Topic: Marketplace

Project Name: **Drigo**

Table of content

1.1 Overview

1.1.1 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.
6.

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:

a) Project CD
    i. Project Demo
    ii. User manual along with Tutorial

b) 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.

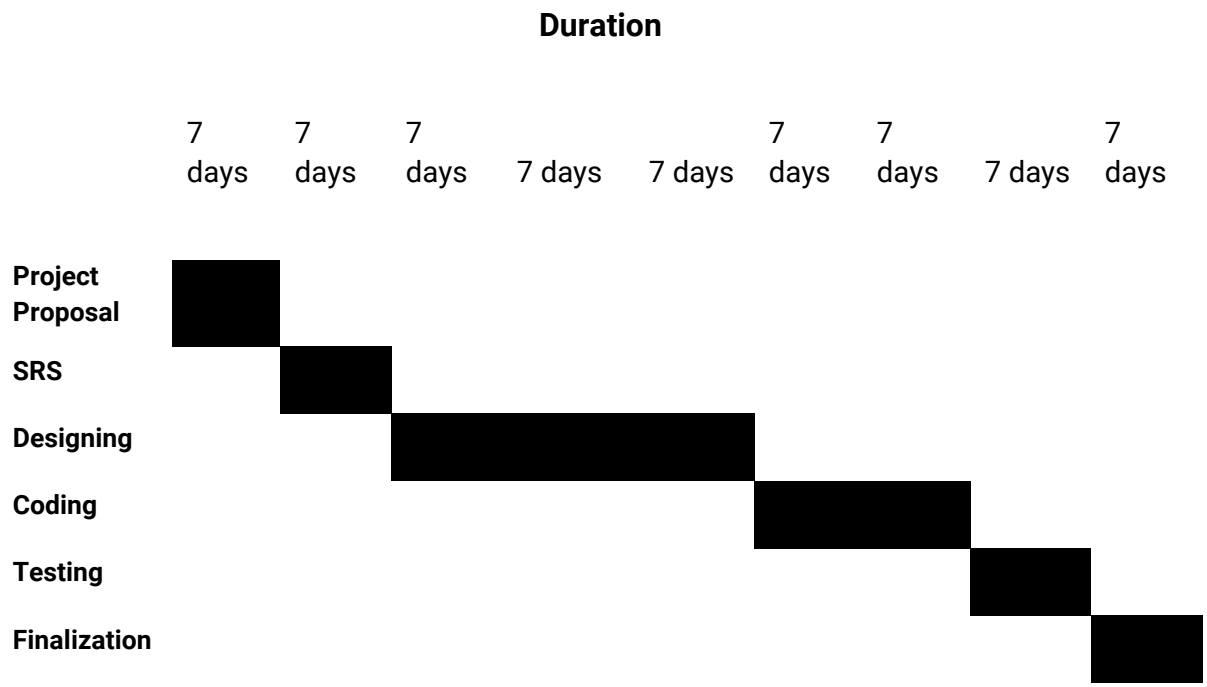


Figure 1.2.1: Time frames for project implementation

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

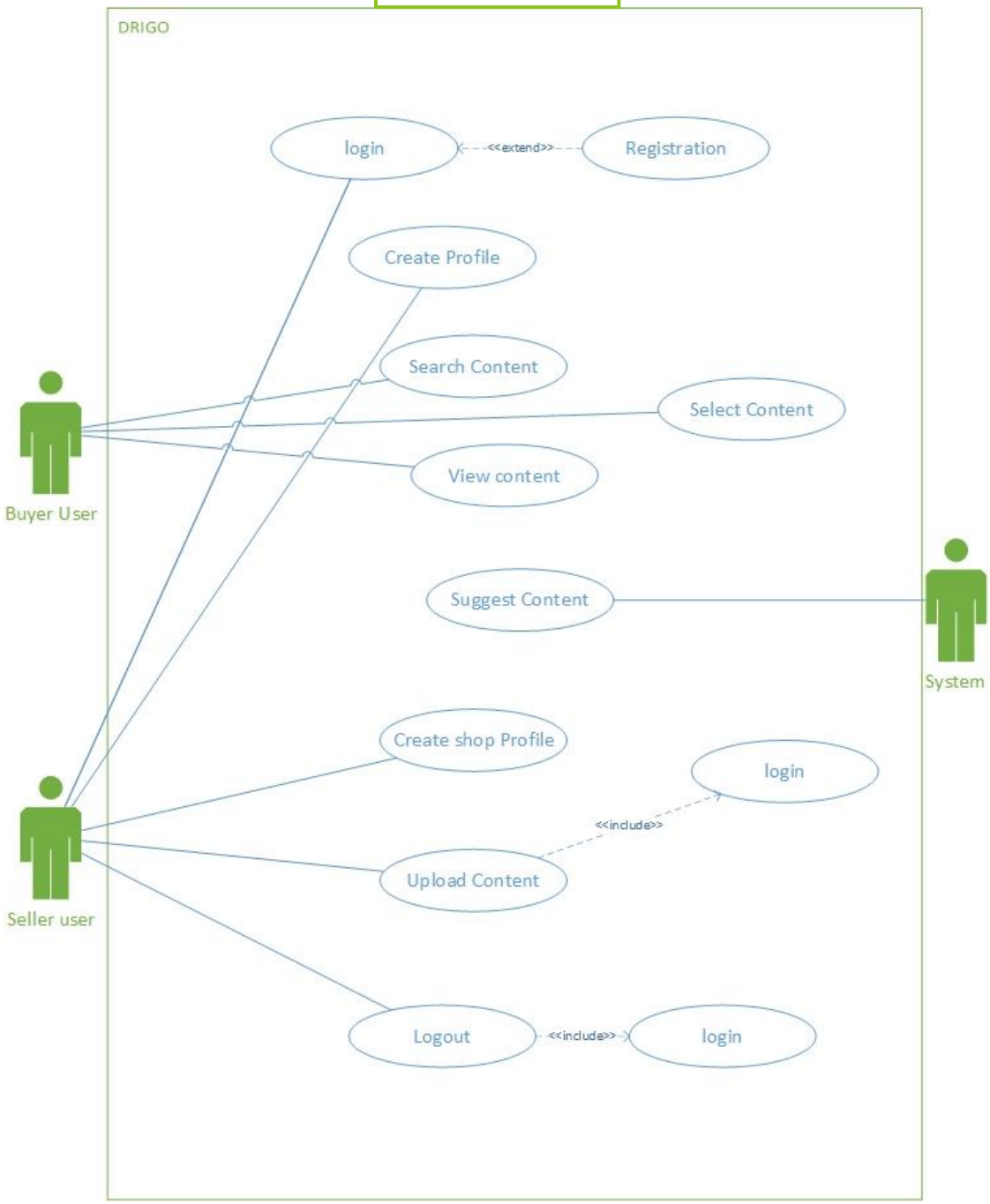
1.5.2 Non-Functional Requirement

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

<table>
<tr><td>Use Case</td><td>Registration</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>Create a new account in the system</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td></td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>A new user can to the register system</td></tr>
<tr><td>Failed End Condition
<the state of the world if goal abandoned</td><td>A new user can’t to the register system</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Seller user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td>Send a registration request</td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Using email address</td></tr>
<tr><td>1.1</td><td>Using username</td></tr>
<tr><td>1.2</td><td>Using a password its must be 8 characters</td></tr>
<tr><td>2</td><td>Submit the register request</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>1a</td><td>Invalid email address system show an invalid data message</td></tr>
<tr><td>2a</td><td>Invalid username system show an invalid data message</td></tr>
<tr><td>3a</td><td>Invalid password system show an remainder message</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>1</td><td>Submit the registration request time should be 5 s</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Use Case</td><td>Login</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>Log into the system with username and password</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td>User must register into the system</td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>User can logged into the system</td></tr>
<tr><td>Failed End Condition
<the state of the world if goal abandoned</td><td>User can’t logged into the system</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Seller user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td>Send a Login request</td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Enter Username</td></tr>
<tr><td>1.1</td><td>Enter password</td></tr>
<tr><td>2</td><td>Valid username and password</td></tr>
<tr><td>2</td><td>Allow access to the system</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>1a</td><td>Invalid username system show wrong username message</td></tr>
<tr><td>2a</td><td>Invalid password system show wrong password message</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>1</td><td>Login time should be 5s</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Use Case</td><td>Create profile</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>Make a Unique identity and craft start-up fir this online media</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td>User must be logged into the system</td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>User can upload their product</td></tr>
<tr><td>Failed End Condition
<the state of the world if goal abandoned</td><td>User can’t upload their product</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Seller user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td>A new profile create request comes in.</td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>User sends new profile to create request</td></tr>
<tr><td>1.1</td><td>User send a request in via Browser</td></tr>
