

*SiteOcation*

*A Website for every site*

## *AUTHORS*

*This document was prepared by:*

<i>Ardavan Gharebaghi Urmia University</i>	<i>Mahdi Rezazade Urmia University</i>	<i>Parsa Mahmoodi Urmia University</i>
--	--	--

## *VERESION HISTORY*

<i>Date</i>	<i>Document Version</i>	<i>Document Revision History</i>	<i>Document Author</i>
<i>01/26/2021</i>	<i>1.0</i>	<i>Initial Draft</i>	<i>- Ardavan Gharebaghi - Mahdi Rezazade - Parsa Mahmoodi</i>

## Table of Contents

<b>Project Definition</b> .....	4
<b>Architecture</b> .....	7
<b>Problem Statement</b> .....	8
<b>Actors List</b> .....	9
<b>Use Cases List</b> .....	10
<b>Use Case Model Diagrams</b> .....	12
Actors Diagram.....	13
Automatic Functions.....	14
Member Functions .....	15
Profile Subsystem Diagram .....	16
<b>Use Case Scenarios</b> .....	17
User Register Use Case Scenario .....	17
Login Use Case Scenario .....	18
Like Comments Use Case .....	19
Add Comment Use Case .....	20
Dislike Comment Use Case.....	21
Search Websites Use Case .....	22
Create Websites Use Case .....	23
Edit Profile Use Case .....	24
Logout Use Case .....	25
<b>Sequence Diagrams</b> .....	26
Sequence Diagram for User Register Use case .....	26
Sequence Diagram for Member Login Use case .....	27
Sequence Diagram for Comment Use case.....	28
<b>Activity Diagrams</b> .....	29
Activity Diagram for User Register Use Case .....	29
Activity Diagram for Member Login Use Case.....	30
Activity Diagram for Comment Use Case.....	31
<b>System Diagram</b> .....	32

## Project Definition

### 1. PURPOSE

*This Document provides a brief overview of Siteocation to promote a shared understanding of it.*

### 2. PROBLEM

*People uses websites daily for their jobs, shopping online and etc. but how they can find the best website for there needs? What if they want to comment their opinion about those websites and what if some other people wants to know those opinions?*

### 3. PROJECT GOAL

*Developing a useful, user-friendly website that allows people to comment about other websites so that other people can benefit from those opinions and choose best website for themselves.*

### 4. PROJECT OBJECTIVES

*Developing a website that:*

- Has a large archive of the websites, their domain address and other useful information*
- Attracts both users and websites*

### 5. KEY STAKEHOLDERS

NAME	DESCRIPTION	RESPONSIBILITIES
Company Management	The owner of the product	Provides the required information about product and approves it
Project Manager	Leads development of the project	Plans, manages, allocates resources, decides priorities, coordinates interactions with customers and users

<i>Designer</i>	<i>The primary lead in the project</i>	<i>Responsible for overall architecture of the product</i>
<i>Developers</i>	<i>Programmers responsible for developing the product</i>	<i>Develop the system</i>
<i>Maintainers</i>	<i>Programmers responsible for developing the product</i>	<i>Maintaining the product after it is released</i>
<i>Support Staff</i>	<i>Employees that will provide support for the product</i>	<i>Provide support to customers and users</i>
<i>Users</i>	<i>User of the product</i>	<i>Using product</i>

#### 6. OUTCOMES/SUCCESS

*The product should reach the desired number of user-base, archive and content in the given time in order to succeed and provide profit for the owners.*

#### 7. ASSUMPTION AND CONSTRAINTS

##### *i. Assumptions*

- a. Users have access to internet*
- b. Users have basic knowledge about using websites*

##### *j. Constraints*

- a. Should be simple and easy to use*

#### 8. RISKS

*There is great chance of product failure in cases of:*

- Not being useful for a large amount of internet users*
- Getting denied by companies and other websites*

