****

Topic: Marketplace

Project Name: **Drigo**

**Table of content**

* 1. **Overview…………………………………………………………………………………………………………….3**
     1. Background…………………………………………………………………………………………………3
     2. Objective………………………………………………………………………………………………………3
     3. Scope……………………………………………………………………………………………………………4
     4. Assumptions and Constraints………………………………………………………………….5
     5. Dependencies and Risks…………………………………………………………………………..5
  2. **Project Delivery…………………………………..…………………………………………………………….5**
     1. Deliverables ……………………………………………………………………………………………….5
     2. Timescales ………………………………………………………………………………………………….6
     3. Work Distribution ……………………………………………………………………………………….6
     4. Project Resources ………………….………………………………………………………………….6
  3. **Summary…………………………………..………………………………………………………………………..7**
  4. **References………………………………………………………………………………………………………….8**
  5. **Requirement Specification..………………..…………………………………………………………….8**
     1. Functional Requirement ……….………………………………………………………………….8
     2. Non-Functional Requirement…………………………………………………………………….10
  6. **Use Case Diagram…………………………………………………………………………………………….12**
  7. **Case Description……………………………………………………………………………………………….13**
  8. **Activity Diagram…………………………………………………………………………………………………22**
  9. **Sequence Diagram…………………………………………………………………………………………….26**

* 1. **Overview**
     1. **Background**

In the traditional system, if a customer wants to take a service or buy a product, he first has to go to the service centre or a store and check whether the product is available in the store. If it is not in the store, it has to be found in another store. There is no guarantee that I will find the product after searching a few stores. Also, when we travel to another place, the place is new to me and I don't have any information about it, so if I need to go to a hospital or a food hotel, it's very difficult to find it and I know right? I can reach my desired goal after walking how many kilometres.

Even after that if I get a food hotel I don't know exactly what kind of service it provides. And I don't know when the shop is open or when it is closed.

In other words, we need many types of products or services in our daily life, which we have to suffer a lot to get.

**1.1.2 Objectives**

Drigo will be a web-based system. Where a customer can see the information of what kind of products are stocked or services offered in each store around his location and he can easily use the searching feature of the web system to find the product or service he needs in the nearest store from his location. There are In this case, if he does not know the location of the shop, there is no problem, he can get the location here through Google Maps. If he wants, he can easily reach the shop using Google Maps. Moreover, the customer can see when the shop is closed and what time it is open.

Here every shopkeeper will get a platform to sell their products or provide services online.

And every customer can rate the shopkeeper bus service holder by giving their opinion.

Here the user will be:

1. Buyer User
2. Seller User

The list of operations that the system will provide are-

1. Provide the nearest shop/service information to the seller user dashboard.
2. Enabling the buyer to view the seller’s profile with a rating.
3. Provide the feature to get the needed product’s/service’s nearest shop/office.
4. Enabling the buyer to provide a rating to the seller profile.
5. Provide the feature to find out the needed shop/office location using Google Maps.

**1.1.3 Scope**

With this system, a user can easily see where he can get his desired product and service at home in a very short time, whereas in the traditional rules, he had to waste a lot of time to find the product or service and go to the store or office to check the actual product. It was time-consuming whether the service was available here. At the same time, a user can go to an unfamiliar location and see very easily which organization will be good for taking his service or product because many customers who have received the service before have rated the organization as good or bad. Whereas in traditional rules a customer could have been cheated very easily.

Moreover, a user can see what kind of service or product is available in all the organizations of his location and the system will suggest the various organizations around his location to the user.

Each seller user can market their offline products online and communicate with their buyers. By doing this, it will be very easy for a user to find the necessary things and he will know whether the organization he needs is open or closed, it will save a lot of time.

**1.1.4 Assumptions and Constraints**

It is assumed that the user is comfortable with computers. Every user must have good knowledge of web surfing. Especially those who will be buyer users must have the idea of ​​running Google Maps. The login and password will be used to identify the user and after entering the system, guests can search for products/services and find the location of the specific product or service through Google Maps.

**1.1.5 Dependencies and Risks**

Users must have web access to use the system. Buyer user or shop owner/service holder needs a huge community to manage the system properly.

The main risk behind project implementation is security. Since this system will maintain a huge community, there will be a lot of data transactions, in which case if someone hacks the system, it will be a complete disaster. So this will be one of our main concerns during development.

**1.2 Project Delivery**

**1.2.1** **Deliverables**

The following contents will be delivered with the project:

1. Project CD
   1. Project Demo
   2. User manual along with Tutorial
2. Documentation

**1.2.2 Timescales**

The time frame for implementing the project is given in Figure 1.2.1.

**1.2.3 Work Distribution**

The work distribution of the project is given in Table 1.2.1.

**1.2.4 Project Resources**

The resources required to finish the project are given in Table 1.2.2.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Duration** | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | 7 days | 7 days | 7 days | 7 days | 7 days | 7 days | 7 days | 7 days | 7 days |
|  |  |  |  |  |  |  |  |  |  |  |
| **Project Proposal** | |  |  |  |  |  |  |  |  |  |
| **SRS** |  |  |  |  |  |  |  |  |  |  |
| **Designing** | |  |  |  |  |  |  |  |  |  |
| **Coding** |  |  |  |  |  |  |  |  |  |  |
| **Testing** | |  |  |  |  |  |  |  |  |  |
| **Finalization** | |  |  |  |  |  |  |  |  |  |

**Figure 1.2.1: Time frames for project implementation**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Project Proposal | MD. MUSTAFIZUR RAHMAN | 7days |
| Software Requirement Specification | MD. MUSTAFIZUR RAHMAN | 7 days |
| Software Design | MD. MUSTAFIZUR RAHMAN | 21 days |
| Coding | MD. MUSTAFIZUR RAHMAN | 14 days |
| Software Testing | MD. MUSTAFIZUR RAHMAN | 7 days |
| Project Finalization | MD. MUSTAFIZUR RAHMAN | 7 days |

**Table 1.2.1 Work Distribution**

|  |  |  |  |
| --- | --- | --- | --- |
| **Hardware Requirements** | | | |
| **Processor** | **RAM** | | **Hard Disk Space** |
| Pentium II or higher | 64 Mb or higher | | 128 Mb or higher |
| **Software Requirements** | | | |
| **Operating System** | | **Database** | |
| For users, no specific OS is required. The server machine must have Windows XP/Vista/7 along with .NET framework 4 and IIS. | | SQL Server 2008 | |

**Table 1.2.2 Project Resources**

**1.3.** **Summary**

Time is our most valuable asset. We cannot waste it when there is the scope of utilizing it in a better way. Our proposed system named Drigo will be developed to meet this purpose. It will save a customer's time in finding the desired product or service and marketing time to a seller or service holder. It will also help a customer make the right decision so that the chances of being cheated or harassed will be reduced.

**1.4. References**

1. HowTo: Write a project proposal [Online] URL:

**http://mogadalai.wordpress.com/2007/05/28/howto-write-a-project-proposal** (Accessed on

13.02.2011)

2. Health Onclick [Online] URL:

**http://www.healthonclick.com** (Accessed on 15.02.2011)

3. Apollo Hospitals [Online] URL:

**http://www.apollohospitals.com** (Accessed on 15.02.2011)

**1.5 Requirement Specification**

The complete requirement specification based on the elicitation process is described in this section.

