COMP 3059 – Capstone Project I

Software Requirements Analysis and Design Assignment

Capstone Project : Academy

Students : Nathalia Silva
Yuri Kusik

George Brown College – T127 – Programmer Analyst Program
COMP 3059- Capstone Project I
Professor – Anjana Shah
Revised - Winter 2020

Table of Contents

1.0 Introduction

The Introduction section provides an overview of the system using software requirements analysis and design for the scope of the system.

Academy will have a platform which will enable users to easily team up with others within their student body. In addition, the user can use this platform to learn more about academic events; therefore, using this application will allow students to better navigate their ways at college.

On Academy, users will be able to join groups according to their interests, they will be able to find academic support from instructors and tutors, and they will be always be aware of important academic events.

1.1 Purpose

The main goal of this project is to fill important gaps that academic institutions present regarding social experiences throughout post-secondary education life. That is the main reason why Academy started to be developed. This application has an incredible business value due its importance in the work towards improving student life.

1.2 Scope

This explains what the proposed system will and will not do. Describe relevant benefits, objectives and goals. The description of scope should be consistent with the Project Plan.

All eligible users must be able to register and login into the application. Once logged in, they can see all available interests (groups), and then start interactions with other users. The system administrator can view, edit and delete any users, interests (groups), events, posts and comments.

The application's domain will be provided by a partner web host company and the companies manager will use it in order to upload the website onto, and all Academy's developers will be working upon it. The website will have a highly-secure database in order to store information of those who have registered into the system.

2.0 System Overview

The System Overview section introduces the system context and design.

This section provides information about Academy's underlying system context, design and technical requirements, and it will also present non-functional requirements.

2.1 Project Perspective

The Project Perspective describes the context and origin of the system by defining whether the system is:

- *a follow-on member of a system family*
- *a replacement for existing systems, or*
- *a new self-contained system*

Currently, the market does not have any application with the same purpose that Academy has. Our proposed application will have a number of features that will enable the individuals to better explorer their academy opportunities.

2.2 System Context

The System Context describes the resulting software within the business case, including strategic issues in which the system is involved or which it specifically addresses.

This web based product is intended for the colleges students; therefore, this application will be deployed into the web, enabling its users to access it from anywhere by using all the most popular web-browsers.

Academy will have an user-friendly interface where all users can operate all of Academy's functionality:
• cross-platform compatibility
• user-friendly and traditional layout
• Profile page
• Interest page
• Login and Registration pages
• Admin environment

As said before, in order to use Academy, all users are required to register on the application. Every new user added to the system will trigger a functionality which will save this data into the app's database, creating a new user profile.

It is important to highlight that the system administrator is able to oversee the system as a whole; then, this persona can edit or delete data in the system.

2.3 General Constraints

General Constraints identify any business or system constraints that will impact the manner in which the software is to be:

- *specified*
- *designed*
- *implemented, or*
- *tested*

This web application is provisioned to be built on secure web application frameworks which will be also highly flexible to work with. Academy will be using databases that match the application scalability.

The developers working in this project will always be extremely mindful about user privacy. Academy's priority is to always protect its user's information; therefore, even though data encryption is out of scope in the project's first phases, this technique is highly consider once the application matures.

2.4 Assumptions and Dependencies

List any assumptions that have been made during the initiation of the project. In addition, list any dependencies that may impact its success or the desired result.

2.4.1 Assumptions

The whole procedure of the application design will compass numerous skills in different areas of web design including interface design; graphic design; standardised code; user experience design; and user accessibility.

After the designing phase, the project will be moving towards the work that the application back-end requires. Only then, Academy will start its testing phase to ensure that the app is functional before its release to the market.

2.4.2 Dependencies

The project is divided by sprints, and every sprint depends on its precursor sprint. Therefore, Business Case approval will be required in order to create the project's High-Level Requirements. Project leads will need information from the Project Vision document in order to start with the development of the Team Charter and Project Plan deliverables. Requirements Assessment has to be defined in order to continue with design and analysis phase. After that, the project will start its designing phase, then Prototyping phase, to finally start the testing phase that precedes the application release.

3.0 Functional Requirements

3.1 List of Functional Requirements (F.R.)

F.R. 1 – Register

- *DESCRIPTION*
  - If users are new and do not have an Academy account, they need to enter their George Brown College's email and personal information in order to register
- *INPUT*
  - Request for first name, last name , email, and password
- *PROCESSING*
  - If all the inputs are valid, the system will store the data and send a activation email to the user's George Brown College's email address
- *OUTPUT*
  - Displays confirmation email sent message

F.R. 2 – Activating account

- DESCRIPTION
  - Users will receive an email if they are eligible to use the app and their registration was successfully processed
- INPUT
  - Users simply need to click on activation link to activate their account
- PROCESSING
  - System retrieves the user's information and store the new account data
- OUTPUT
  - Displays registration completed message

F.R. 3 – Login

- DESCRIPTION
  - Registered users can access their account by providing email and password
- INPUT
  - User's mail and password are inserted
- PROCESSING
  - Validate the information and process the request - if they are both valid - the user will be granted with access to Academy
- OUTPUT
  - Displays Academy's homepage

F.R. 4 – Logout

- DESCRIPTION
  - Logged users can end their session
- INPUT
  - The user requests to be logged out, clicking on logout button
- PROCESSING
  - User session is terminated
- OUTPUT
  - Displays the login page