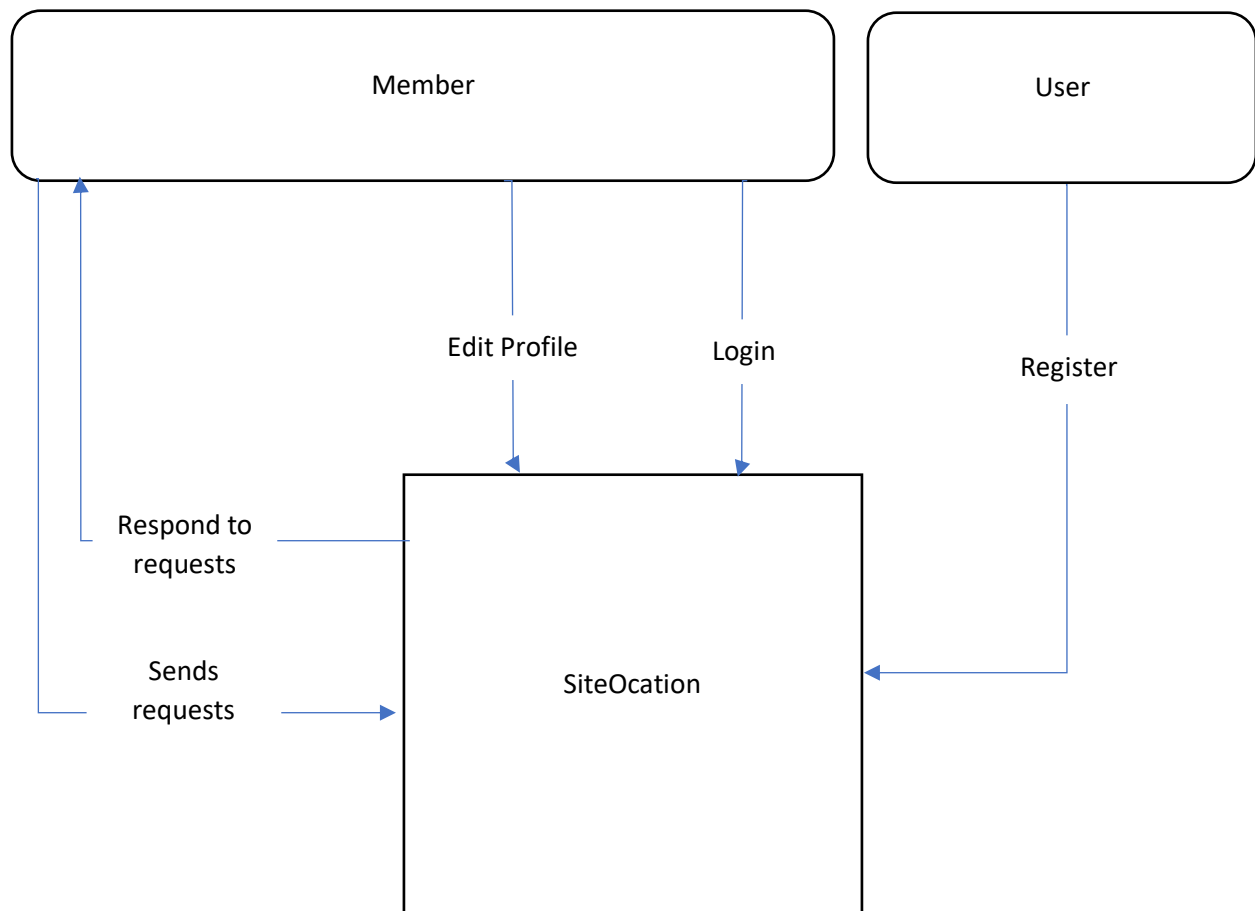
9. *ESTIMATED COST*  
*Not defined yet*

10. *ESTIMATED DURATION*

*DATE:* *01/19/2021*

*Project Manager:* *Mahdi Rezazade*

## Architecture



*This project will be written by DJANGO Framework and will use MVC type of architecture.*

- **Benefits:**  
*This method will give development team an easier approach and familiar approach to the project development process.*
- **Disadvantages:**  
*Development team are not familiar with this methodology.*

## Problem Statement

<i>Statement</i>	<i>Urgency</i>	<i>Solution</i>
<i>Time between a user request and system response should be under 1 second.</i>	<i>High</i>	<i>Better Server</i>
<i>System should be able to provide service to 1000 users at same time</i>	<i>Medium</i>	<i>More Server</i>



## Actors List

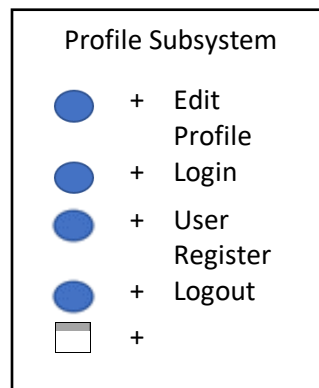
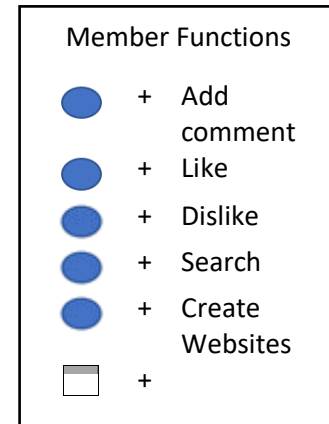
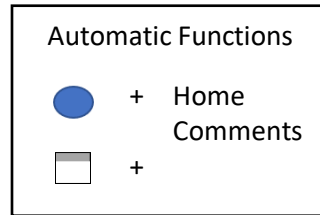
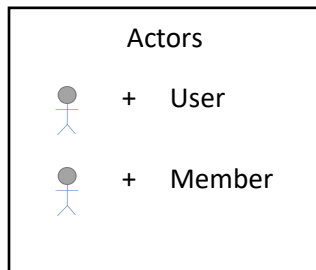
<i><b>Name</b></i>	<i><b>Description</b></i>
<i>User</i>	<i>People who will use the websites</i>
<i>Member</i>	<i>People who will use the product as a consumer / Websites</i>

## Use Cases List

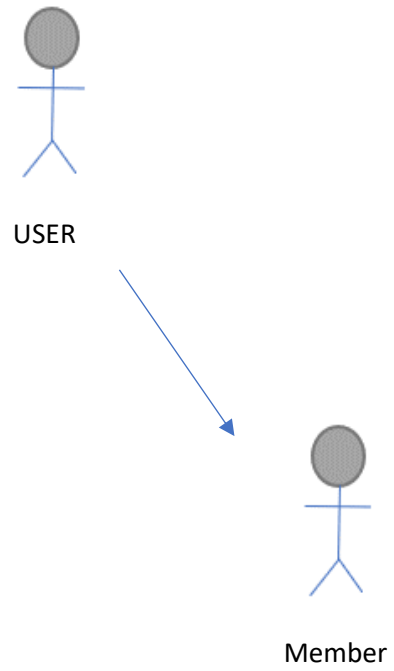
<i>Use Case</i>	<i>Description</i>	<i>Actors Involved</i>
<i>User Register</i>	<i>A new Member registers by using email</i>	<i>User</i>
<i>Login</i>	<i>An existing Member logs in by email</i>	<i>Member</i>
<i>Edit Profile</i>	<i>An existing Member edits details in their profile</i>	<i>Member</i>
<i>Home comments</i>	<i>When a User opens website, first thing they will see</i>	<i>Member / User</i>
<i>Comment</i>	<i>A Member comments</i>	<i>Member</i>
<i>Like</i>	<i>Members likes other comments</i>	<i>Member</i>
<i>Search</i>	<i>Member searches websites</i>	<i>Member / User</i>
<i>Dislike</i>	<i>Member can follow comments of a website</i>	<i>Member</i>
<i>Create Websites</i>	<i>Members can create websites profile</i>	<i>Member</i>