**1.5.1 Functional Requirements**

The Functional Requirements Specification is designed to be read by a general audience. Readers should understand the system, but no particular technical knowledge should be required to understand the document

|  |  |
| --- | --- |
| **FR-01** | **Registration & Login** |
|  | After entering the URL via any web browser, in the software index page and the user want to access their profile/dashboard/upload new product info, they must register first. After every time of access, he has to log in first before accessing these things.  On the other hand, if a seller user wants to buy some product via the site ordering process he needs to log in / register otherwise none. |
| **Stakeholder** | **Seller User** |

|  |  |
| --- | --- |
| **FR-02** | **User Profile** |
|  | A seller can create a user profile on the Drigo platform.  Through this, they will convey their identity in this marketplace.  It’s created through the information. Should not be allowed to have more than one profile. |
| **Stakeholder** | **Seller User** |

|  |  |
| --- | --- |
| **FR-03** | **Update User Information** |
|  | Users can update their old information. The users can update the details of the members |
| **Stakeholder** | **Seller User** |

|  |  |
| --- | --- |
| **FR-04** | **Upload Content/ Product** |
|  | When a user is operating this social media that is the time he can upload content such as video, photo, text etc |
| **Stakeholder** | **Seller User** |

|  |  |
| --- | --- |
| **FR-05** | **Submit** |
|  | Users can submit their updated information to the server. |
| **Stakeholder** | **Seller User** |

|  |  |
| --- | --- |
| **FR-06** | **Search Content** |
|  | A user can search the specific content information with shop location information and location. |
| **Stakeholder** | **Seller User, Buyer User** |

|  |  |
| --- | --- |
| **FR-07** | **Select Content** |
|  | A Buyer user can select a specific category of content. |
| **Stakeholder** | **Buyer User** |

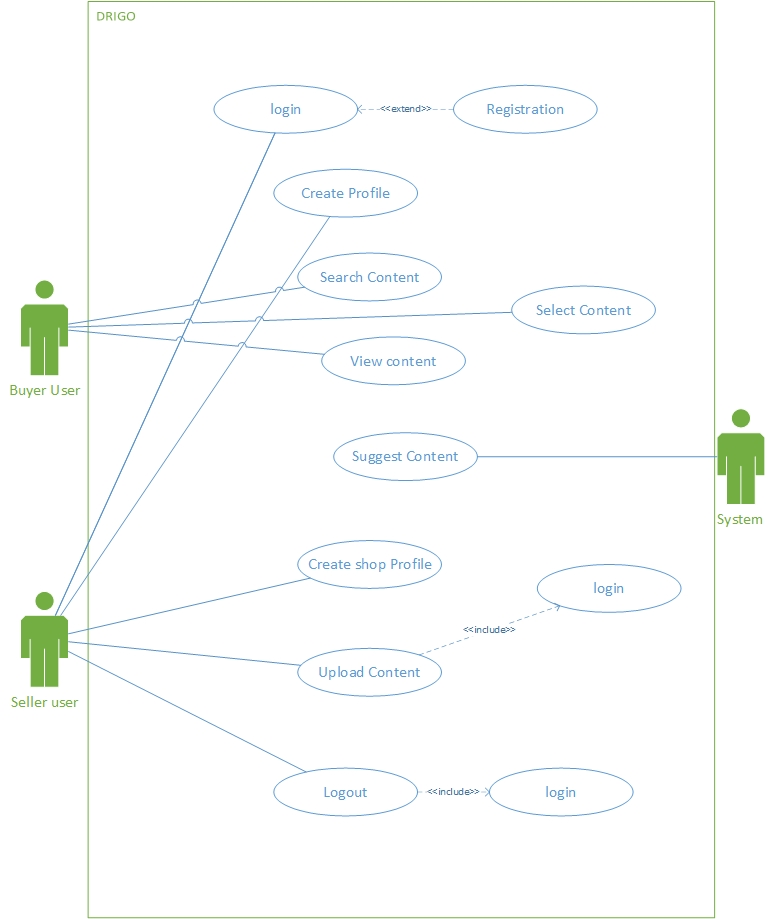
|  |  |
| --- | --- |
| **FR-08** | **Create Shop Profile** |
|  | A seller user can create his shop profile to display his product. |
| **Stakeholder** | **Seller User** |

**1.5.2 Non-Functional Requirement**

1. **Security**
2. **More Accessibility**
3. **More reliability**
4. **More Affordability**
5. **More Sustainability**
6. **Active Log.**
7. **Shop Rating**

**1.6 Use Case Diagram**

Use Case Diagram

****

**1.7 Case Description**

|  |  |
| --- | --- |
| **Use Case** | Registration |
| **Goal**  <a longer statement of the goal in context if needed> | Create a new account in the system |
| **Preconditions**  <what we expect is already the state of the world> |  |
| **Success End Condition**  <the state of the world if goal abandoned> | A new user can to the register system |
| **Failed End Condition**  <the state of the world if goal abandoned | A new user can’t to the register system |
| **Primary Actors:**  **Secondary Actors:** | Seller user |
| **Trigger**  <the action upon the system that start use case> | Send a registration request |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | Using email address | | 1.1 | Using username | | 1.2 | Using a password its must be 8 characters | | 2 | Submit the register request | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | | 1a | Invalid email address system show an invalid data message | | 2a | Invalid username system show an invalid data message | | 3a | Invalid password system show an remainder message | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | | 1 | Submit the registration request time should be 5 s | |

|  |  |
| --- | --- |
| **Use Case** | Login |
| **Goal**  <a longer statement of the goal in context if needed> | Log into the system with username and password |
| **Preconditions**  <what we expect is already the state of the world> | User must register into the system |
| **Success End Condition**  <the state of the world if goal abandoned> | User can logged into the system |
| **Failed End Condition**  <the state of the world if goal abandoned | User can’t logged into the system |
| **Primary Actors:**  **Secondary Actors:** | Seller user |
| **Trigger**  <the action upon the system that start use case> | Send a Login request |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | Enter Username | | 1.1 | Enter password | | 2 | Valid username and password | | 2 | Allow access to the system | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | | 1a | Invalid username system show wrong username message | | 2a | Invalid password system show wrong password message | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | | 1 | Login time should be 5s | |

|  |  |
| --- | --- |
| **Use Case** | Create profile |
| **Goal**  <a longer statement of the goal in context if needed> | Make a Unique identity and craft start-up fir this online media |
| **Preconditions**  <what we expect is already the state of the world> | User must be logged into the system |
| **Success End Condition**  <the state of the world if goal abandoned> | User can upload their product |
| **Failed End Condition**  <the state of the world if goal abandoned | User can’t upload their product |
| **Primary Actors:**  **Secondary Actors:** | Seller user |
| **Trigger**  <the action upon the system that start use case> | A new profile create request comes in. |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | User sends new profile to create request | | 1.1 | User send a request in via Browser | | 2 | Drigo can capture user’s First Name, Last Name, username, email, address, gender | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | | 1 | Process response should be 5s | |