F.R. 5 – Anonymous comment

- DESCRIPTION
  - Posting relevant information in a post with an anonymous status (nameless)
- INPUT
  - Filling out the post form, and select the "Anonymous" option before publishing it
- PROCESSING
  - After the system validates the form, it stores the post into the database; then, it publishes the information on the platform
- OUTPUT
  - Post is published

F.R. 6 – Comment

- DESCRIPTION
  - Posting relevant information in a post attaching the user name to the post
- INPUT
  - Filling out the comment form, and select the “Publish” button
- PROCESSING
  - After the system validates the form, it stores the post into the database; then, it publishes the information on the platform
- OUTPUT
  - Post is published

F.R. 7 – Reset Password

- DESCRIPTION
  - If registered users wish to change their personal password, they can either use the “Forgot Password” option at login, or they can use the “Reset Password” option available at their Profile Page
- INPUT
  - Provide the system with user’s email
- PROCESSING
  - Submitting this form will trigger the system to check if the email address provided is valid. If so, the user will receive an email that will enable the password change into the database
- OUTPUT
  - Displays message “Please Check your Email for Password Reset Link”

F.R. 8 – Edit interests

- DESCRIPTION
  - Allows to add or remove field of interest in the user’s account
- INPUT
  - User fill out the profile form after hitting the “Edit” button at the Profile page
- PROCESSING
  - After the form is submitted by the user, the system checks and validates the information; then, it processes the request, storing edited fields into the database
- OUTPUT
  - System applies and shows the changes, displaying a completion message

F.R. 9 – Comment Management

- DESCRIPTION
  - System Administration ability of adding, modifying or deleting any user comment
- INPUT
  - Select comment, click the Edit button, change data and click the Save button
- PROCESSING
  - Data is validated and saved into the database
- OUTPUT
  - System displays a “Changes Saved” message

F.R. 10 – Post

- DESCRIPTION
  - Allows users to create a new relevant post in the platform
- INPUT
  - Fill out “New Post” form
- PROCESSING
  - Data is validated and saved into the database. New post is created, but it can be modified by the system administrator at any time
- OUTPUT
  - System displays a “New Post Created” message

F.R. 11 – Post management

- DESCRIPTION
  - System Administration ability of adding, modifying or deleting any user’s post
- INPUT
  - Select post, click the Edit button, change/delete data, and save changes
- PROCESSING
  - Data is validated and saved into the database
- OUTPUT
  - System displays a “Changes Saved” message

F.R. 12 – Create Event

- DESCRIPTION
  - Allows users to create a new Event which is relevant to post in the platform
- INPUT
  - Fill out “New Event” form
- PROCESSING
  - Data is validated and saved into the database. New Event is created, but it can be modified by the system administrator at any time
- OUTPUT
  - System displays a “New Event Created” message

F.R. 13 – Event management

- DESCRIPTION
  - System Administration ability of adding, modifying or deleting Academy's events
- INPUT
  - Select Event, click the Edit button, change/delete data, and save changes
- PROCESSING
  - Data is validated and saved into the database
- OUTPUT
  - System displays a "Changes Saved" message

F.R. 14 – Remove User

- DESCRIPTION
  - Deleting an user from Academy
- INPUT
  - Select the user to be deleted, and click on "Remove User" button, confirm deletion when the confirmation box pops-out
- PROCESSING
  - Remove an user with all its data from Academy
- OUTPUT
  - System displays a "Changes Saved" message

F.R. 15 – User questionnaire

- DESCRIPTION
  - Users can fill out a non-mandatory questionnaire (Soft Skills) in order to predict a match with possible posts' discussion to join within Academy
- INPUT
  - User selects answers for all the questionnaire's registered questions
- PROCESSING
  - System retrieves the provided information, storing this data
- OUTPUT
  - System displays a "Survey Completed" message, and display list of posts

F.R. 16 – Badge

- DESCRIPTION
  - Allows an user to have a badge displayed in the user profile
- INPUT
  - Use hits "Request Badge" button at Profile Page, fill out a form by sending a pdf/doc as a proof of identity. This proof might be a proof of employment or an endorsement letter
- PROCESSING
  - After the form is submitted by the user, the system checks and validates the information; then, it processes the request. If badge is approved, the system administrator will add either an instructor or a tutor badge
- OUTPUT
  - System displays a "Documentation Sent" message