<i>Logout</i>	<i>Members can log out</i>	<i>Member</i>
---------------	----------------------------	---------------

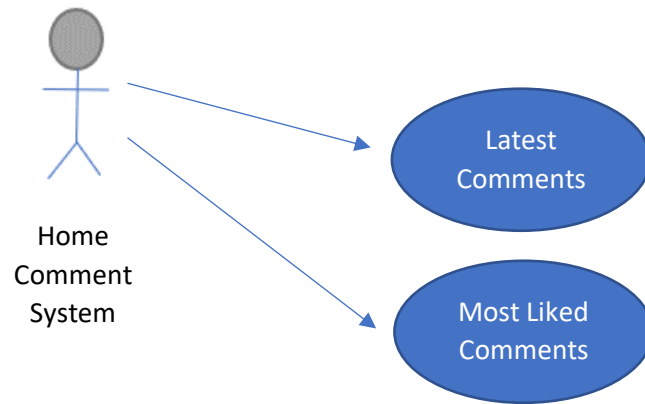
## Use Case Model Diagrams



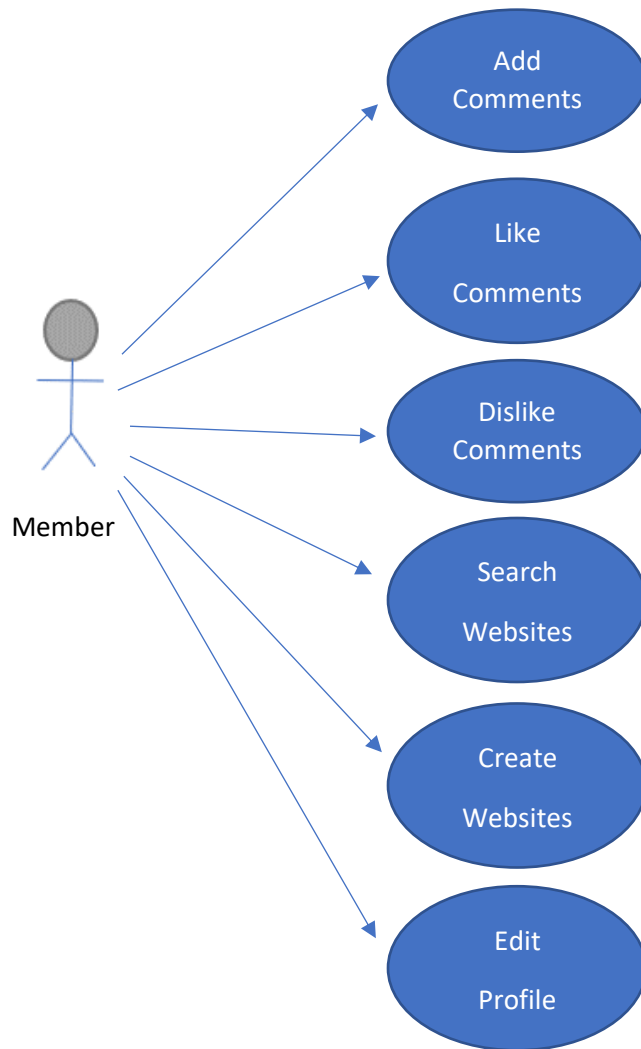
## Actors Diagram



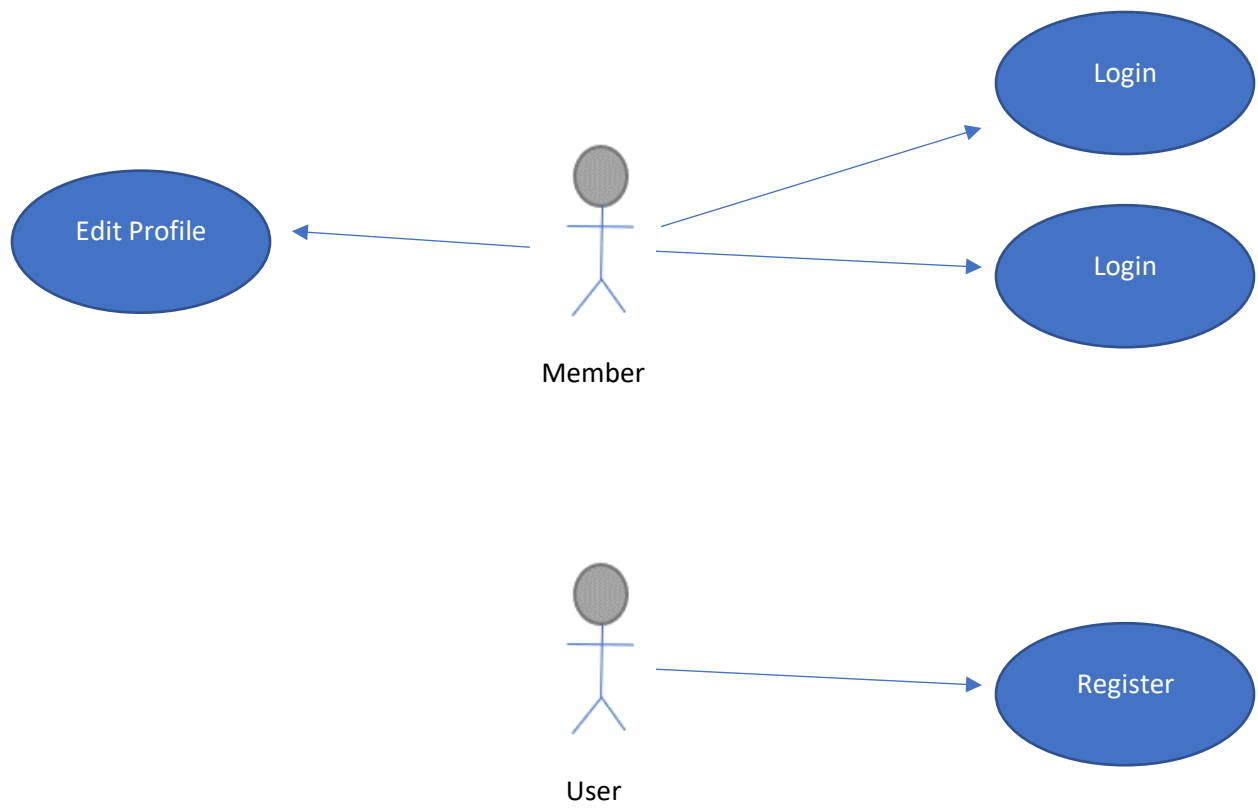
## Automatic Functions



## Member Functions



## Profile Subsystem Diagram





## Use Case Scenarios

### User Register Use Case Scenario

Use Case Name	User Register	
Use Case ID	01	
Priority	High	
Primary Actor	User	
Other Actors	-	
Description	A new User registers by email	
Precondition	User should have email	
Trigger	User clicking button	
Course of Event	Actor Action	System Response
	Step 1: The User selects registration  Step 3: The user selects completion of registration.	Step 2: The System responds by displaying registration form  Step 4: The system confirms the info  Step 5: The system informs user of registration completion
Conclusion	This use case concludes when user is informed of registration completion	
Postcondition	User can log in now	
Business Rules	Member should have valid E-mail address	
Implementation Constraints and Specifications	Use case must be available all the time It should support 100 concurrent requests	
Assumptions	The member can cancel registration at any time	
Open Issues	None	

## Login Use Case Scenario

Use Case Name	Login	
Use Case ID	02	
Priority	High	
Primary Actor	User	
Other Actors	-	
Description	This use case describes the event of a user logging in. If login was successful he/she will be logged in to his/her account.	
Precondition	User should already have an account.	
Trigger	This use case is initiated when user selects the option to login.	
Course of Events	Actor Action	System Response
	Step 1: The user requests to login  Step 3: The user fills the forms and presses login	Step 2: The system responds by displaying the form  Step 4: System confirms the info and logs user in
Conclusion	This use case concludes when member is informed of login completion	
Postcondition	The user will be granted the permissions he/she has	
Business Rules	None	
Implementation Constraints and Specifications	Use case must be available all the time It should support 100 concurrent requests	
Assumptions	The member can cancel login process at any time	
Open Issues	None	

## Like Comments Use Case

Use Case Name	Like Comment	
Use Case ID	03	
Priority	High	
Primary Actor	Member	
Other Actors	-	
Description	A Member can like comments of a websites	
Precondition	Members should be logged in	
Trigger	Member	
Course of Event	Actor Action	System Response
	Step 1: The Member selects comment like button and request to like	Step 2: The System responds by showing the liked colored button