|  |  |
| --- | --- |
| **Use Case** | Suggest Content |
| **Goal**  <a longer statement of the goal in context if needed> | System suggest content via a user location |
| **Preconditions**  <what we expect is already the state of the world> |  |
| **Success End Condition**  <the state of the world if goal abandoned> | Suggest content based on the nearest location |
| **Failed End Condition**  <the state of the world if goal abandoned | System don’t suggest content based on the nearest location |
| **Primary Actors:**  **Secondary Actors:** | Buyer user |
| **Trigger**  <the action upon the system that start use case> |  |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | System researching with the user location | | 1.1 | System suggest nearest content/shop | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | | **1** | If the system can’t get the user’s location then it can’t suggest the nearest content/shop | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | | 1 | Suggest content time should be 5s | |

|  |  |
| --- | --- |
| **Use Case** | Search Content |
| **Goal**  <a longer statement of the goal in context if needed> | Find out a specific contnent |
| **Preconditions**  <what we expect is already the state of the world> |  |
| **Success End Condition**  <the state of the world if goal abandoned> | User can find specific content easily |
| **Failed End Condition**  <the state of the world if goal abandoned | User can’t find specific content easily |
| **Primary Actors:**  **Secondary Actors:** | Buyer user |
| **Trigger**  <the action upon the system that start use case> | Call search request |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | Input content or users information | | 1.1  1.2 | Submit search request with the information  System show the output or specific thinks which you want | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | | **1** | The finding information out of the server | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | | 1 | Search time should be 2s | |

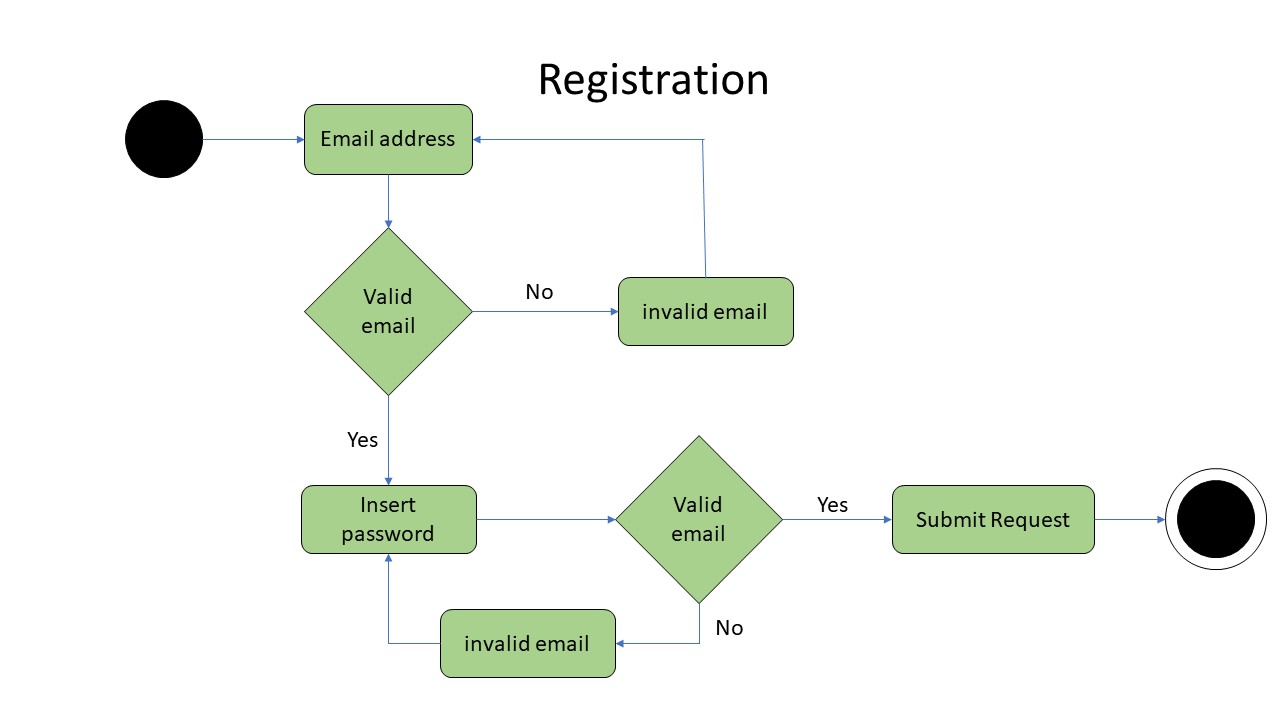
|  |  |
| --- | --- |
| **Use Case** | Select Content |
| **Goal**  <a longer statement of the goal in context if needed> | Select a specific content |
| **Preconditions**  <what we expect is already the state of the world> |  |
| **Success End Condition**  <the state of the world if goal abandoned> | User can select specific content |
| **Failed End Condition**  <the state of the world if goal abandoned | User can select specific content |
| **Primary Actors:**  **Secondary Actors:** | Buyer user |
| **Trigger**  <the action upon the system that start use case> | Click the menu button |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | Go to the category list | | 1.1 | Select the specific content | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | | **1.1a** | Finding information out of the category | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | | 1 | select time should be 2s | |

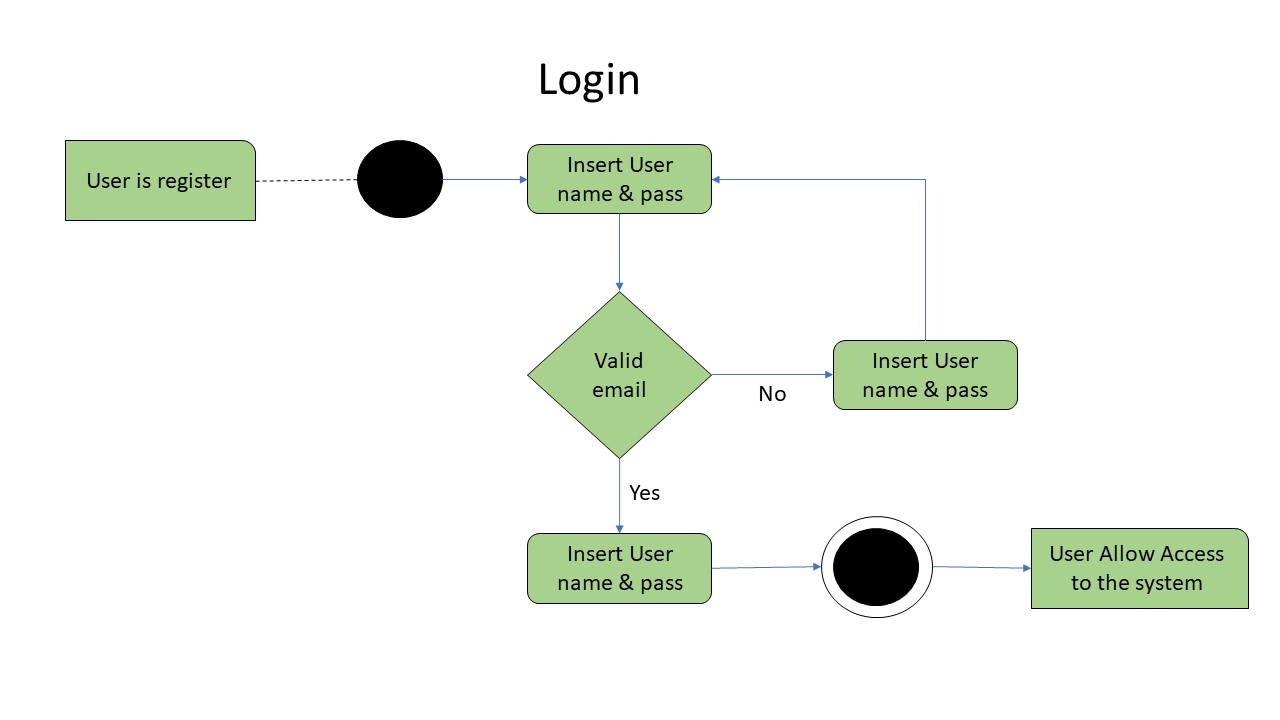
|  |  |
| --- | --- |
| **Use Case** | Shop Profile |
| **Goal**  <a longer statement of the goal in context if needed> | Create shop Profile |
| **Preconditions**  <what we expect is already the state of the world> |  |
| **Success End Condition**  <the state of the world if goal abandoned> | Users can make their own shop profile. |
| **Failed End Condition**  <the state of the world if the goal abandoned | The user can’t make his shop profile. |
| **Primary Actors:**  **Secondary Actors:** | Buyer user |
| **Trigger**  <the action upon the system that start use case> | Send a Request to make a new shop profile |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | Input shop name | | 1.1 | Input trad license number | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | |

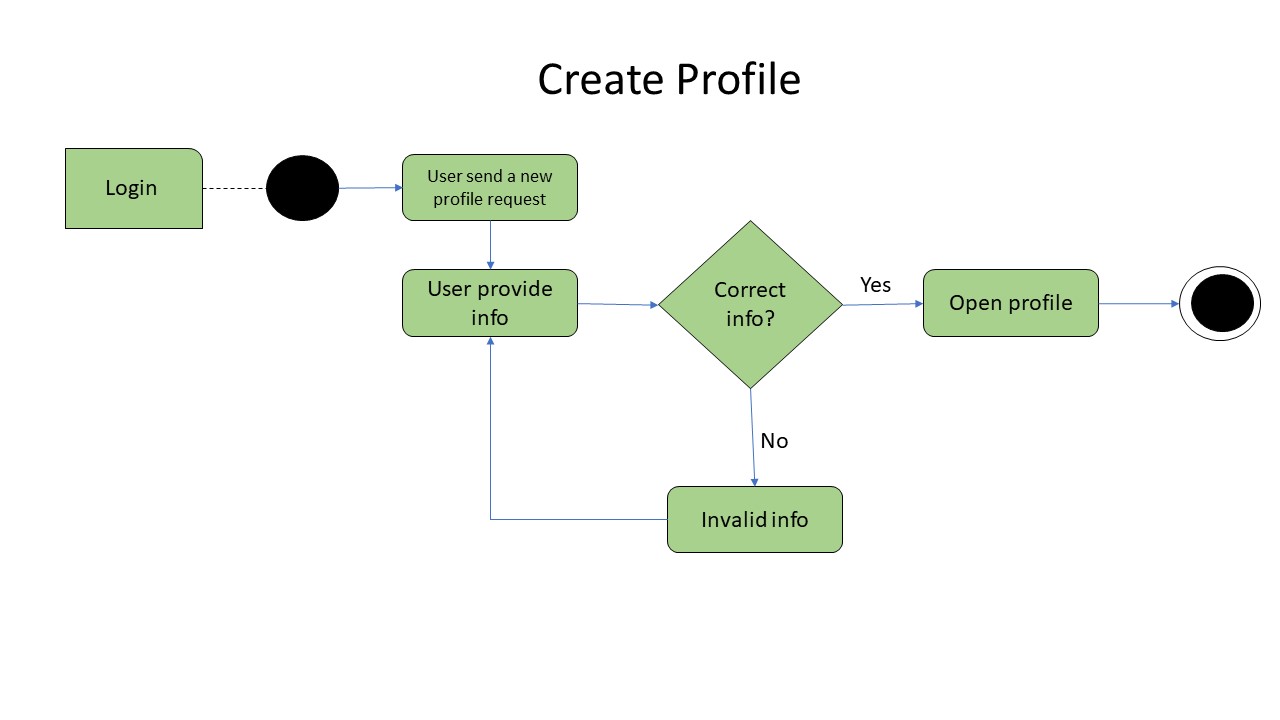
|  |  |
| --- | --- |
| **Use Case** | Upload content |
| **Goal**  <a longer statement of the goal in context if needed> | Users can upload a new product information |
| **Preconditions**  <what we expect is already the state of the world> | Users must log into the system |
| **Success End Condition**  <the state of the world if goal abandoned> | Users can upload new product |
| **Failed End Condition**  <the state of the world if the goal abandoned | The user can’t upload new product |
| **Primary Actors:**  **Secondary Actors:** | Buyer user |
| **Trigger**  <the action upon the system that start use case> |  |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | Upload new product information like image | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | |

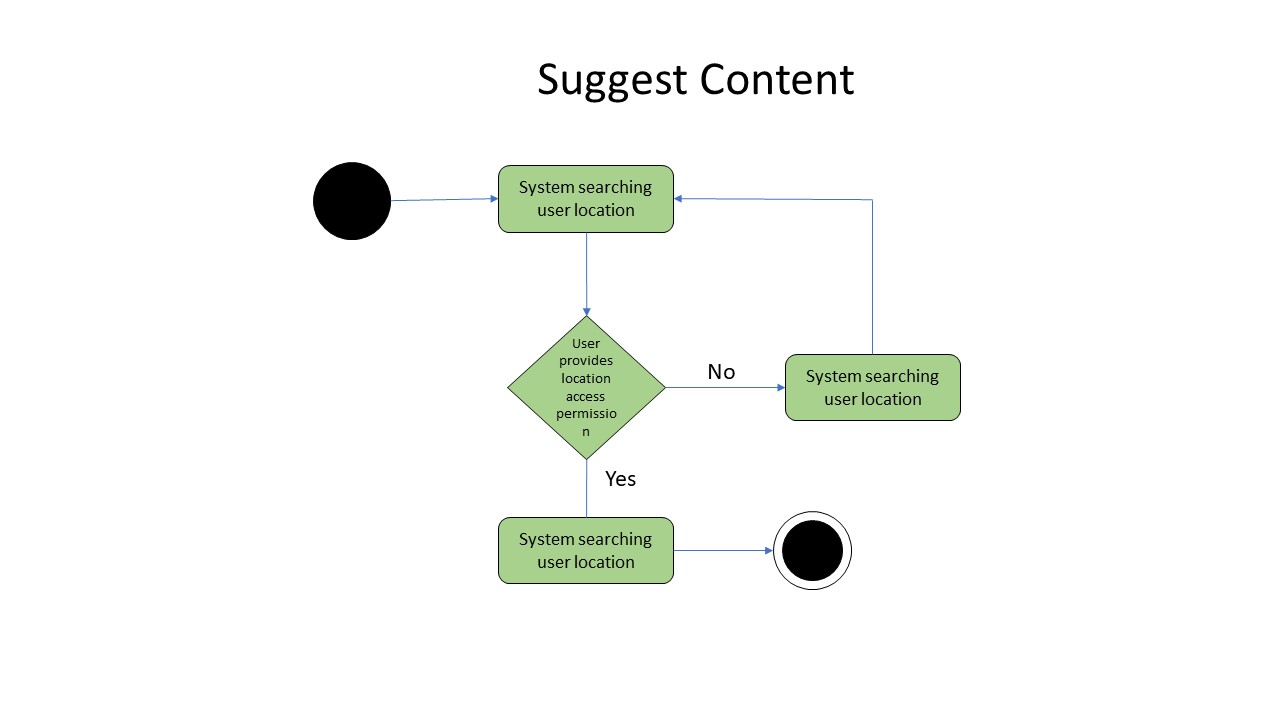
|  |  |
| --- | --- |
| **Use Case** | Logout |
| **Goal**  <a longer statement of the goal in context if needed> | Logout into the system |
| **Preconditions**  <what we expect is already the state of the world> | The system must logged into the system |
| **Success End Condition**  <the state of the world if goal abandoned> | Users can logout into the application |
| **Failed End Condition**  <the state of the world if the goal abandoned | The user can’t logout into the application |
| **Primary Actors:**  **Secondary Actors:** | Buyer user |
| **Trigger**  <the action upon the system that start use case> | Send logout request |
| **Description/ Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean-up after> | |  |  | | --- | --- | | **Step** | **Action** | | 1 | Select logout | | 1.1 | Submit a logout request it the system | |
| **Alternative Flows**  <a: condition causing branching>  <al: action or name of sub-use case> | |  |  | | --- | --- | | **Step** | **Branching Action** | |
| **Quality Requirements** | |  |  | | --- | --- | | **Step** | **Requirement** | | 1 | Logout time should be 5s | |

