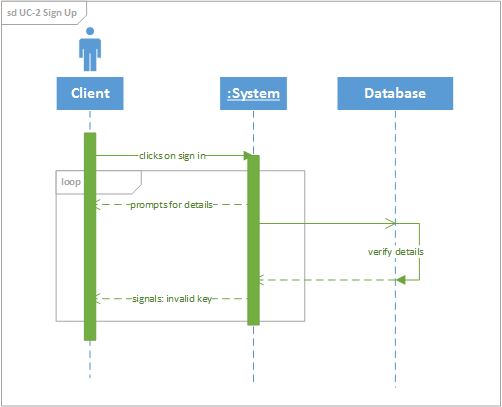


Figure 2 Sequence Diagram for Use Case 2

Figure 2 shows the sequence diagram of Use Case 2 which is “Sign In”. It is used to sign in clients. First the client clicks the “Sign In” hyperlink. Next, the system prompts the client to enter their details. Then the client enters the required details. After that the System prepares a database query to verify whether the details provided are valid.

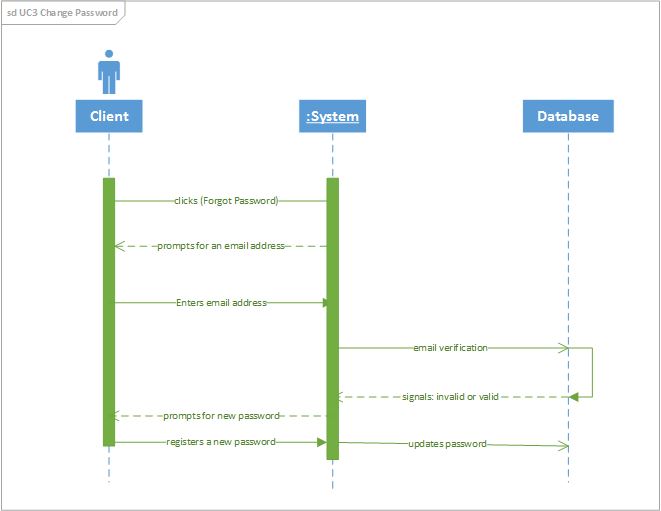


Figure 3 Sequence Diagram for Use Case 3

Figure 3 shows the sequence diagram of Use Case 3 which is “Sign In”. It is used to change a client’s password. First, the clients clicks the “Forgot Password” hyperlink and the system will prompt the client to enter a registered password. The system prepares a database query to verify whether the email address provided is in the database. If the email address is found in the database, then an email will be sent to the client with a link that when clicked will direct the client to a page where they can register a new password. Then after that the system will prepare another database query to update the old password.

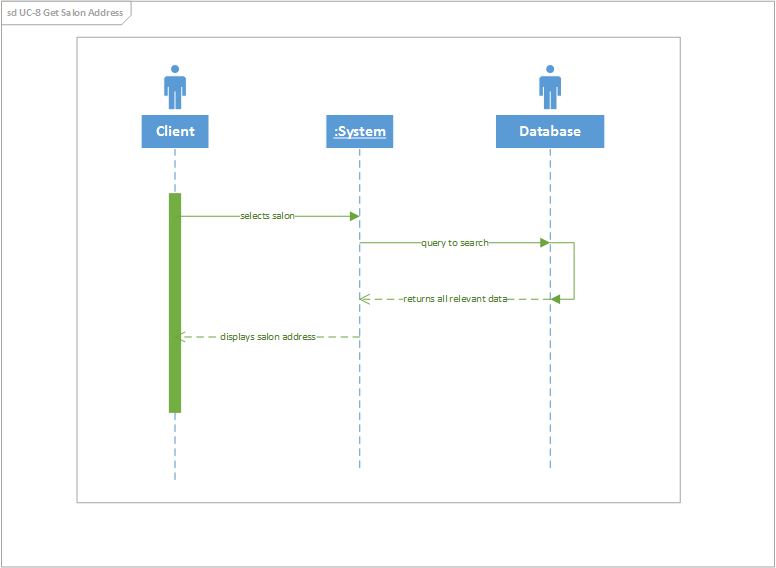


Figure 4 Sequence Diagram for Use Case 8

Figure 4 shows the sequence diagram of Use Case 8 which is “Get Salon Address”. It is used to find a salon’s address. There are many ways a client can get this stage of finding a salon’s address; regardless of which search parameter the user will find a button that will link the salon and its address. Once a user has selected a salon from a list generated by search parameters there will be an address button on the salon’s page.