<tr><td>2</td><td>Drigo can capture user’s First Name, Last Name, username, email, address, gender</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>1</td><td>Process response should be 5s</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Use Case</td><td>Suggest Content</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>System suggest content via a user location</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td></td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>Suggest content based on the nearest location</td></tr>
<tr><td>Failed End Condition
<the state of the world if goal abandoned</td><td>System don’t suggest content based on the nearest location</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Buyer user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td></td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>System researching with the user location</td></tr>
<tr><td>1.1</td><td>System suggest nearest content/shop</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>1</td><td>If the system can’t get the user’s location then it can’t suggest the nearest content/shop</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>1</td><td>Suggest content time should be 5s</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Use Case</td><td>Search Content</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>Find out a specific contnent</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td></td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>User can find specific content easily</td></tr>
<tr><td>Failed End Condition
<the state of the world if goal abandoned</td><td>User can’t find specific content easily</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Buyer user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td>Call search request</td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Input content or users information</td></tr>
<tr><td>1.1
1.2</td><td>Submit search request with the information
System show the output or specific thinks which you want</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>1</td><td>The finding information out of the server</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>1</td><td>Search time should be 2s</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Use Case</td><td>Select Content</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>Select a specific content</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td></td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>User can select specific content</td></tr>
<tr><td>Failed End Condition
<the state of the world if goal abandoned</td><td>User can select specific content</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Buyer user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td>Click the menu button</td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Go to the category list</td></tr>
<tr><td>1.1</td><td>Select the specific content</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>1.1a</td><td>Finding information out of the category</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>1</td><td>select time should be 2s</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Use Case</td><td>Shop Profile</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>Create shop Profile</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td></td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>Users can make their own shop profile.</td></tr>
<tr><td>Failed End Condition
<the state of the world if the goal abandoned</td><td>The user can’t make his shop profile.</td></tr>
<tr><td>Primary Actors:

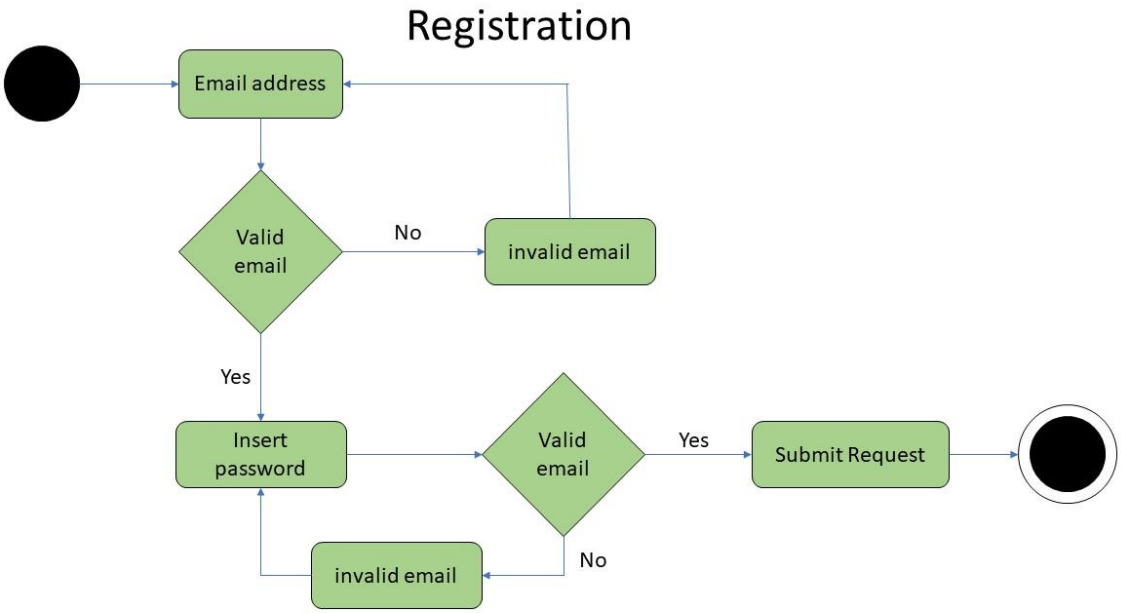
Secondary Actors:</td><td>Buyer user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td>Send a Request to make a new shop profile</td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td><table><tr><th>Step</th><th>Action</th></tr><tr><td>1</td><td>Input shop name</td></tr><tr><td>1.1</td><td>Input trad license number</td></tr></table></td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td><table><tr><th>Step</th><th>Branching Action</th></tr></table></td></tr>
<tr><td>Quality Requirements</td><td><table><tr><th>Step</th><th>Requirement</th></tr></table></td></tr>
</table>