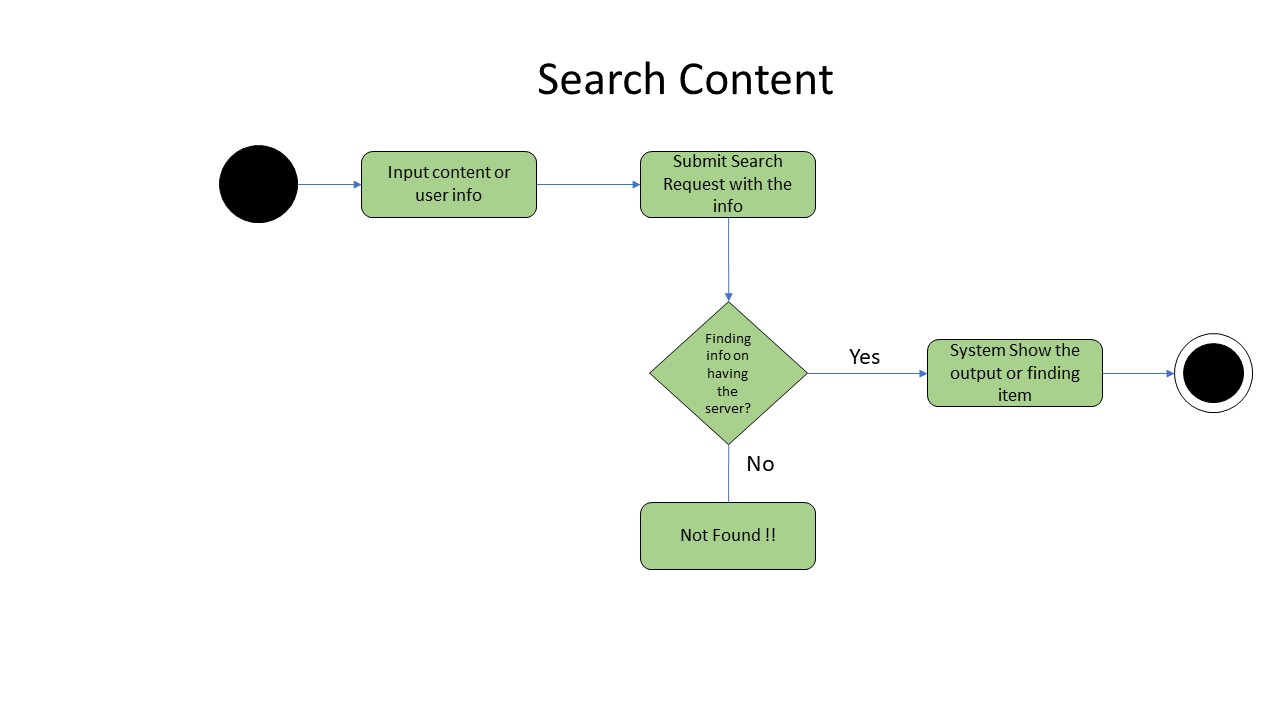
**1.8 Activity Diagram**

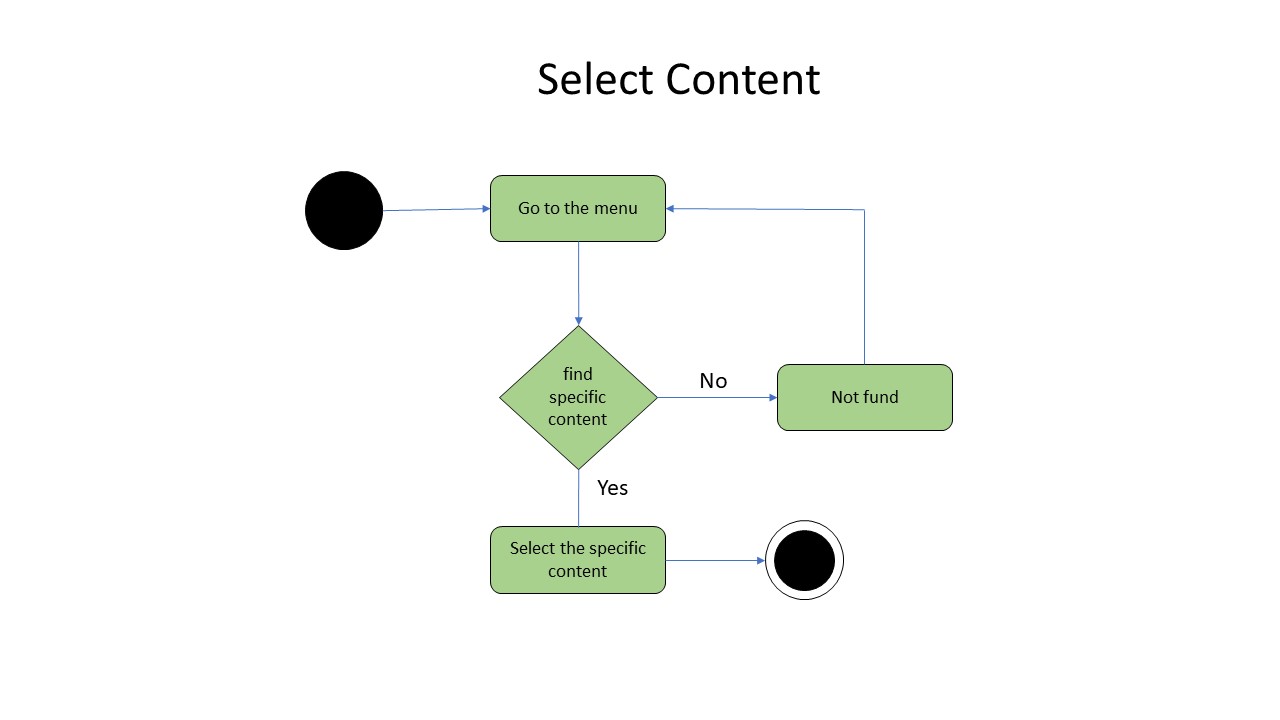
****

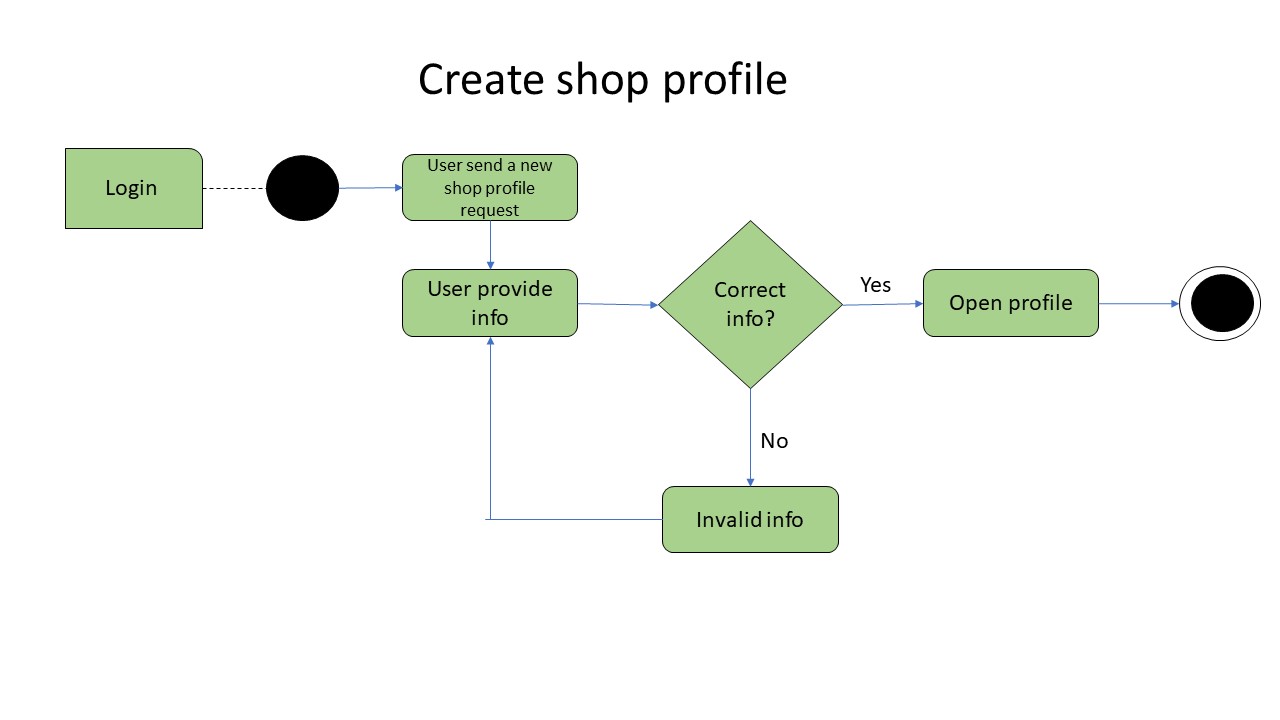
****

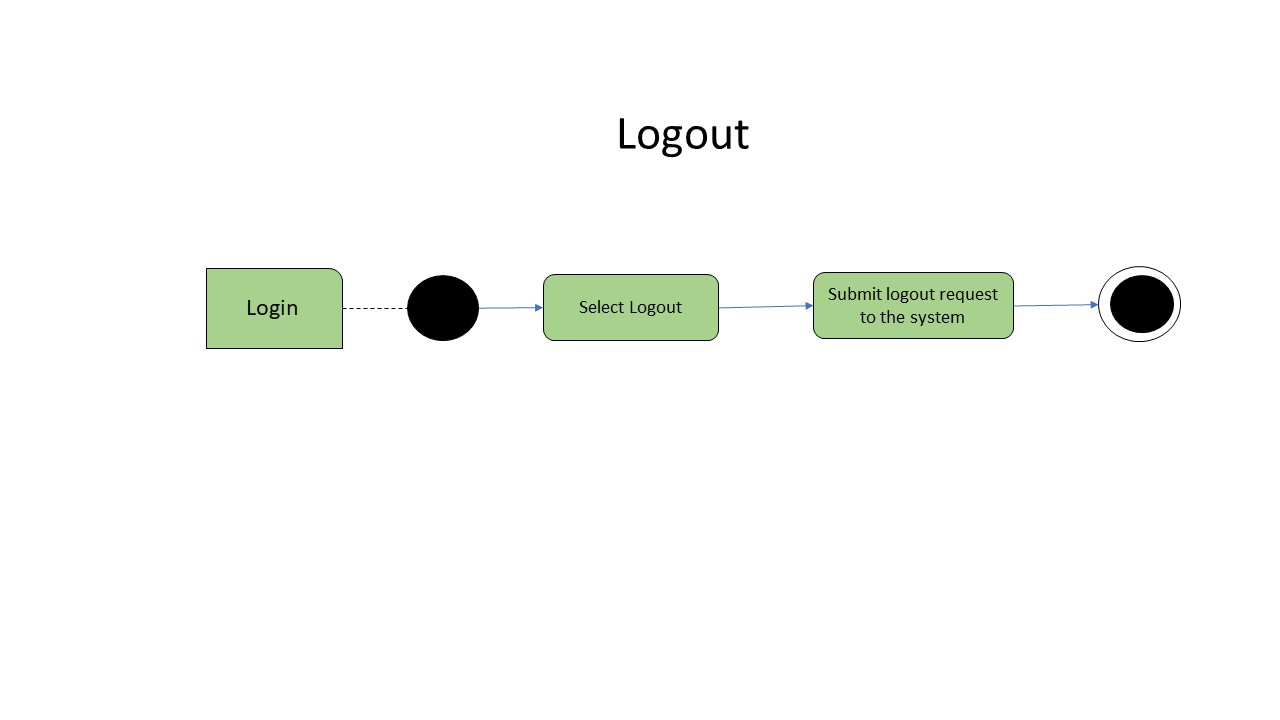