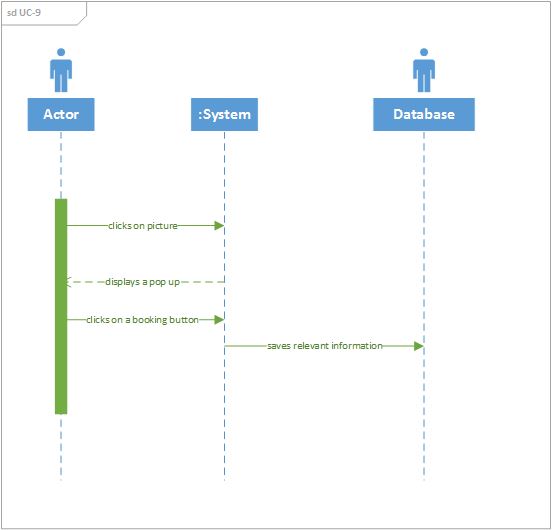


Figure 5 Sequence Diagram for Use Case 9

Figure 5 shows the sequence diagram of Use Case 9 which is “Book”. It is used to book an appointment with the salon if there is a hair style that you like. Upon clicking an picture a pop up block will be invoked. The bocks has a booking button and the Facebook share a hairstyle.

* + 1. **Effort Estimation**

**Standard Equations:**

Duration = UCP + PF

UCP = UUCP\*TCF\*ECF

UUCP = UAW + UUCW

The formula for calculating UCP is composed of three variables:

1. Unadjusted Use Case Points (UUCP), which measures the complexity of the functional requirements.

2. The Technical Complexity Factor (TCF), which measures the complexity of the nonfunctional requirements

3. The Environment Complexity Factor (ECF), which assesses the development team’s experience and their development environment.

UCP = UUCP × TCF × ECF

|  |  |  |
| --- | --- | --- |
| Actor Type | Description of how to recognize the actor type | Weight |
| Simple | The actor is another system which interacts with our system. | 1 |
| Average | The actor is a person interacting through a text-based user interface. | 2 |
| Complex | The actor is a database interacting through SQL | 3 |

Table 3.4-1 Actor classification and associated weights

Unadjusted Use Case Points (UUCPs) are computed as a sum of these two components:

1. The Unadjusted Actor Weight (UAW), based on the combined complexity of all the actors in all the use cases.

2. The *Unadjusted Use Case Weight (UUCW)*, based on the total number of activities (or steps) contained in all the use case scenarios.

|  |  |  |  |
| --- | --- | --- | --- |
| Actor Name | Description of relevant characteristics | Complexity | Weight |
| Not Logged in user | Not Logged In users interacts through a user interface. | Simple | 1 |
| Logged In user | Logged In users interacts through a user interface. | Average | 2 |
| Database | This is another system that interacts with our system. | Complex | 3 |

Table 3.4-2 Actor classification for optimum hair finder

UAW=3 × Complex + 2 × Average + 1 × Simple= 3 × 1 + 2 × 1 + 1 × 1 = 6

|  |  |  |
| --- | --- | --- |
| Use Case Category | Description of how to recognize the use case category | Weight |
| Simple | Simple user interface. Up to one participating actor (plus initiating actor). Number of steps for the success scenario: ≤3. If presently available, its domain model includes ≤3 concepts. | 5 |
| Average | Moderate interface design. Two or more participating  actors. Number of steps for the success scenario: 4 to 7. If presently available, its domain model includes between 5 and 10 concepts. | 10 |
| Complex | Complex user interface or processing. Three or more  participating actors. Number of steps for the success  scenario: 7. If available, its domain model includes≥10 concepts. | 15 |

Table 3.4-3 Use case weights based on the number of transactions

The UUCW is calculated by tallying the use cases in each category, multiplying each count by its specified weighting factor (Table 4.3-3), and then adding the products.