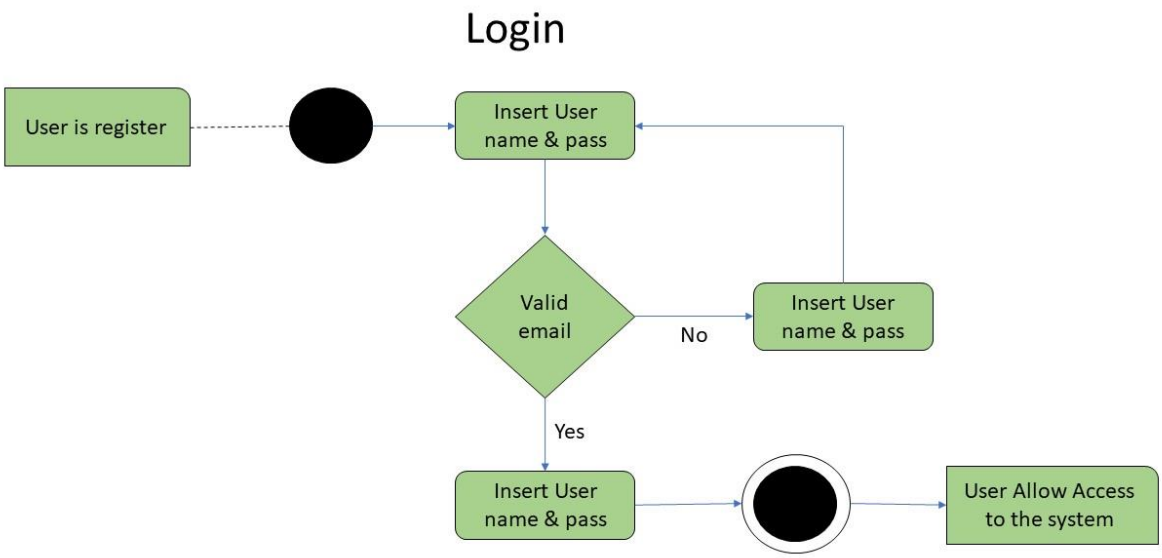
Use Case	Upload content
Goal <a longer statement of the goal in context if needed>	Users can upload a new product information
Preconditions <what we expect is already the state of the world>	Users must log into the system
Success End Condition <the state of the world if goal abandoned>	Users can upload 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**

<table>
<tr><td>Use Case</td><td>Logout</td></tr>
<tr><td>Goal
<a longer statement of the goal in context if needed></td><td>Logout into the system</td></tr>
<tr><td>Preconditions
<what we expect is already the state of the world></td><td>The system must logged into the system</td></tr>
<tr><td>Success End Condition
<the state of the world if goal abandoned></td><td>Users can logout into the application</td></tr>
<tr><td>Failed End Condition
<the state of the world if the goal abandoned</td><td>The user can’t logout into the application</td></tr>
<tr><td>Primary Actors:

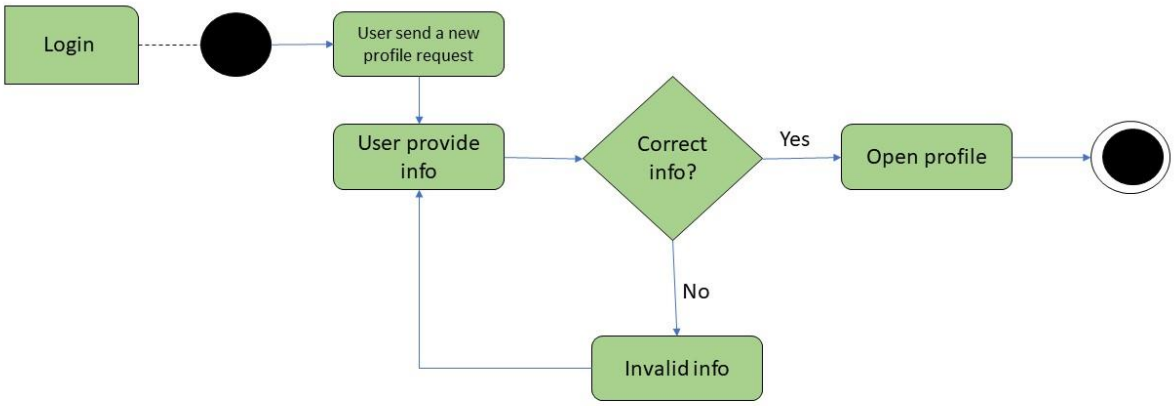
Secondary Actors:</td><td>Buyer user</td></tr>
<tr><td>Trigger
<the action upon the system that start use case></td><td>Send logout request</td></tr>
<tr><td>Description/ Main Success Scenario
<the steps of the scenario from trigger to goal delivery and any clean-up after></td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select logout</td></tr>
<tr><td>1.1</td><td>Submit a logout request it the system</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows
<a: condition causing branching>
<al: action or name of sub-use case></td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>
<table>
<tr><th>Step</th><th>Requirement</th></tr>
<tr><td>1</td><td>Logout time should be 5s</td></tr>
</table>
</td></tr>
</table>