Conclusion	This use case concludes when user is informed of like process completed	
Postcondition	Member can unlike now	
Business Rules	Member should be logged in	
Implementation Constraints and Specifications	Use case must be available all the time It should support 500 concurrent requests	
Assumptions	Member haven't disliked	
Open Issues	None	

## Add Comment Use Case

Use Case Name	Add Comment	
Use Case ID	04	
Priority	High	
Primary Actor	Member	
Other Actors	-	
Description	A Member can add comments for a websites	
Precondition	Members should be logged in	
Trigger	Member	
Course of Event	Actor Action	System Response
	Step 1: The Member enters comment in its section  Step 2: The Member selects comment add button and request to submit the comment	Step 3: The System responds by showing the liked color  Step 4: The comment section of a websites gets update
Conclusion	This use case concludes when user is informed of submission process completed	
Postcondition	-	
Business Rules	Member should be logged in	
Implementation Constraints and Specifications	Use case must be available all the time It should support 500 concurrent requests	
Assumptions	The member can cancel commenting at any time	
Open Issues	None	

## Dislike Comment Use Case

Use Case Name	Dislike Comment	
Use Case ID	05	
Priority	High	
Primary Actor	Member	
Other Actors	-	
Description	A Member can dislike comments of a websites	
Precondition	Members should be logged in	
Trigger	Member	
Course of Event	Actor Action	System Response
	Step 1: The Member selects comment dislike button and request to like	Step 2: The System responds by showing the disliked colored button
Conclusion	This use case concludes when user is informed of dislike process completed	
Postcondition	Member can un-dislike now	
Business Rules	Member should be logged in	
Implementation Constraints and Specifications	Use case must be available all the time It should support 500 concurrent requests	
Assumptions	Member haven't liked	
Open Issues	None	

## Search Websites Use Case

Use Case Name	Search	
Use Case ID	06	
Priority	High	
Primary Actor	User	
Other Actors	Member	
Description	A User can search specific websites	
Precondition	-	
Trigger	User	
Course of Event	Actor Action	System Response
	Step 1: The user requests to search by pressing search button	Step 2: The system responds by displaying the search bar
	Step 3: The user fills the bar and presses search button	Step 4: System processes the request and shows the relevant websites
Conclusion	This use case concludes when user is being shown the website page	
Postcondition	-	
Business Rules	-	
Implementation Constraints and Specifications	Use case must be available all the time It should support 500 concurrent requests	
Assumptions	-	
Open Issues	None	