3.2 Use Cases

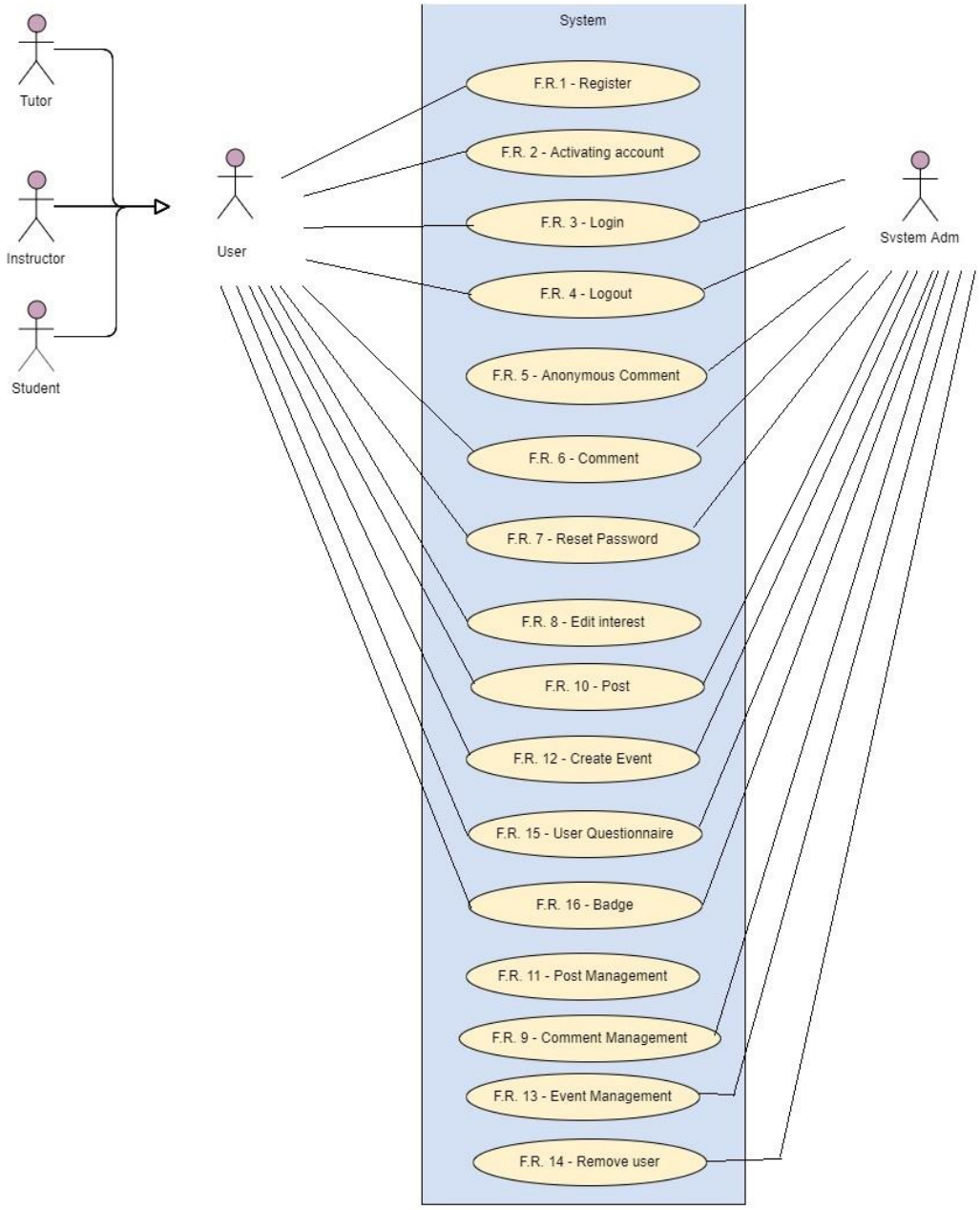


Figure 1- Use Case

3.3 Data Modelling and Analysis

- Normalized Data Model Diagram

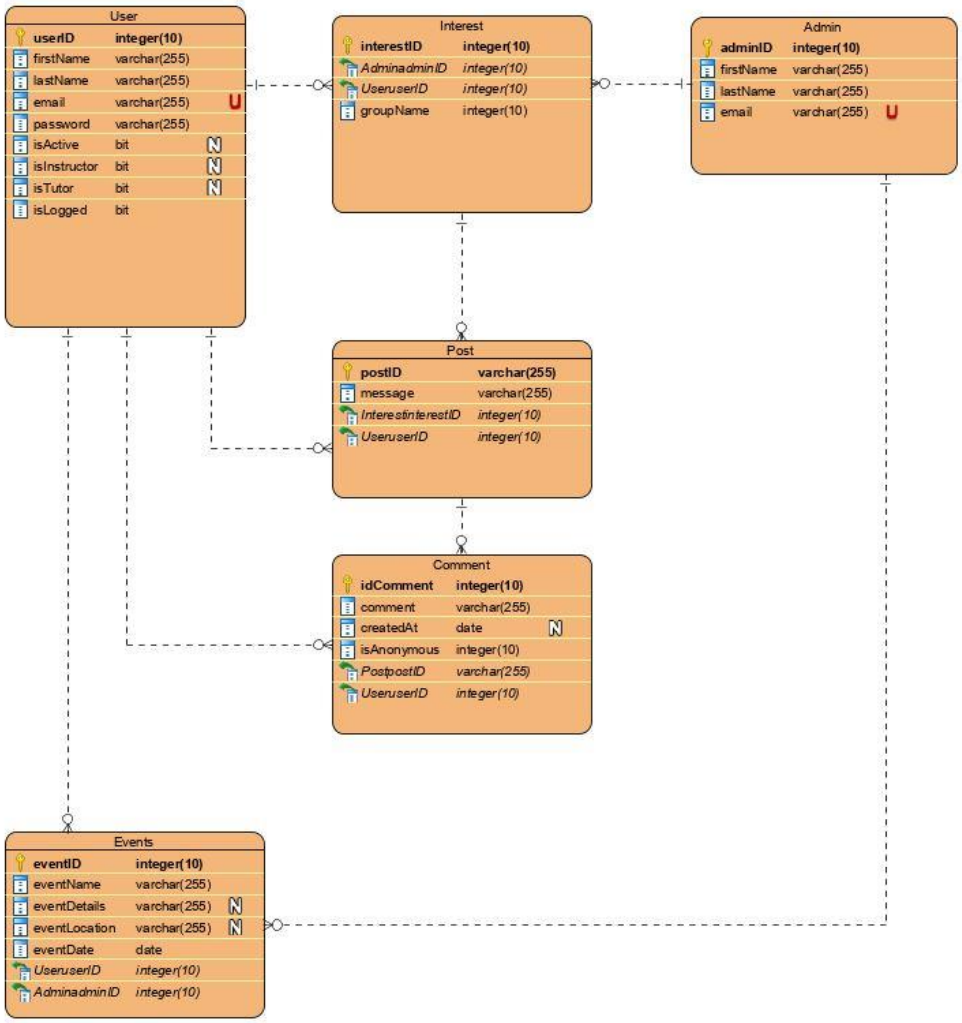