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case | Description | Category | Weight |
| UC-1 | Simple user interface. 4 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |
| UC-2 | Simple user interface. 4 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |
| UC-3 | Complex user interface. 6 steps for the main success scenario. Two participating actors (User, Database). | Average | 10 |
| UC-4 | Simple user interface. 4 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |
| UC-5 | Simple user interface. 3 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |
| UC-6 | Complex user interface. 5steps for the main success scenario. Two participating actors (User, Database). | Average | 10 |
| UC-7 | Simple user interface. 3 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |
| UC-8 | Complex user interface. 6 steps for the main success scenario. Two participating actors (User, Database). | Average | 10 |
| UC-9 | Simple user interface. 4 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |
| UC-10 | Complex user interface. 6 steps for the main success scenario. Two participating actors (User, Database). | Average | 10 |
| UC-11 | Simple user interface. 3 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |
| UC-12 | Simple user interface. 2 steps for the main success scenario. 2 participating actors. (User, Database). | Simple | 5 |

Table 3.4-4 Use case classification for optimum hair finder

UUCW=3 × Simple + 2 × Average + 4 × Complex = 8 × 5 + 3 × 10 + 0 × 15= 70

UUCP= UAW + UUCW = 6 + 70 = 76

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Technical  Factor | Description | Weight | Perceived  Complexity | Calculated  Factor |
| T1 | Distributed, Web-based system | 2 | 3 | 6 |
| T2 | Users expect good performance but nothing exceptional | 1 | 3 | 3 |
| T3 | End-user expects efficiency but there are  no exceptional demands | 1 | 3 | 3 |
| T4 | Ease of use is very important | 0.5 | 5 | 2.5 |
| T5 | No portability concerns beyond a desire  to keep database vendor options open | 2 | 2 | 4 |
| T6 | Easy to change minimally required | 1 | 1 | 1 |
| T7 | Security is a significant concern | 1 | 5 | 5 |
| T8 | No direct access for third parties | 1 | 0 | 0 |
| T9 | No unique training needs | 1 | 0 | 0 |
| Technical Factor Total | | | | 24.5 |

Table 3.4-5: Technical complexity factors for the Optimum hair finder

Constant-1 (*C*1) = 0.6

Constant-2 (*C*2) = 0.01

*Wi* = weight of *i*th technical factor.

*Fi* = perceived complexity of *i*th technical factor.

TCF = 0.6 + (0.01 × 24.5) = 0.845

According to equation (UCP = UUCP\*TCF\*ECF), this results in a reduction of the UCP by 8.5%.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Environmental  Factor | Description | Weight | Perceived  Complexity | Calculated  Factor |
| E1 | Beginner familiarity with the UML based development | 1.5 | 1 | 1.5 |
| E2 | Some familiarity with application  problem | 0.5 | 2 | 1 |
| E3 | Some knowledge of object-oriented  approach | 1 | 2 | 2 |
| E4 | Highly motivated, but some team  members occasionally slacking | 1 | 4 | 4 |
| E5 | Stable requirements expected | 2 | 5 | 10 |
| E6 | No part-time staff will be involved | -1 | 0 | 0 |
| E7 | Programming language of average difficulty will be used | -1 | 3 | -3 |
| Environmental Factor Total | | | | 15.5 |

Table 3.4-6: Environmental complexity factors and their weights.

Constant-1 (*C*1) = 1.4

Constant-2 (*C*2) = -0.03.

Therefore we calculate the ECF = C1 + C2 × EFT

ECF = 1.4 + (-0.03 × 15.5) = 0.935

So we calculate the final UCP:

UCP = 76 × 0.845 × 0.935 = 60.0457 Use Case Points.

The Use case points (UCP) are a measure of software size. When we set Productivity Factor PF=28 per use case point, we can get our project Duration=UCP×PF, where PF=28 hours per use case point.

Duration=UCP×PF=60×28=1680 hours.

**4. Domain Analysis**

**4.1 Domain Models**

**4.1.1 Concept Definitions**

|  |  |  |
| --- | --- | --- |
| **Responsibility Description** | **Type** | **Concept Name** |
| Coordinate actions of all concepts associated with a use case, a logical grouping of use cases, or the entire system and delegate the work to other concepts. | D | Controller |
| Container for user’s authentication data, such as username, password, email address, etc. | K | Key |
| Verify whether or not the credentials entered by the user are valid. | D | Checker |
| Prepare a database query to add new registered clients | D | Database connection |
| Sign in the client into the system to have access to more functionality | D | Sign In |
| Verify and register a new password into the database | D | Password Changer |
| Drop down specifying the search parameters. | K | Search Requests |
| Prepare a database query that best matches the actor’s search  criteria and retrieve the records from the database | D | Database Queries |
| Reads the data that the user types in | K | Text Reader |
| Prepare a database query to update salon’s business information | D | Database connection |
| Get user’s current location and the distance from the salon | D | Get Address |
| Display a route from user’s current location to the salon | K | Route |
| Display salons on map | K | Map |
| Create appointments with salons | D | Booking |
| Display salon’s hairstyle pictures | K | Interface |
| Share and like salons hairstyle picture on Facebook | D | Share |