## Create Websites Use Case

Use Case Name	Create Website	
Use Case ID	07	
Priority	High	
Primary Actor	Member	
Other Actors	-	
Description	A Member can add new websites	
Precondition	Members should be logged in	
Trigger	Member	
Course of Event	Actor Action	System Response
	<p>Step 1: The Member selects add new website option</p> <p>Step 3: The member fills the forms and submit</p>	<p>Step 2: The System responds by showing the specific form</p> <p>Step 4: The System gets the form and responds the compilation message</p> <p>Step 5: After the confirmation of details, a new website will be added and informed the member who added it by email</p>
Conclusion	This use case concludes when user is informed of the final results by email	
Postcondition	-	
Business Rules	Member should be logged in	
Implementation Constraints and Specifications	<p>Use case must be available all the time</p> <p>It should support 500 concurrent requests</p>	
Assumptions	Member is logged in	
Open Issues	None	

## Edit Profile Use Case

Use Case Name	Edit Profile	
Use Case ID	08	
Priority	High	
Primary Actor	Member	
Other Actors	-	
Description	This use case describes the act of setting (or changing) the avatar or the profile description of a Member. The updated data must be uploaded to the company servers, replacing the old data	
Precondition	Members should be logged in	
Trigger	This use case is initiated when the user chooses the “Edit Profile” option	
Course of Event	Actor Action	System Response
	<p>Step 1: The Member chooses the “Edit Profile” option</p> <p>Step 3: The Member fills in the fields</p> <p>Step 5: The Member saves the changes</p>	<p>Step 2: The System opens editable fields for the avatar and the profile description</p> <p>Step 4: The System checks the fields for pre-defined offensive content</p> <p>Step 6: The System loads the new data from the user’s device, replacing the old data in the database(s)</p>
Conclusion	This use case is concluded when the new data is uploaded to The System	
Postcondition	Profile data must be updated in The Member’s device	
Business Rules	Member should be logged in	
Implementation Constraints and Specifications	<p>Use case must be available all the time</p> <p>It should support 500 concurrent requests</p>	
Assumptions	<p>Member is logged in</p> <p>Members can edit the profile data unlimited times</p>	
Open Issues	Which languages must the app support?	