Figure 2- Normalized Data Model Diagram

3.4 Activity Diagrams

- F.R. 1 and F.R. 2 : Registration and Account Activation

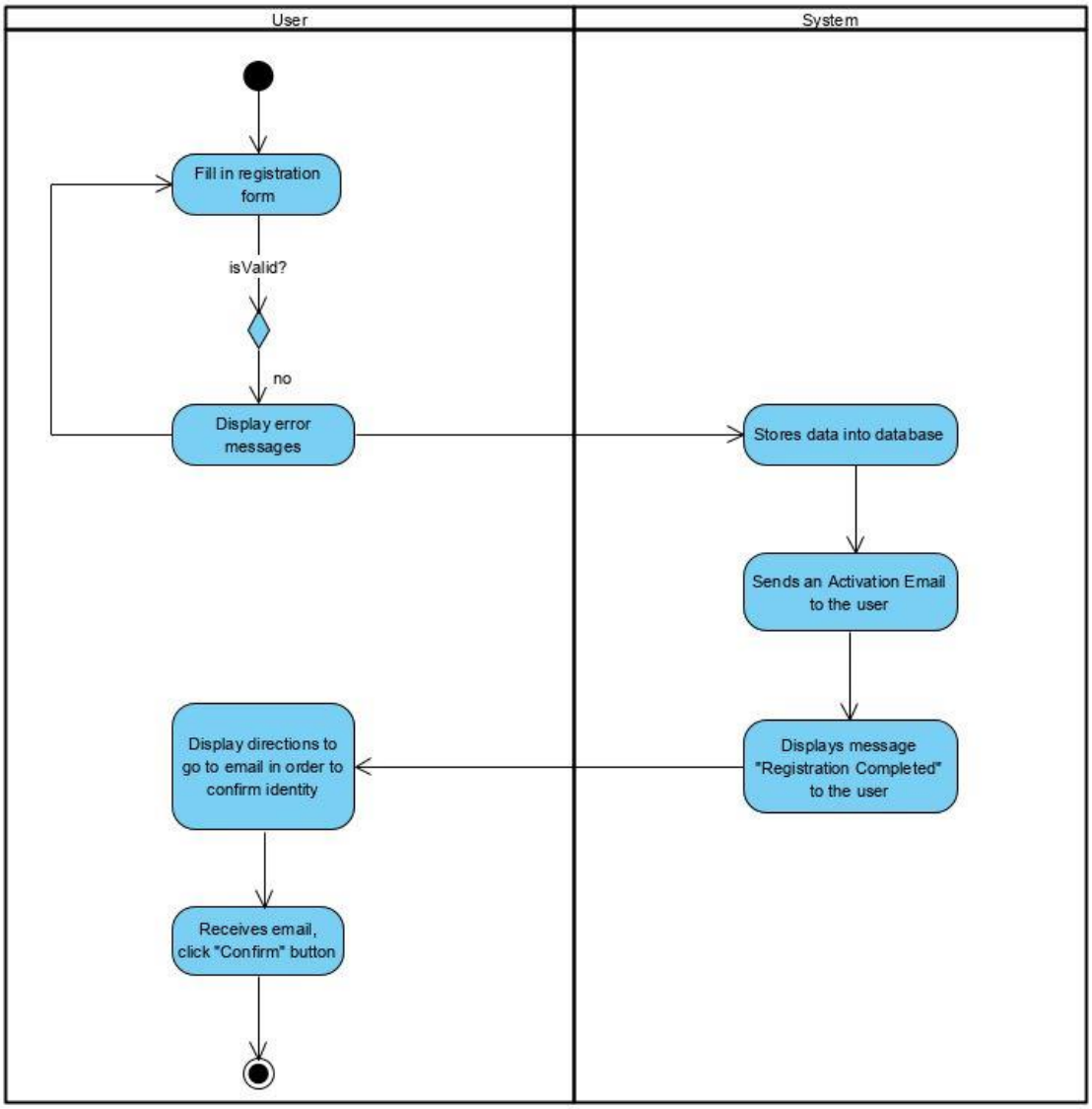