Table 4.1.1 Concepts Definition

**4.1.2 Association Definitions**

|  |  |  |
| --- | --- | --- |
| **Concept Pair** | **Association Description** | **Association Name** |
| **Controller <->Checker** | Controller calls checker to check if information is valid and Checker returns results to Controller or checker requests to use database and controller returns results | **Generate requests** |
| **Text Reader <-> Controller** | Text Reader sends signals to the Controller or the Controller sends signals to the Text Reader | **Requests Generator** |
| **Checker <-> Key** | Checks if the key is valid or not | **Verifier** |
| **Controller<->DB Connection** | Controller generates requests | **Render Requests** |
| **Text Reader <-> Booking** | Text Reader sends data into the booking form and the booking form sends signals to the Text Reader | **Convey data** |
| **Controller<->Interface** | Controller renders its results and generate requests to Interface to display | **Generate requests** |
| **Controller<->DB Connection** | Controller generates requests | **Render requests** |
| **Controller <->Map** | Controller generates request to view the salons on the Map. | **Generate requests** |
| **Controller <->Route** | Controller generates request to view the salon’s route. | **Generate requests** |
| **Password Changer <->**  **Database Queries** | Password Changer sends a request to the database to change the password. | **Password Update** |
| **Search Requests <->**  **Database Queries** | Search Requests sends a search parameter to the database and the database prepares a query and returns the results. | **Search** |
| **Address <->**  **Database Queries** | Address sends a database query to retrieve a salon’s address. | **Address finder** |

Table 4.1.2 Association Definition

**4.1.3 Attribute Definitions**

|  |  |  |
| --- | --- | --- |
| **Concept** | **Attribute** | **Attribute Description** |
| **Text Reader** | **Read Data** | Reads the data that the user types in |
| **Controller** | **Information Receiver** | Coordinate actions of all concepts associated with a use case, a logical grouping of use cases, or the entire system and delegate the work to other concepts. |
| **Key** | **Credentials** | Container for user’s authentication data, such as username, password, email address, etc. |
| **Interface** | **Show Route**  **Show Map**  **Show Pictures** | Display a route from user’s current location to the salon.  Display salons on map  Display salon’s hairstyle pictures |
| **DB connection** | **Add new Client**  **Retrieve Data**  **Update Data** | Prepare a database query to add new registered clients  Prepare a database query that best matches the actor’s search criteria and retrieve the records from the database  Prepare a database query to update salon’s business information |
| **Share** | **Share** | Share salons hairstyle picture on Facebook |
| **Search Requests** | **Search** | Drop down specifying the search parameters. |

**Table 4.1.3 Attribute Definition**

**4.1.4 Traceability Matrix**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **UC6** | **UC7** | **UC8** | **UC9** | **UC10** | **UC11** | **UC12** |
| **Text Reader** | X | X | X |  | X |  |  |  | X |  |  |  |
| Controller | X | X | X | X | X | X | X | X | X |  | X | X |
| DB Connection | X |  | X |  | X |  | X |  | X |  | X |  |
| Password Changer |  |  | X |  |  |  |  |  |  |  |  |  |
| Deactivator |  |  |  |  |  |  |  |  |  |  |  |  |
| Get Address |  |  |  | X |  |  |  | X |  |  |  |  |
| Booking |  |  |  |  |  |  |  |  | X |  |  |  |
| Info |  |  |  |  |  |  |  |  |  |  |  |  |
| Database Queries | X |  | X |  |  | X |  |  | X |  | X |  |
| Map |  |  |  |  |  | X |  |  |  |  |  |  |
| Share |  |  |  |  |  |  |  |  |  |  |  | X |
| Route |  |  |  |  |  |  | X |  |  |  |  |  |
| Interface | X | X | X | X | X | X | X | X | X |  | X | X |

**Table 4.1.4 Traceability Matrix**

**4.2 System Operation Contracts**

**OC-1**

**Precondition:** The system displays the main home page and displays the dropdown menu.

**Postcondition:** The system filters the results based on which option was selected.

**OC-2**

**Precondition:** The system displays the available pictures as a result of the categories selected.

**Postcondition:** Upon clicking a picture, a pop up which allows the user to see the price range of the hairstyle, a booking link and the Facebook share button; to share the hairstyle picture on Facebook.