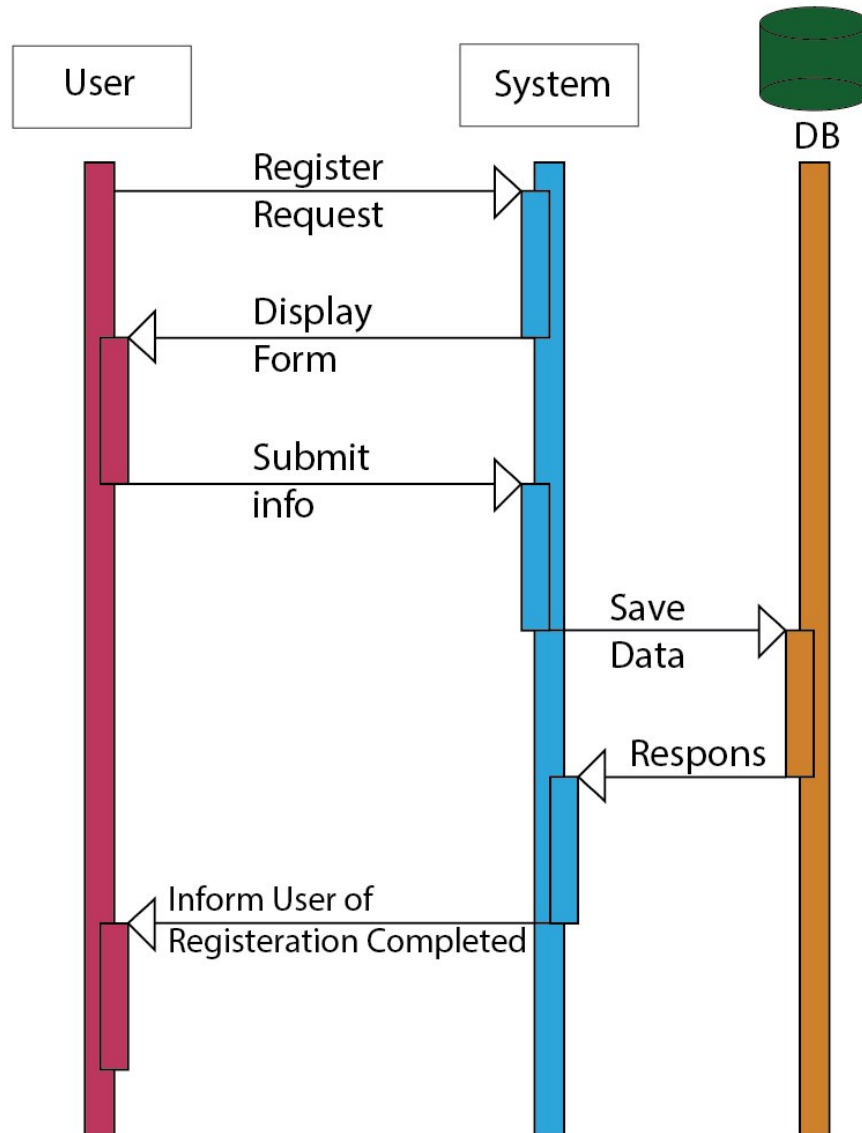


## Logout Use Case

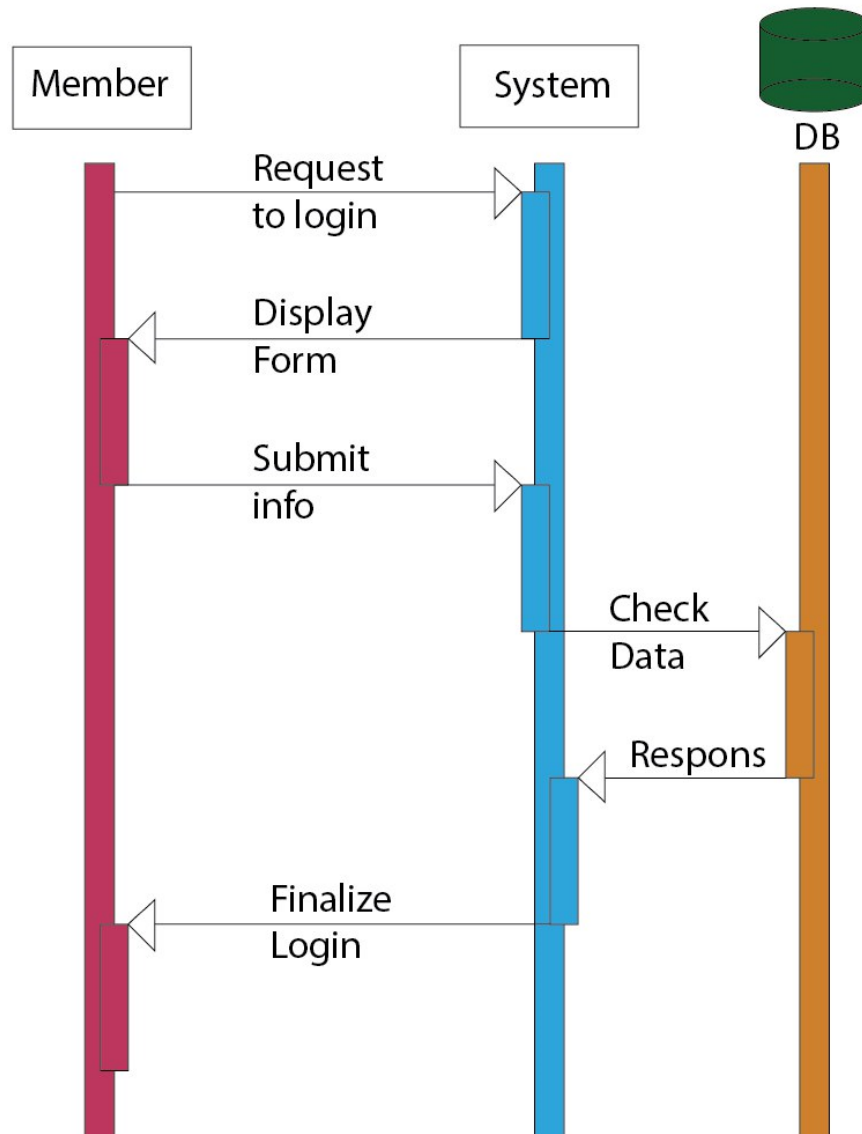
Use Case Name	Logout	
Use Case ID	09	
Priority	High	
Primary Actor	Member	
Other Actors	-	
Description	This use case describes the event of a Member loges out of website.	
Precondition	Member should already have been logged in.	
Trigger	This use case is initiated when Member selects the option to logout.	
Course of Events	Actor Action	System Response
	Step 1: The Member requests to logout	Step 2: The system responds by logging out the user  Step 3: System will inform Member that logging out was successful
Conclusion	This use case concludes when Member is informed of logout completion	
Postcondition	All Member permissions and access will switch to User ones	
Business Rules	None	
Implementation Constraints and Specifications	Use case must be available all the time It should support 500 concurrent requests	
Assumptions	-	
Open Issues	None	

## Sequence Diagrams

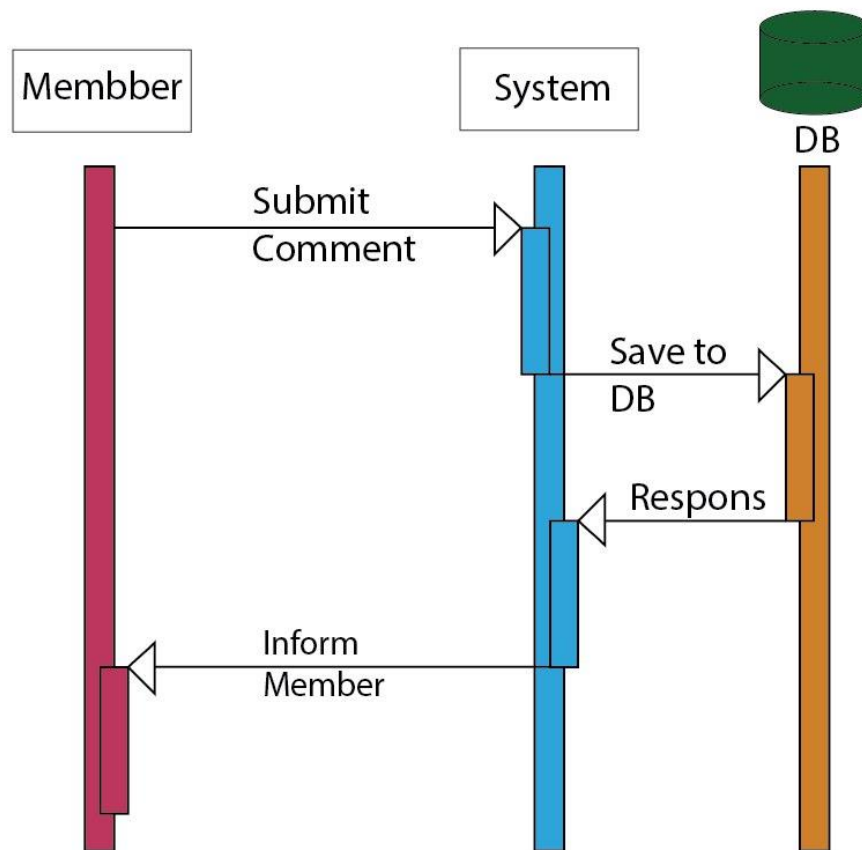
Sequence Diagram for User Register Use case