Figure 3 - Registration

- F.R. 3 : Login

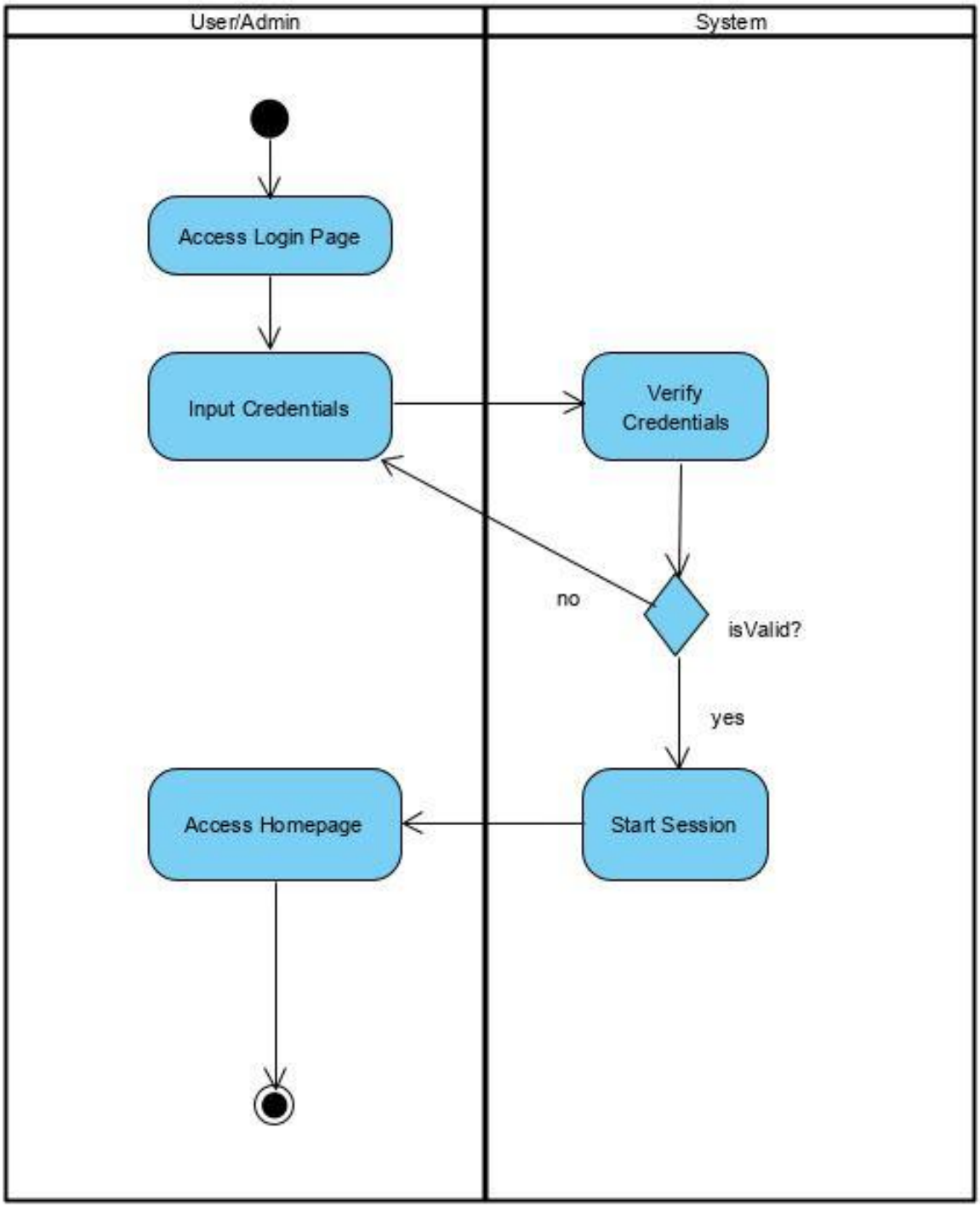


Figure 4 - Login

- F.R. 4 : Logout

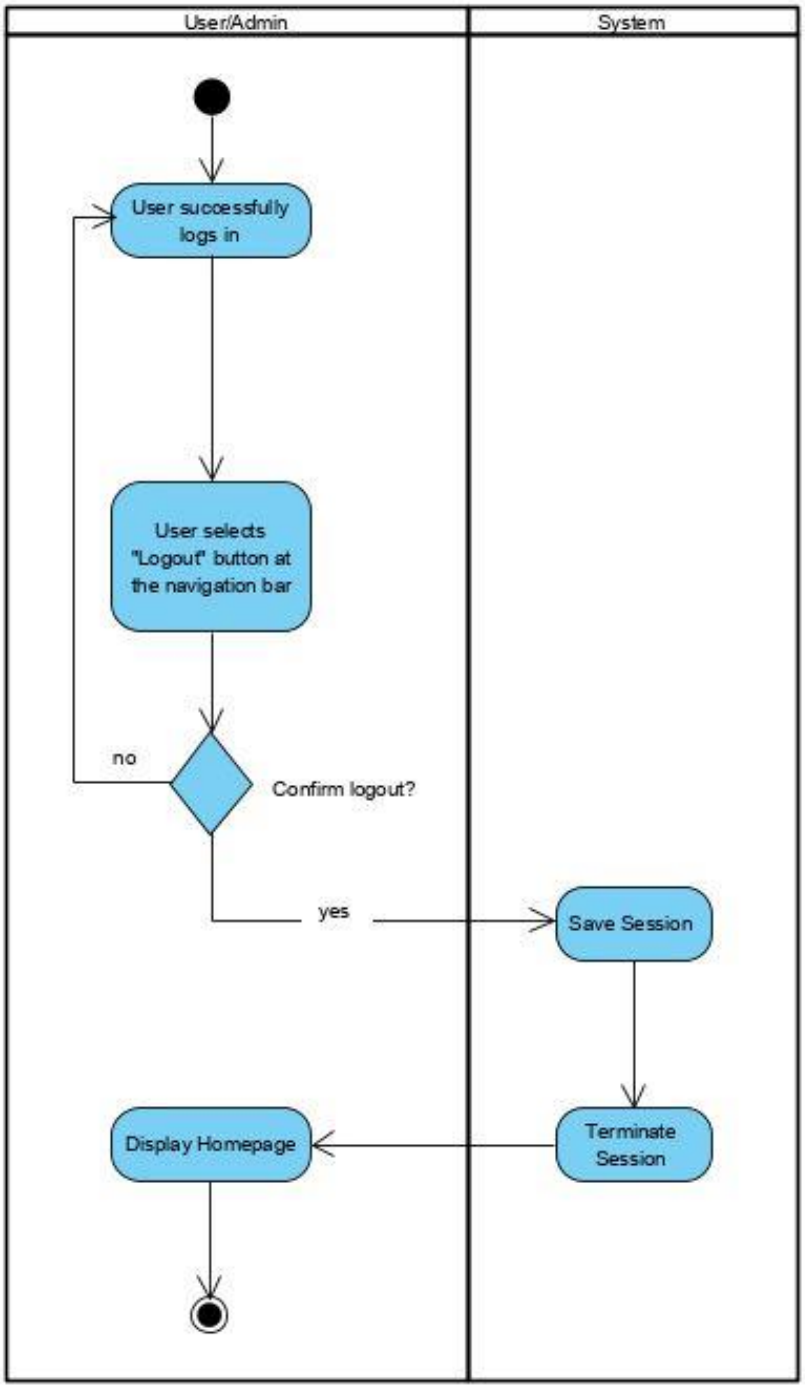


Figure 5- Logout