1.8 Activity Diagram

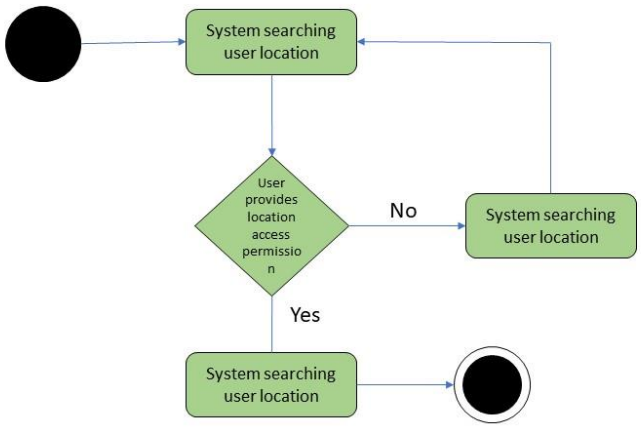




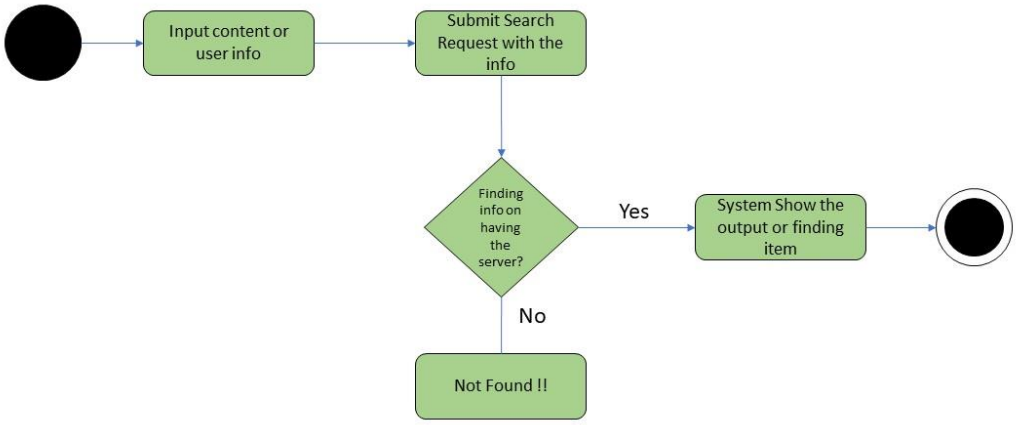
Create Profile



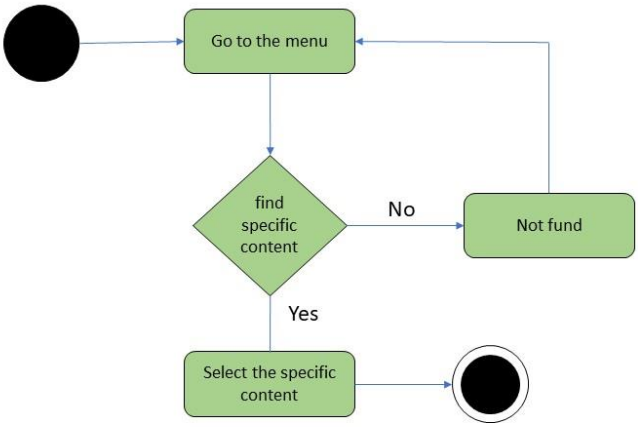
Suggest Content



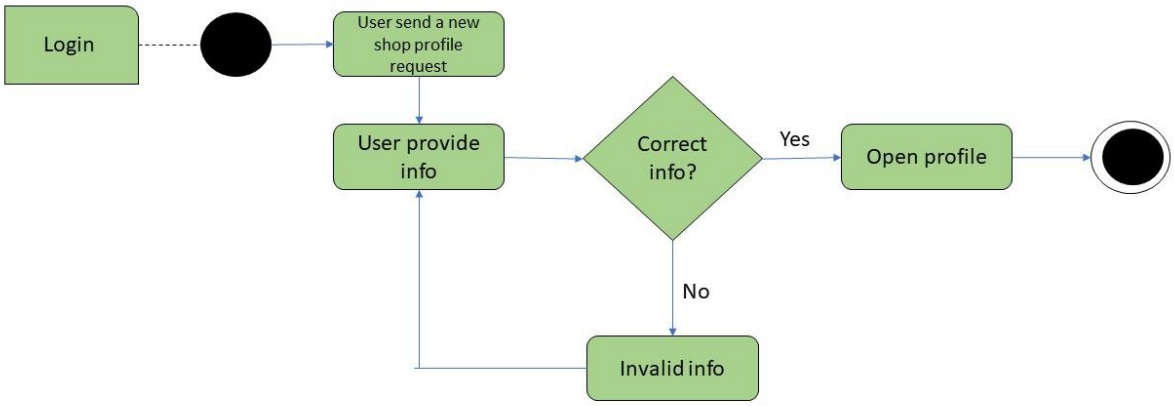
Search Content