****

****

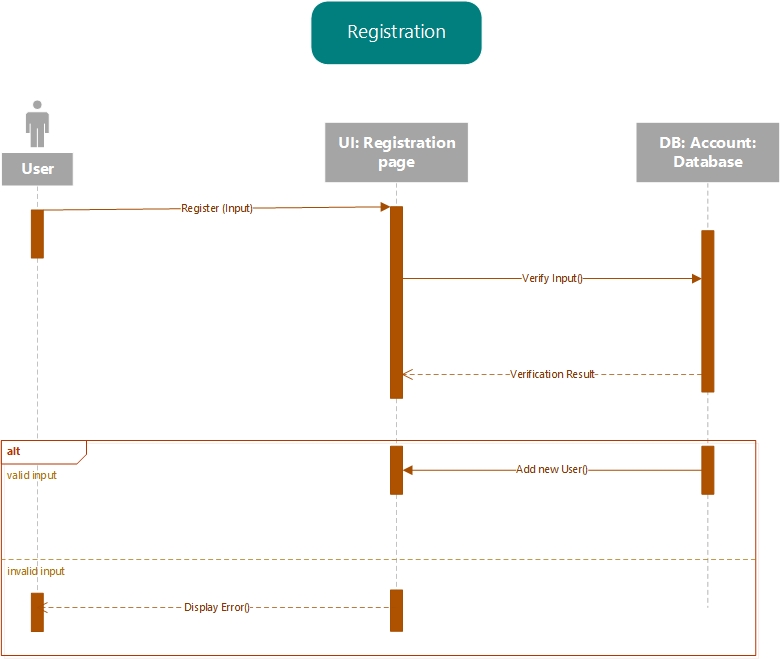
****

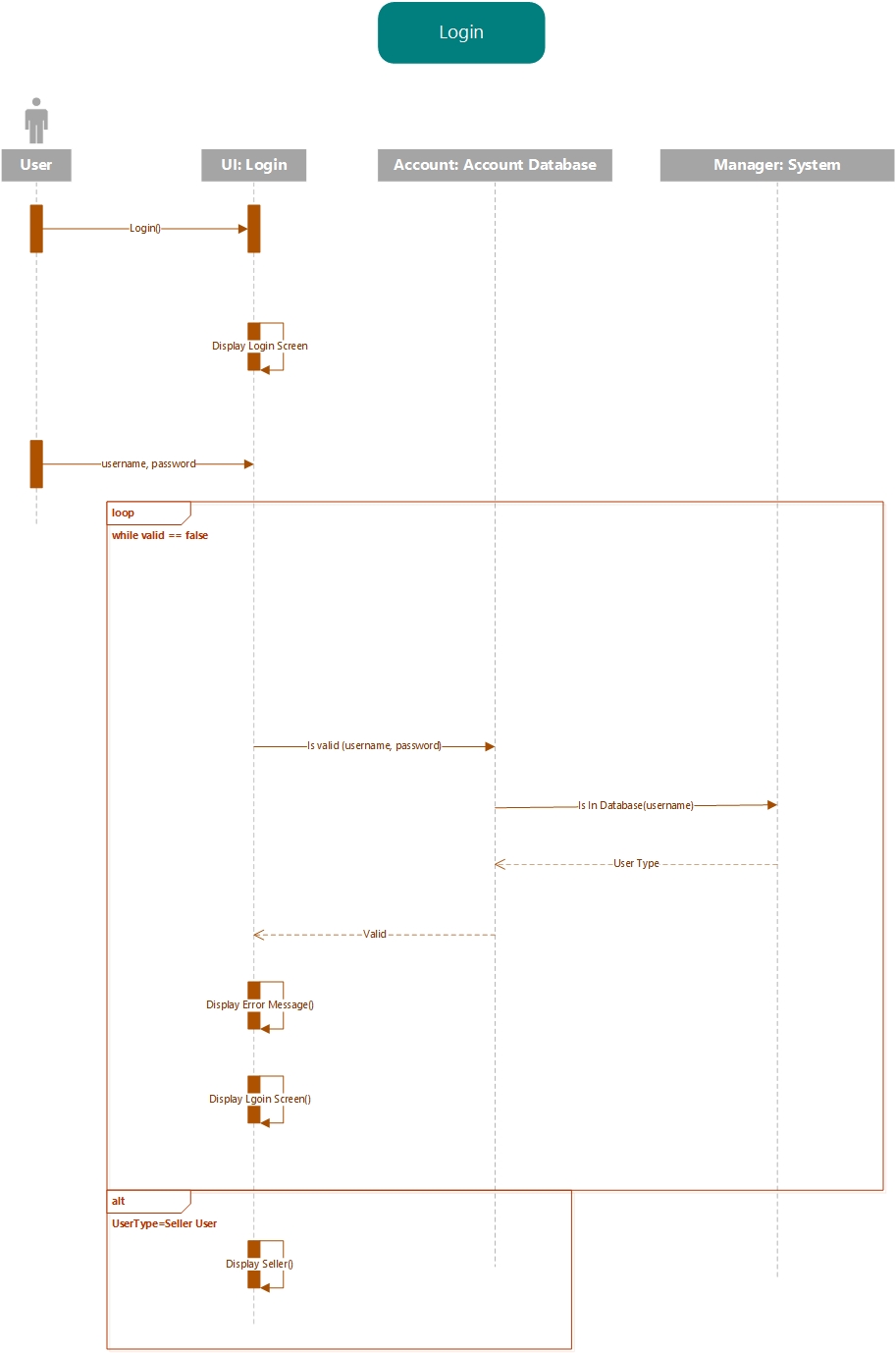
****

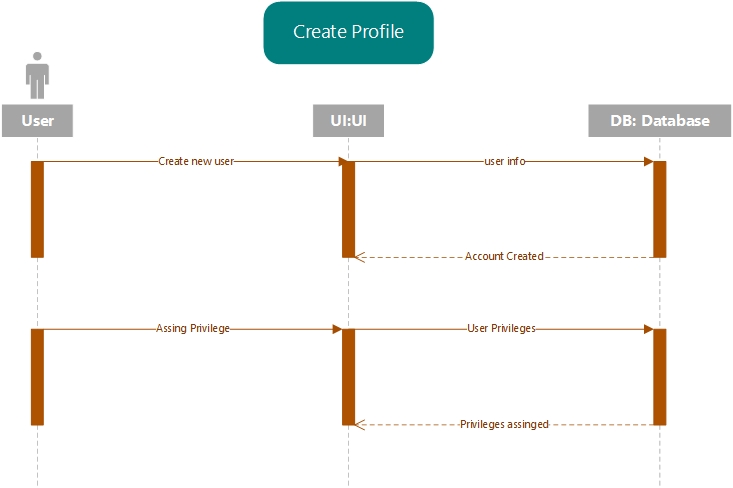
****

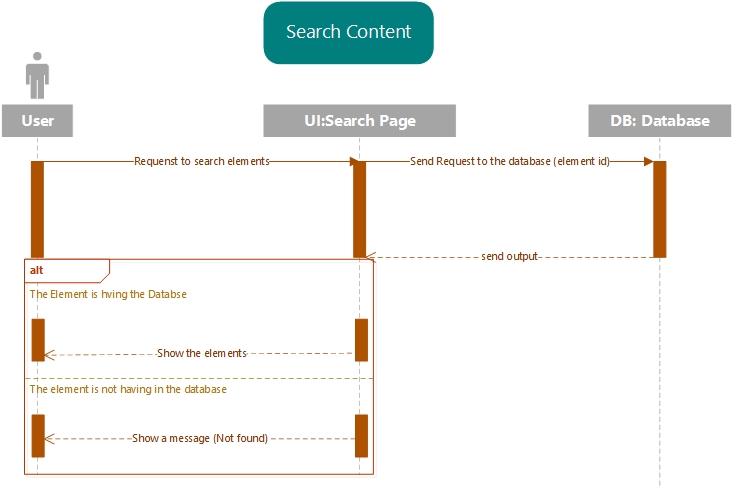
****

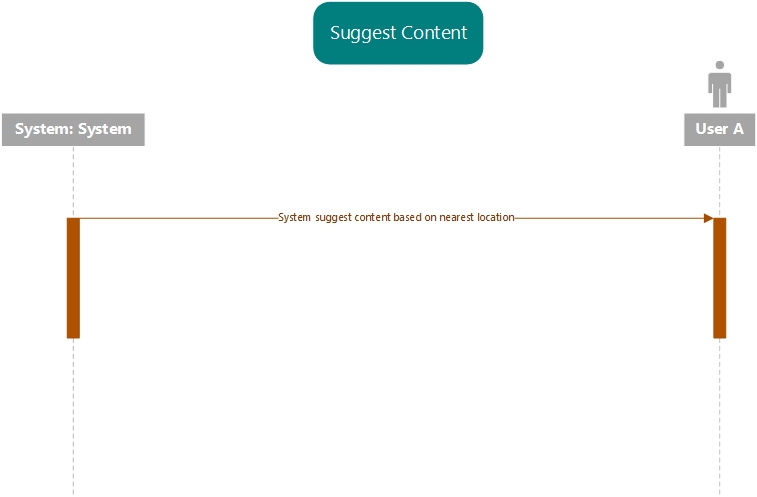