- F.R. 5 and F.R. 6 : Anonymous Comment and Comment

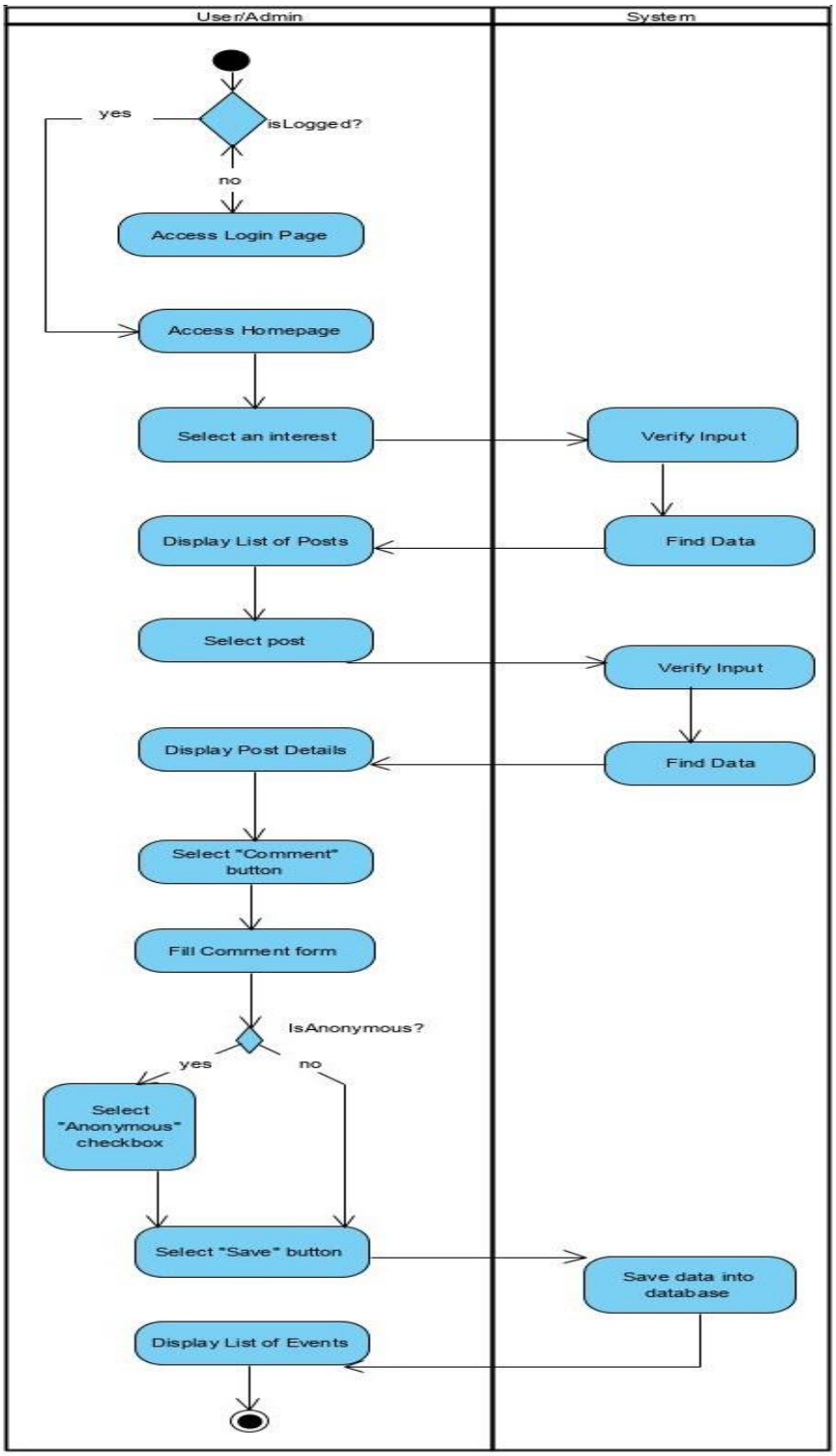


Figure 6 – Anonymous/General Comment

- F.R. 7 – Reset Password