## Sequence Diagram for Member Login Use case

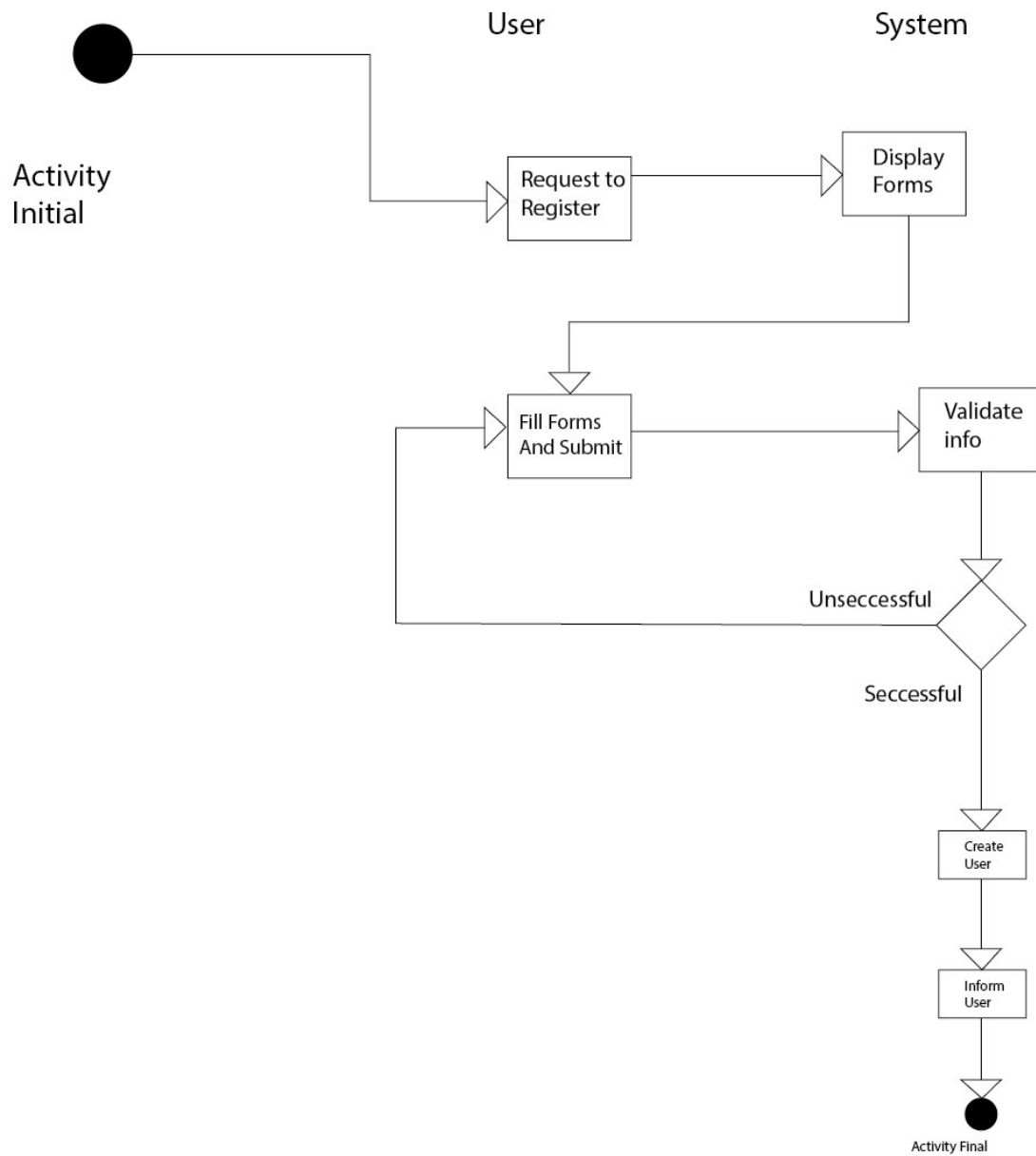


## Sequence Diagram for Comment Use case

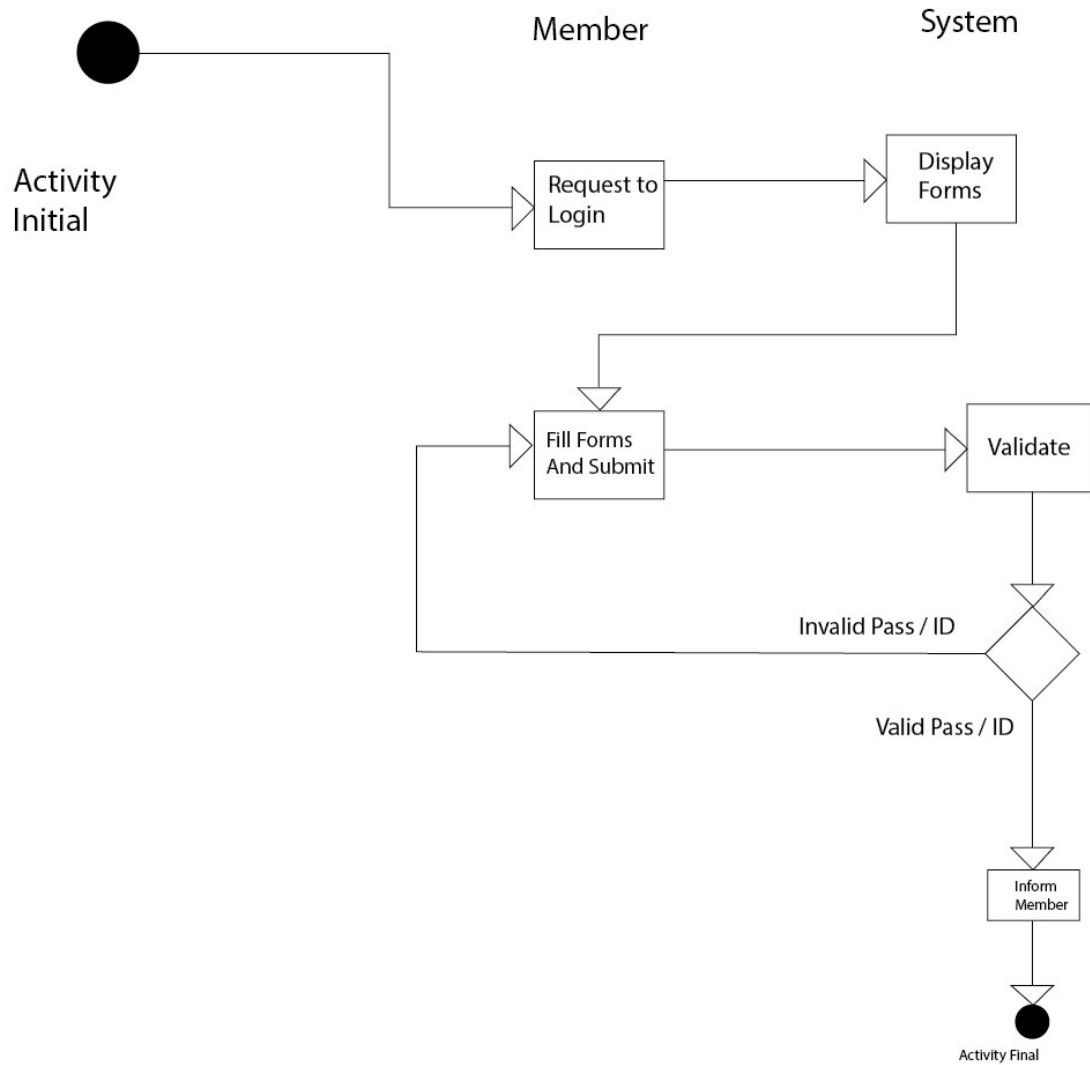