**1.9 Sequence Diagram**

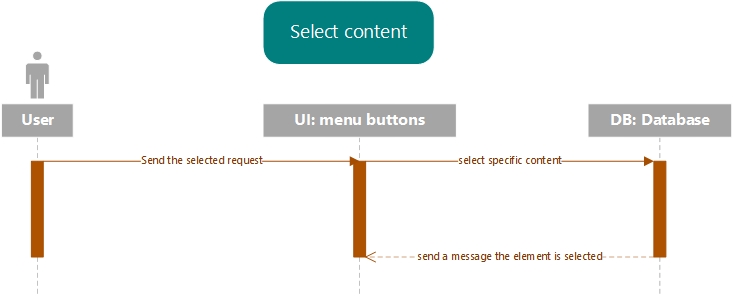
****

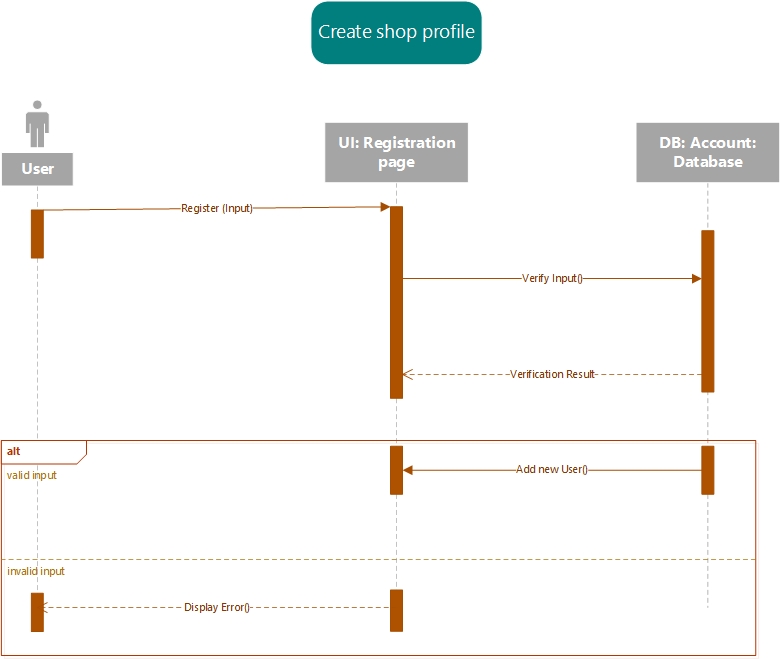
****

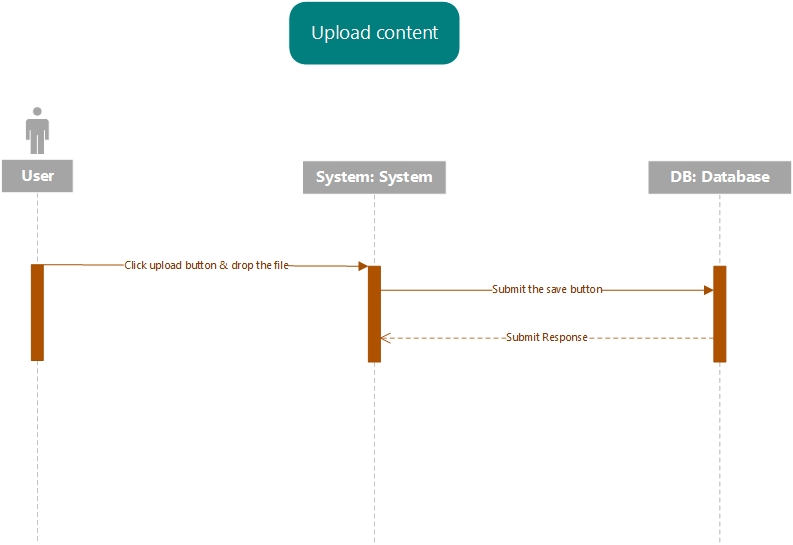
****

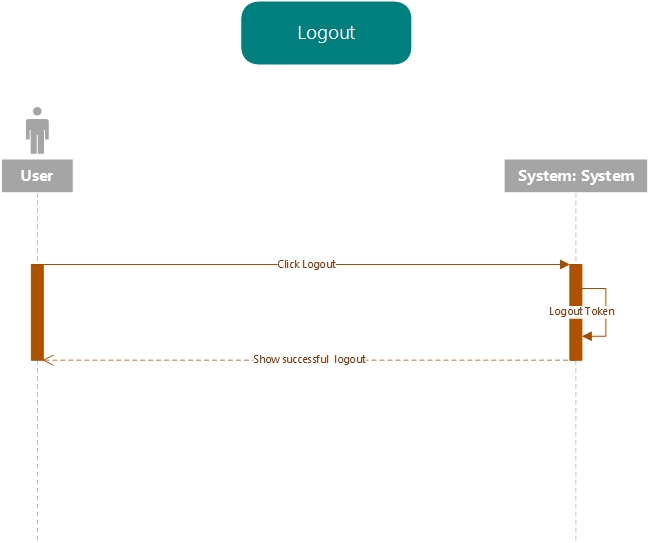
****

****

****

****

****

****