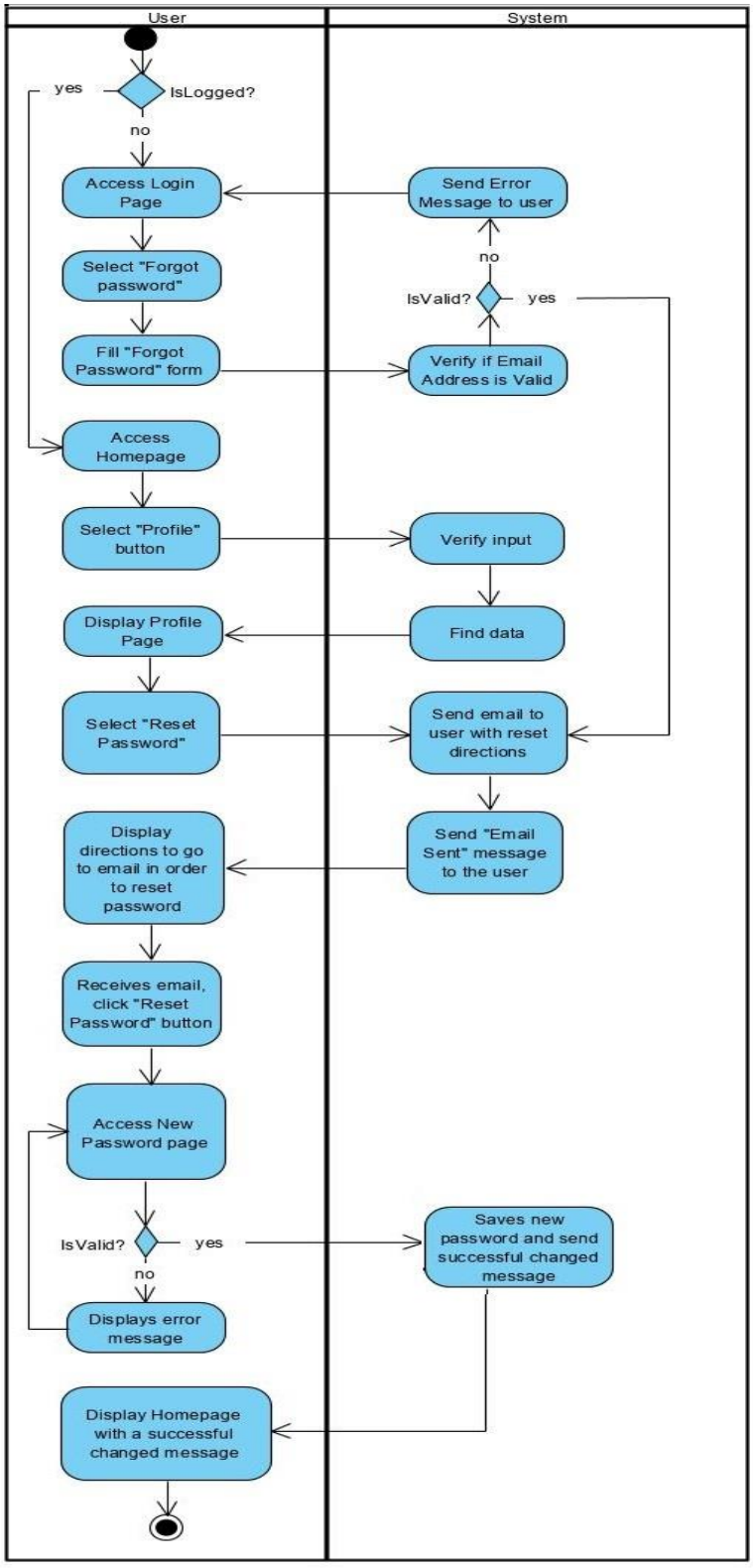


Figure 7- Password Reset

- F.R. 8 – Edit interests

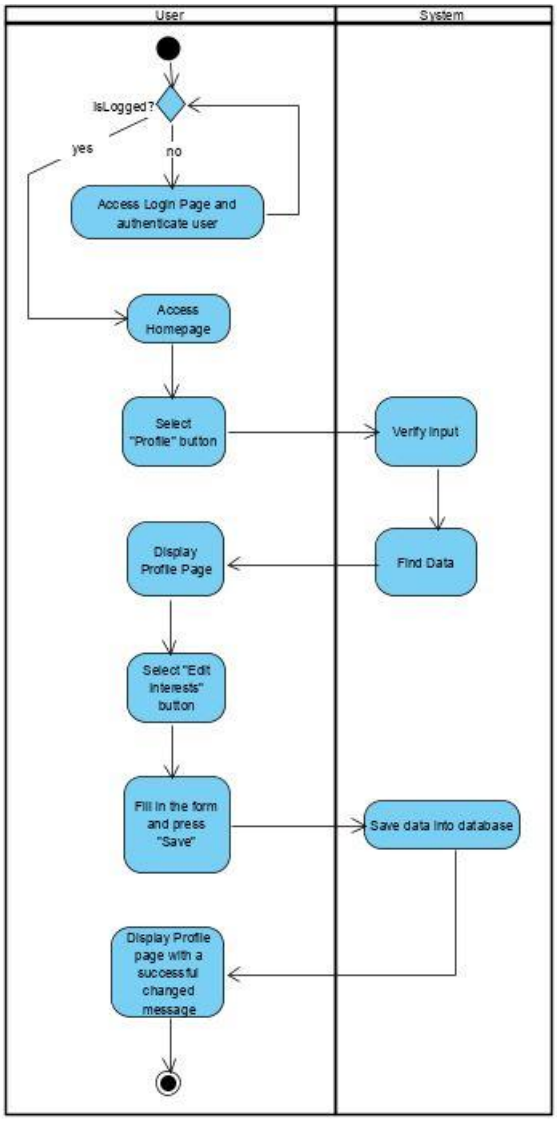


Figure 8 - Edit Interests

- F.R. 9 – Comment Management