**OC-3 Hairstyle**

**Precondition:** The user has clicked the hairstyle dropdown and selected the hairstyle of their choice.

**Postcondition:** The system returns a list of salons that offer the hairstyle selected.

**OC-4 Location**

**Precondition:** The user has clicked the location dropdown and selected the location where they want to find a salon.

**Postcondition:** The system returns a list of salons in the location selected.

**OC-5 Min and Max**

**Precondition:** The user has selected the Min/Max dropdown and selected the min/max amount of price range

**Postcondition:** The system returns a list of salons that offers hairstyles on the amount selected.

**OC-6**

**Precondition:** The system displays available options at the salon page, one of which is view salon on map.

**Postcondition:** Google map displays the salon on the map.

**OC-7**

**Precondition:** After the system has generated the salon lists.

**Postcondition:** The system returns the salon and provide a booking and view salon on map buttons.

**5. Design of Tests**

**5.1 Test cases**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-1** | |
| Use Case Tested: | **UC-1:Sign Up** | |
| Pass/fail Criteria: | **The test passes if the user enters a correct Facebook account or Google account with a correct password.** | |
| Input Data: | Text | |
| Test Procedure: | | Expected Result: |
| **Step 1. Type in an incorrect email address and/or password** | | **System asks the user to ensure that the email address/password provided is correct; prompts the user to try again** |
| **Step 2. Type in the correct email address and password** | | **System redirects the user to another page;** |

**Table 1 TC-1**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-2** | |
| Use Case Tested: | **UC-2 Sign In** | |
| Pass/fail Criteria: | **The test passes if the user enters a correct email address with a correct password.** | |
| Input Data: | Text | |
| Test Procedure: | | Expected Result: |
| **Step 1. Type in an incorrect email address and/or password** | | **System asks the user to ensure that the email address/password provided is correct; prompts the user to try again** |
| **Step 2. Type in the correct email address and password** | | **System redirects the user to another page;** |

**Table 2 TC-2**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-3** | |
| Use Case Tested: | **UC-1 Sign Up** | |
| Pass/fail Criteria: | **The test passes if the user enters a valid email address with a password of not less than 6 characters.** | |
| Input Data: | Text | |
| Test Procedure: | | Expected Result: |
| **Step 1. Type in an invalid email address and/or password** | | **System asks the user to ensure that the email address/password provided is valid; prompts the user to try again** |
| **Step 2. Type in the correct/valid email address and password** | | **System registers the user to the database;** |

**Table 3 TC-3**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-4** | |
| Use Case Tested: | **UC-6 :Search By Category** | |
| Pass/fail Criteria: | **The test passes when the user specifies the Hairstyle category in the list then the hairstyles corresponding to those categories ONLY appear on the Users screen.** | |
| Input Data: |  | |
| Test Procedure: | | Expected Result: |
| **Step 1. Select the Hairstyle Category you want to view from the drop down list** | | **The selected category will be marked** |
| **Step 2. Click on the Search Button** | | **A list of Salons that offer the selected Hairstyle Category will be displayed on the Screen.** |

**Table 4 TC-4**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-5** | |
| Use Case Tested: | **UC-12 Photographs** | |
| Pass/fail Criteria: | **The test passes if the user can view all the pictures of hairstyles a salon has registered with our databases.** | |
| Input Data: |  | |
| Test Procedure: | | Expected Result: |
| **Step 1. Click on the *View Salon Gallery* Button for any Salon** | | **The application redirects the user to a page with the picture gallery matched with that Salon.** |
|  | |  |