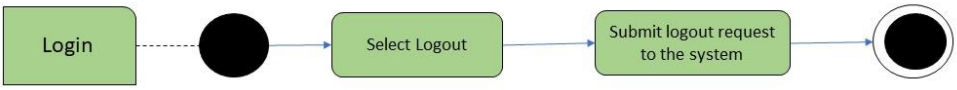
Select Content



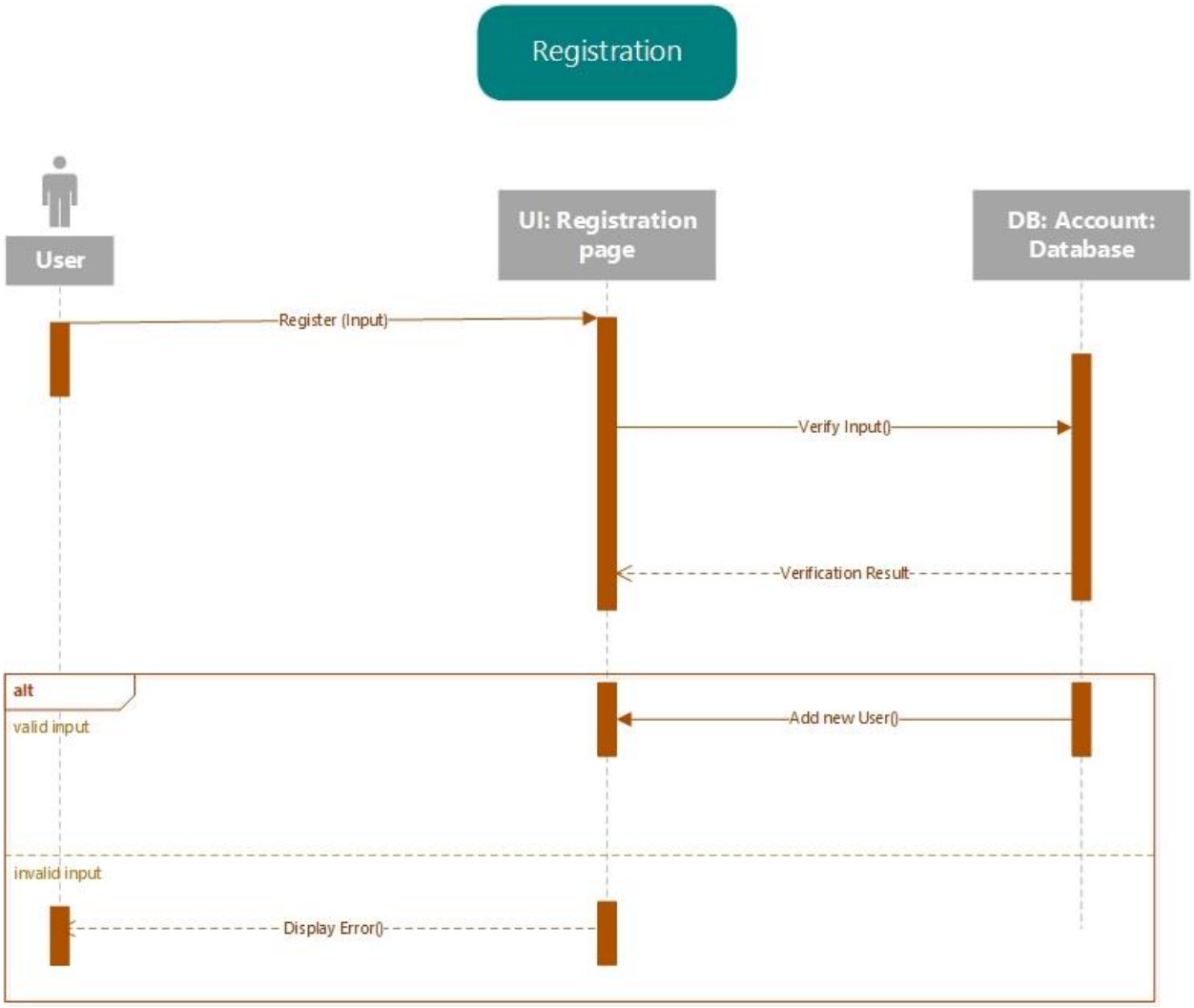
Create shop profile

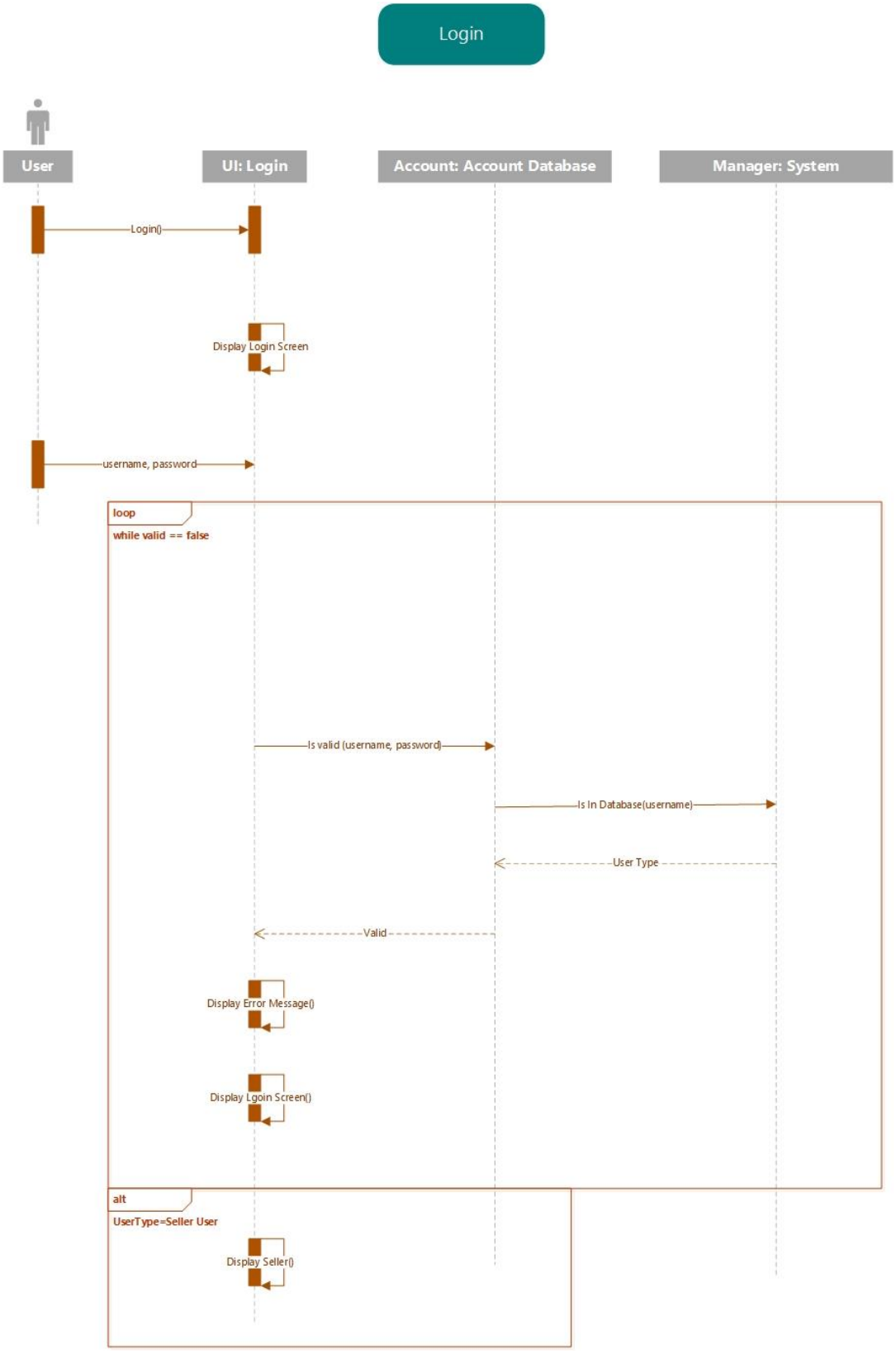


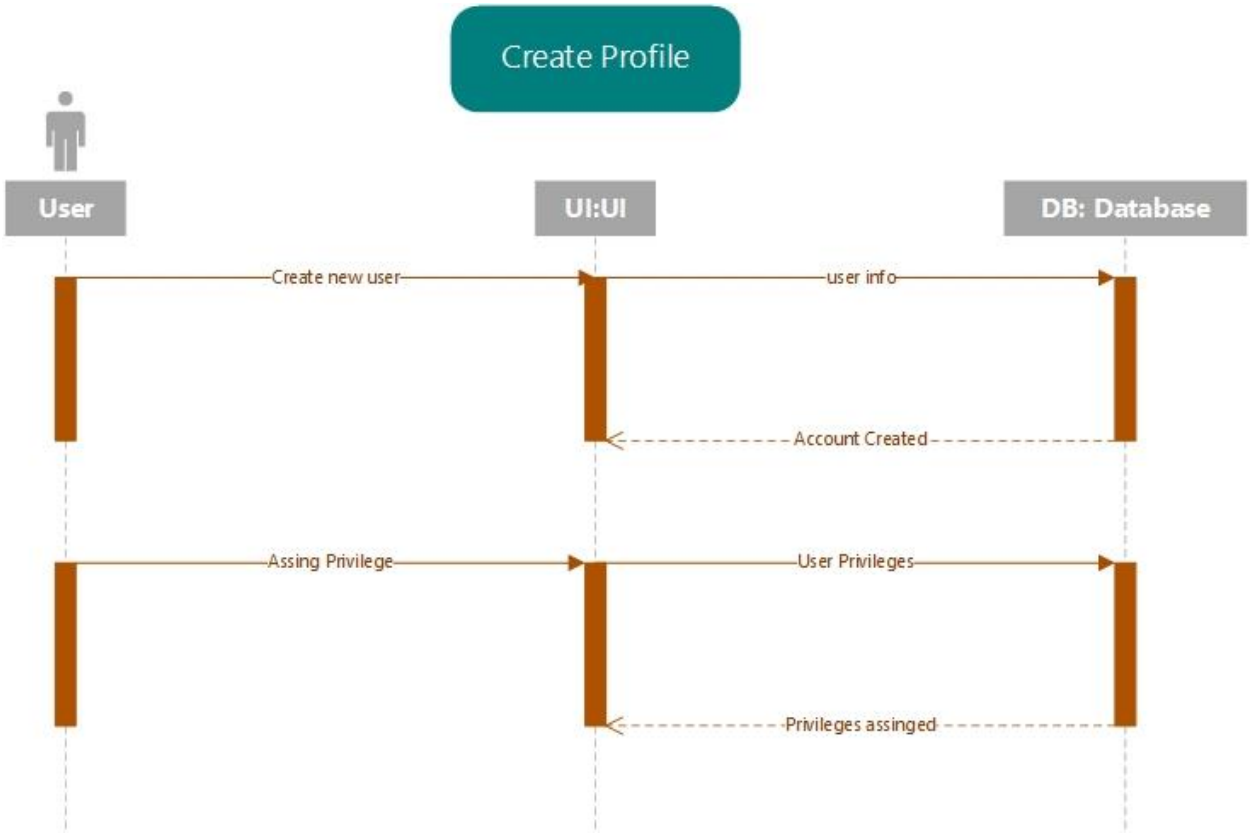
Logout