## Activity Diagrams

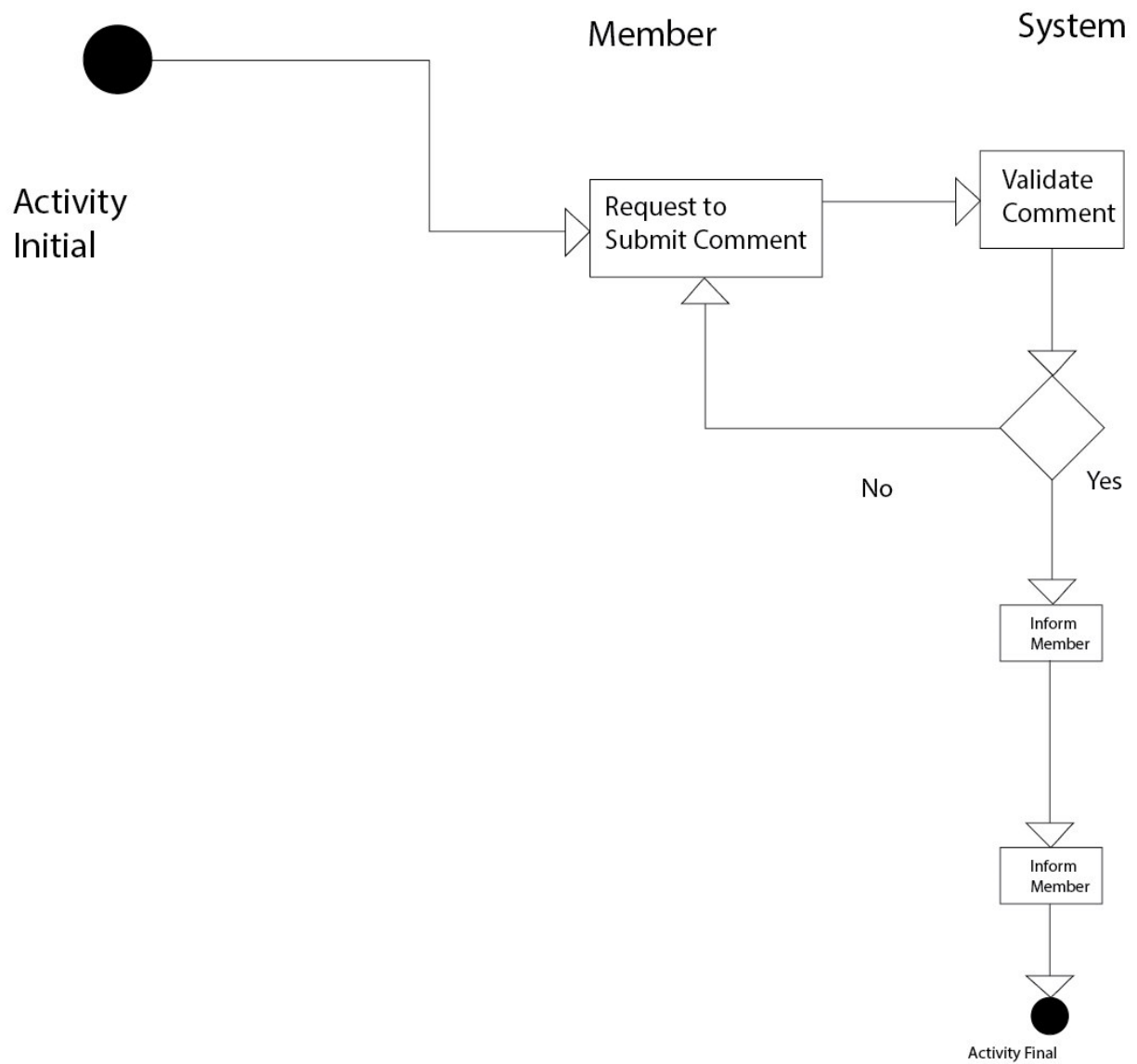
### Activity Diagram for User Register Use Case



## Activity Diagram for Member Login Use Case



## Activity Diagram for Comment Use Case



## System Diagram