**Table 5 TC-5**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-6:Booking** | |
| Use Case Tested: | **UC-10** | |
| Pass/fail Criteria: | **The test passes if the user can make a booking with a Salon of their choice.** | |
| Input Data: | String | |
| Test Procedure: | | Expected Result: |
| **Step 1. Click on the *View Salon Gallery* Button for any Salon**  **Step 2. Click on any hairstyle picture**  **Step 3. Fill in all the required fields and then click on the *Submit* button** | | **The application redirects the user to a page with the picture gallery matched with that Salon.**  **A page with the booking form will appear on the user’s screen.**  **A new page will be displayed and will signal completion.** |
|  | |  |

**Table 6 TC-6**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-7** | |
| Use Case Tested: | **UC-9** | |
| Pass/fail Criteria: | **The test passes if the user can view an address of any Salon they want.** | |
| Input Data: | Submit Button | |
| Test Procedure: | | Expected Result: |
| **Step 1. Search for Salons by Location and Category** | | **The application redirects the user to a page with a list of Salons that offer the selected Hairstyle Category at the selected Location, the Salon address will appear below the Salon name.** |
|  | |  |

**Table 7 TC-7**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-8** | |
| Use Case Tested: | **UC-7** | |
| Pass/fail Criteria: | **The test passes if the user can see their desired Salon location on a map.** | |
| Input Data: | **Submit Button** | |
| Test Procedure: | | Expected Result: |
| **Step 1. Search for Salons by Location and Category**  **Step 2. Click on the *View Salon On Map* button** | | **The application redirects the user to a page with a list of Salons that offer the selected Hairstyle Category at the selected Location, and next to the Salon name, there is a *View Salon On Map* button.**  **The application will display a *Google* map with the Salon pinned on it.** |
|  | |  |

**Table 8 TC-8**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-9** | |
| Use Case Tested: | **UC-5** | |
| Pass/fail Criteria: | **The test passes if the user can search for Salons by their location.** | |
| Input Data: | **Submit Button** | |
| Test Procedure: | | Expected Result: |
| **Step 1. On the home page, Select the location from the dropdown list of locations we cater for, and then click on the *Search* button.**  **Step 2. Select Hairstyle Category,**  **And then click on the *Search* button** | | A second search option will appear, that is to select the Hair Category.  The application will display a list of the Salons in that selected location. |
|  | |  |

**Table 9 TC-9**

|  |  |  |
| --- | --- | --- |
| Test-case Identifier: | **TC-10** | |
| Use Case Tested: | **UC-8** | |
| Pass/fail Criteria: | **The test passes if the user can see an outlined route on a map from his/her current location** | |
| Input Data: | **Submit Button** | |
| Test Procedure: | | Expected Result: |
| **Step 1. On the home page, Select the location from the dropdown list of locations we cater for, and then click on the *Search* button.**  **Step 2. Select Hairstyle Category,**  **And then click on the *Search* button**  **Step 3.Click the *View Salon on map* button,** | | **A second search option will appear, that is to select the Hair Category.**  **The application will display a list of the Salons in that selected location, and then** |
|  | |  |

* 1. **Functional Unit Tests**

Test Case ID: TC1 Search By Price Range

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Steps** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1. | Choose A Hairstyle | “Chiskop” | Nothing should happen | Nothing happened | Pass | N/A |
| 2. | Choose the minimum price range | “15” | Nothing should happen | Nothing happened | Pass | N/A |
| 3. | Choose the maximum price range | “15” | Nothing should happen | Nothing happened | Pass | N/A |
| 4. | Click the button | N/A | A list of hair salons should appear | No List Appeared | Fail |  |

TC1 Search By Price Range

Test case ID: TC2 Sign Up

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Steps** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1. | Select the sign Up button | N/A | A pop Up with a sign Up form | A pop Up with a sign Up form | Pass | N/A |
| 2. | Choose to sign up using a Facebook account or manually sign up by filling in the form. | “Name”,  “Surname”,  “email address” and password. | Details to be added to the database. | Details are added to the database | Pass | N/A |

TC2 Sign up

Test case ID: TC3 Sign In

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Steps** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1. | Select the sign In button | N/A | A pop Up with a sign Up form | A pop Up with a sign Up form | Pass | N/A |
| 2. | Choose to sign up using a Facebook account or manually sign up by filling in the form. | “Name”,  “Surname”,  “email address” and password. | A new page with a dropdowns. | A new page with a dropdowns. | Pass | N/A |

TC3 Sign In