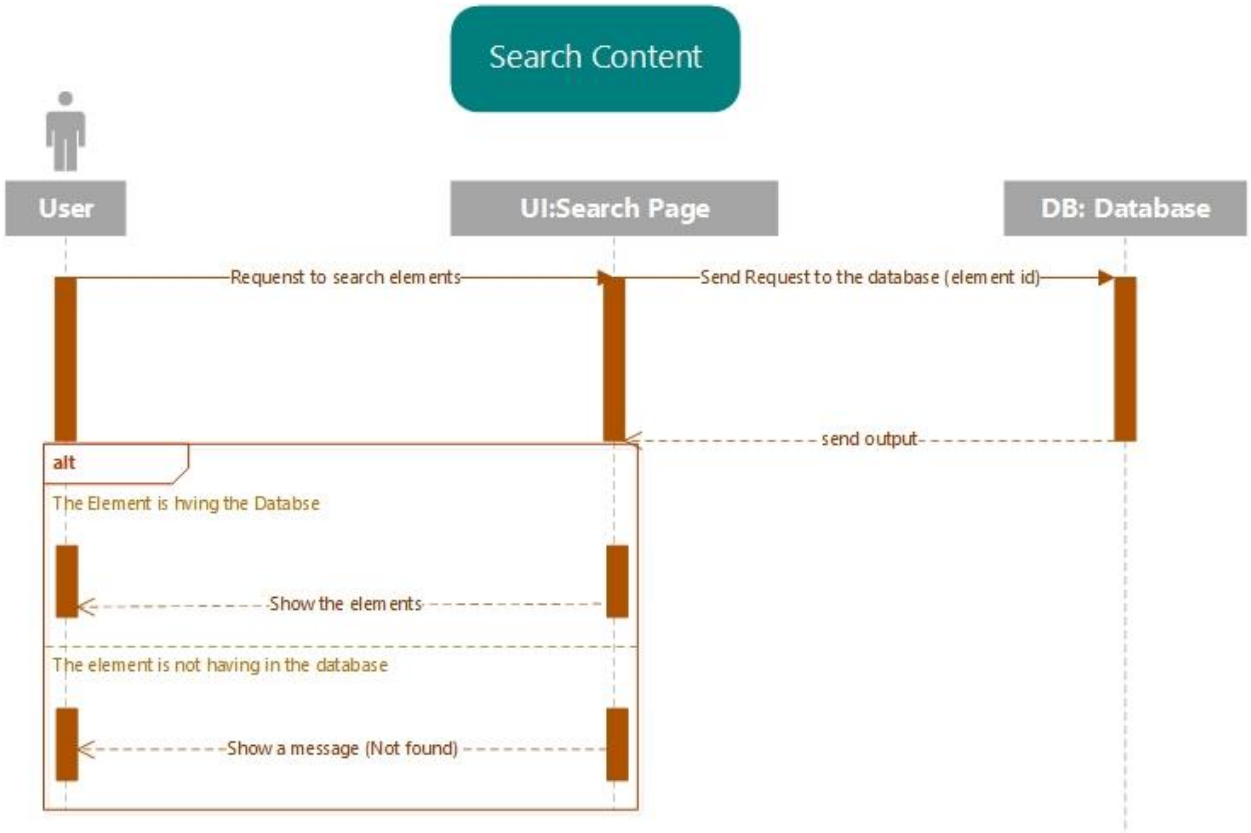


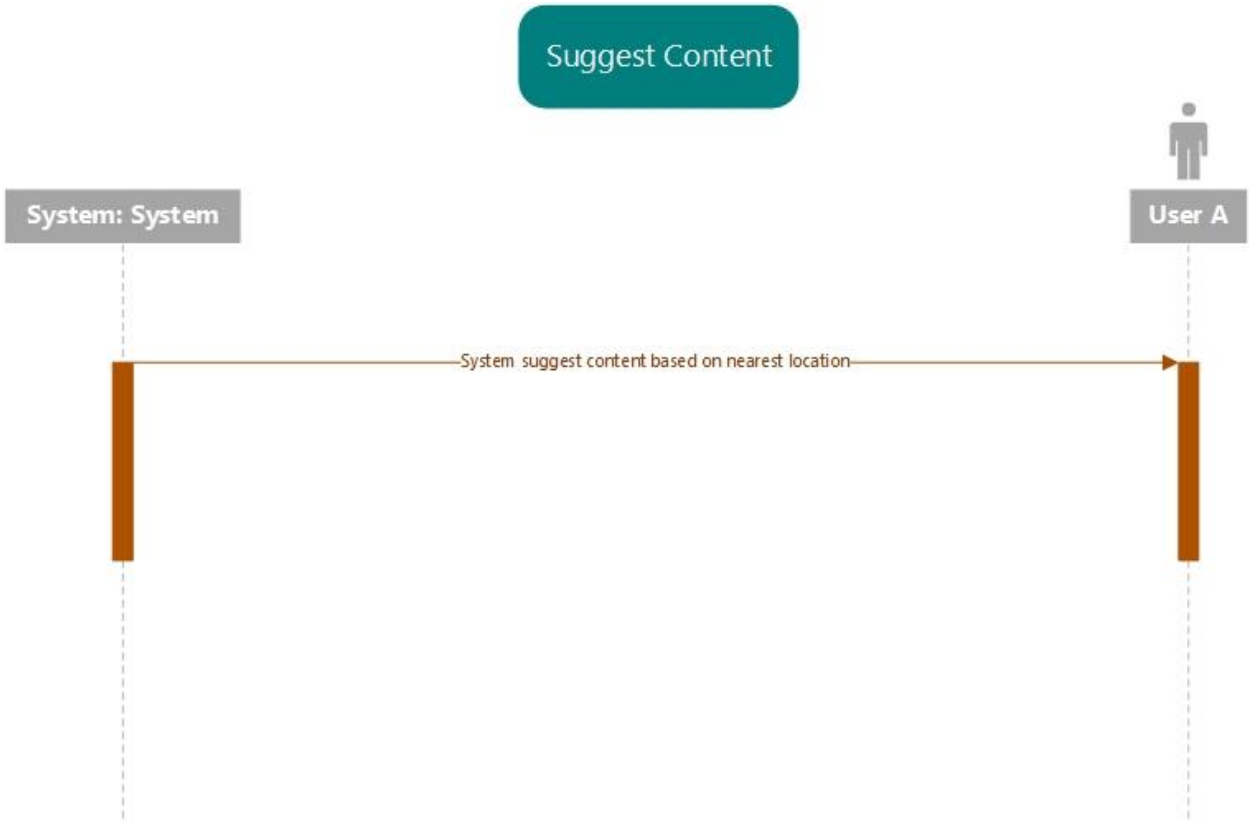
1.9 Sequence Diagram

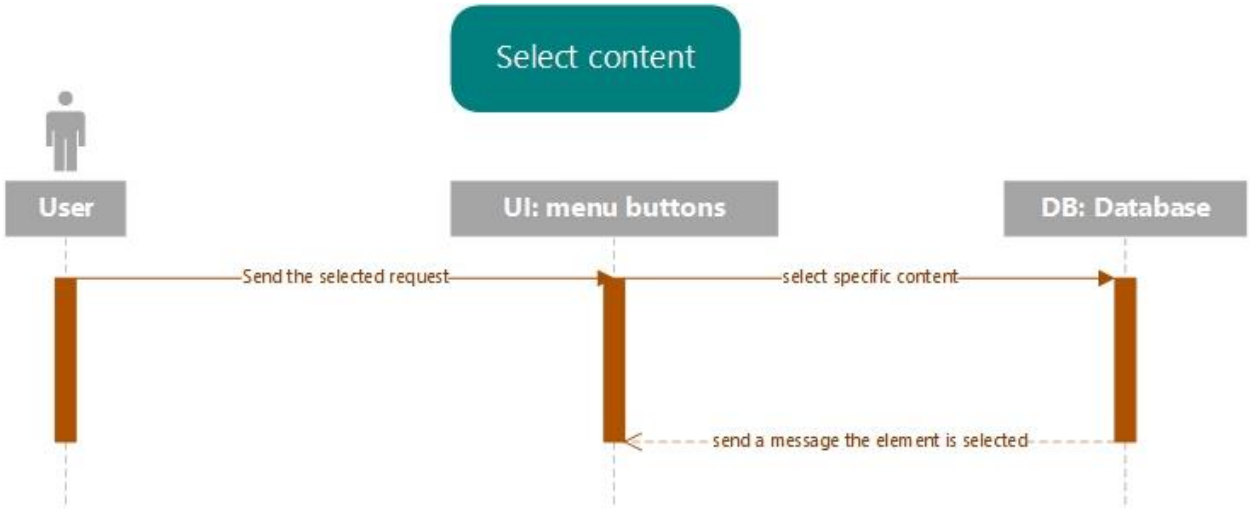


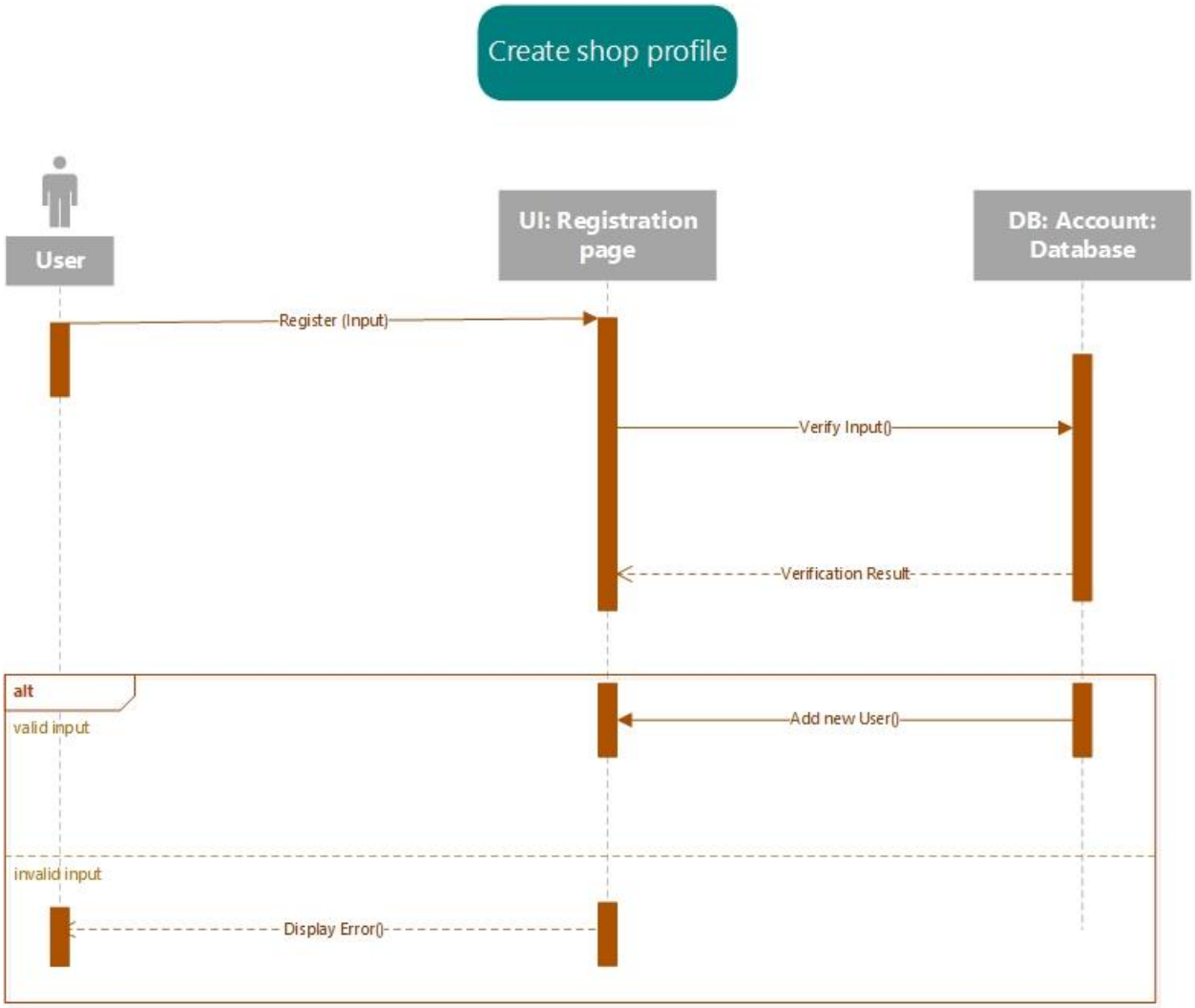

Login
User
UI: Login
Account: Account Database
Manager: System
Login()
Display Login Screen
username, password
loop
while valid == false
Is valid (username, password)
Is In Database(username)
User Type
Valid
Display Error Message()
Display Lgoin Screen()
alt
UserType=Seller User
Display Seller()

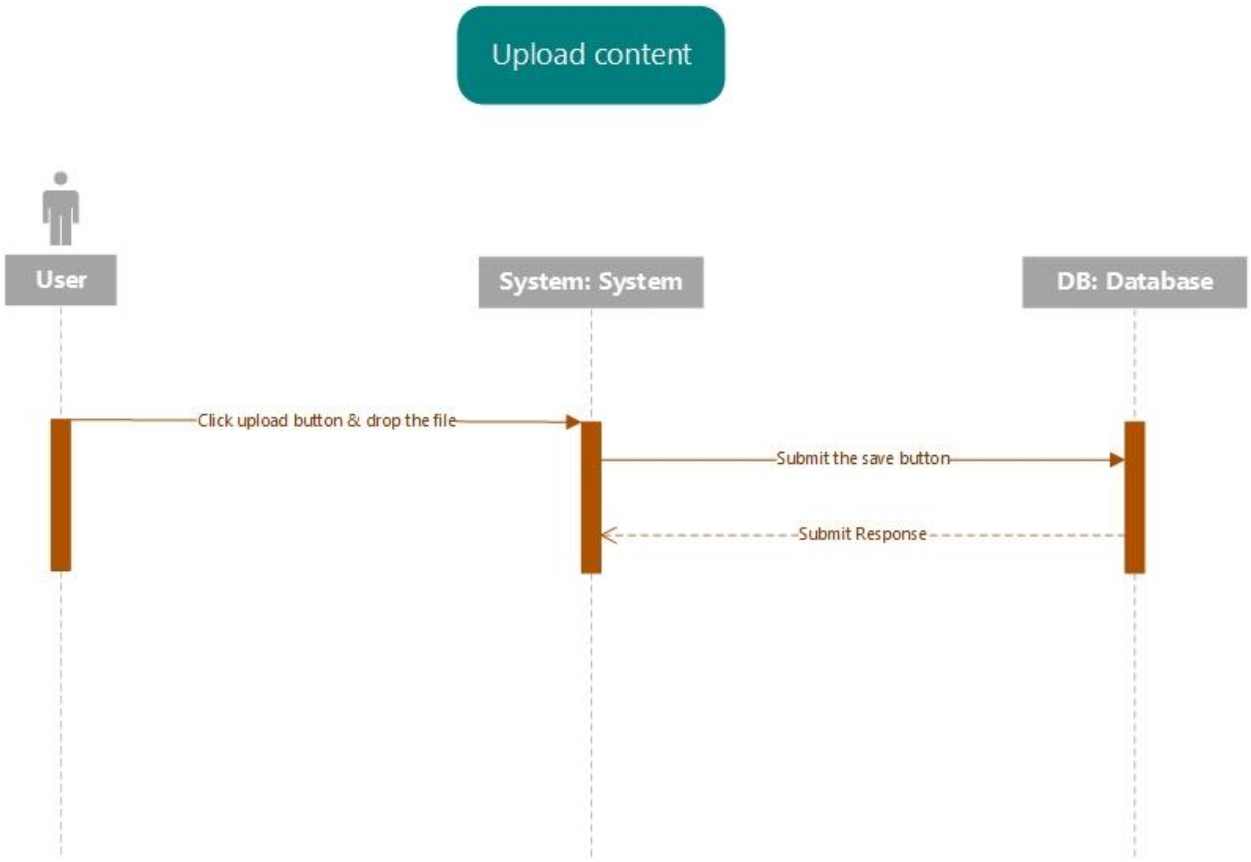

Create Profile
User
UI:UI
DB: Database
Create new user
user info
Account Created
Assing Privilege
User Privileges
Privileges assinged


Search Content
User
UI:Search Page
DB: Database
Requenst to search elements
Send Request to the database (element id)
send output
alt
The Element is hving the Databse
Show the elements
The element is not having in the database
Show a message (Not found)

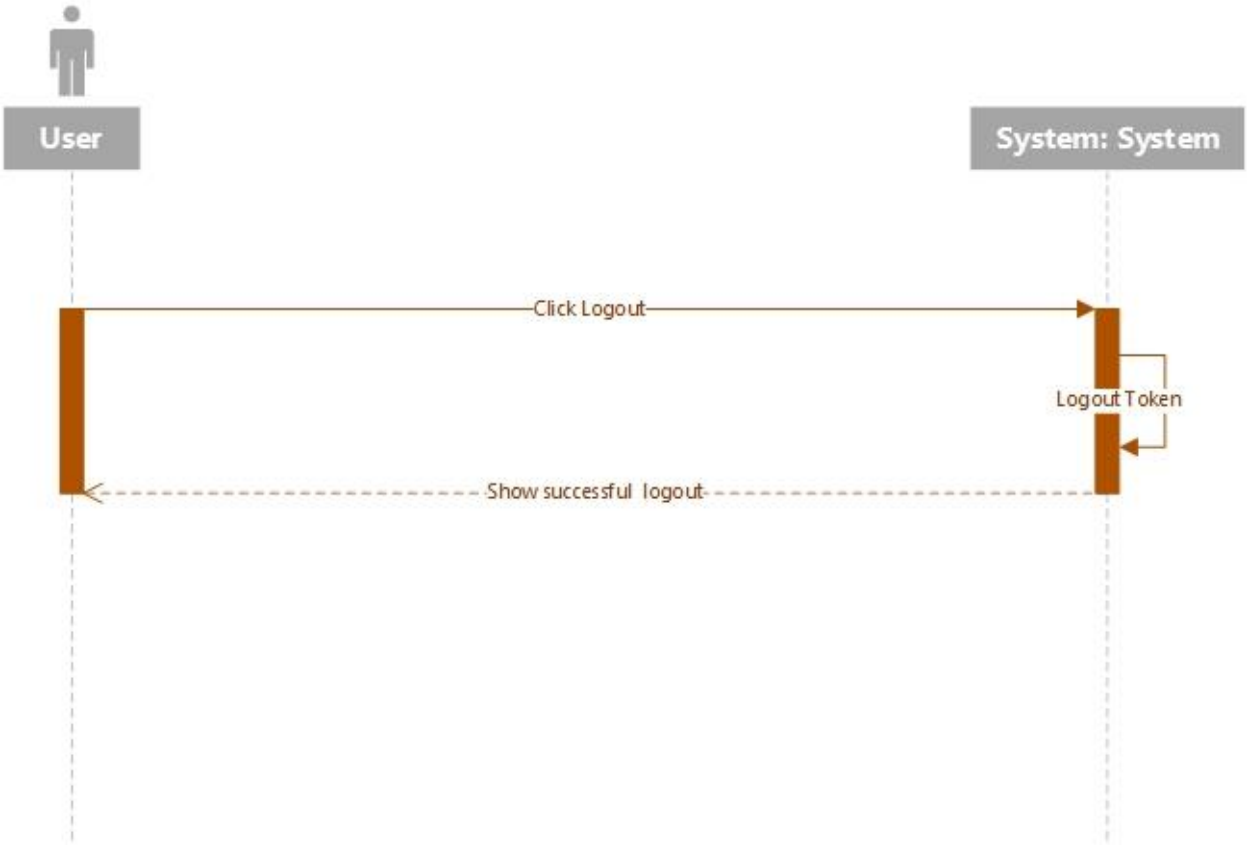

Suggest Content
System: System
User A
System suggest content based on nearest location


Select content
User
UI: menu buttons
DB: Database
Send the selected request
select specific content
send a message the element is selected


Create shop profile
User
UI: Registration page
DB: Account: Database
Register (Input)
Verify Input()
Verification Result
alt
valid input
Add new User()
invalid input
Display Error()


Upload content
User
System: System
DB: Database
Click upload button & drop the file
Submit the save button
Submit Response

Logout


User
System: System
Click Logout
Logout Token
Show successful logout