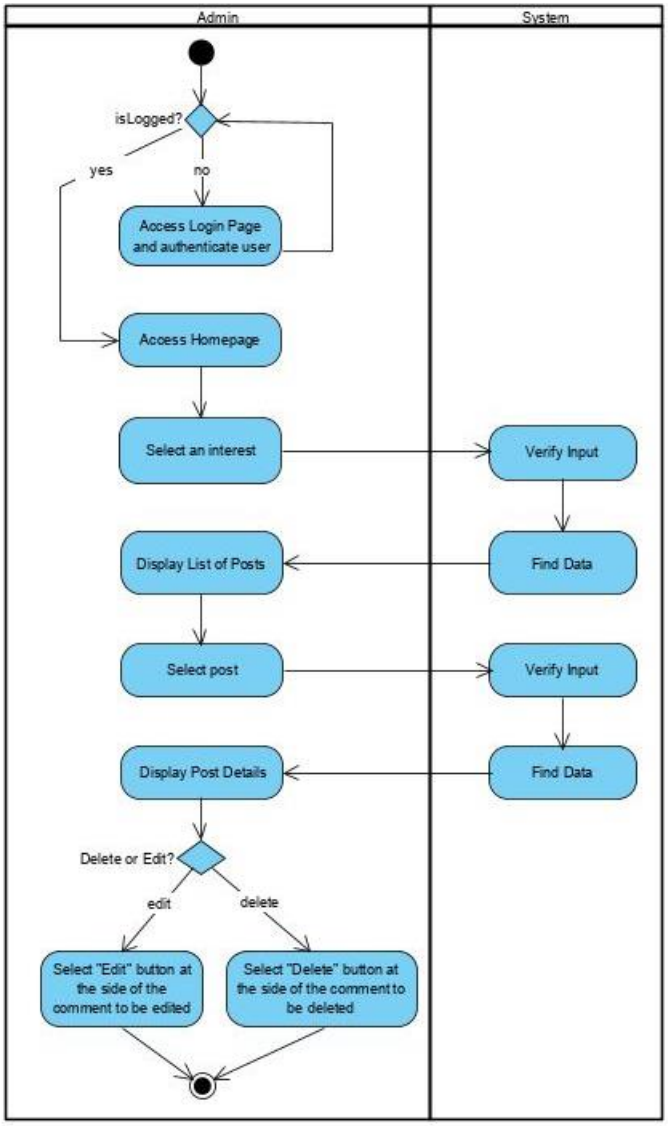


Figure 9 - Comment Management

- F.R. 10 – Post

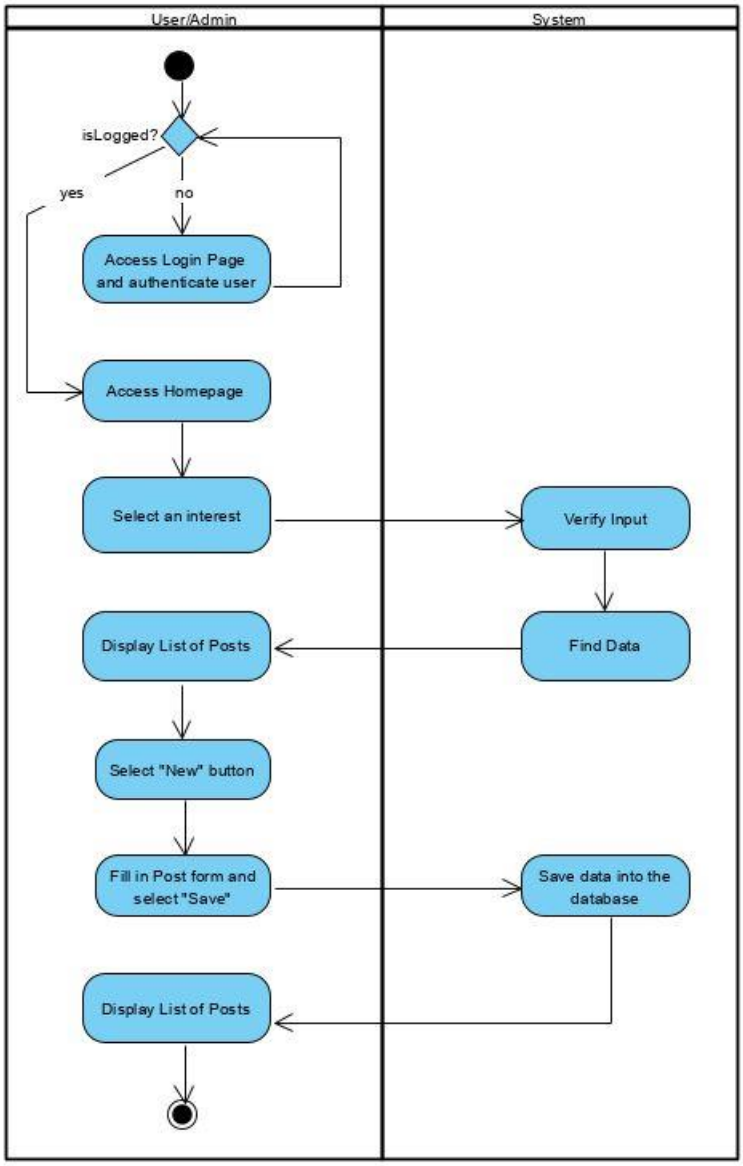


Figure 10 - Post

- F.R. 11 – Post management

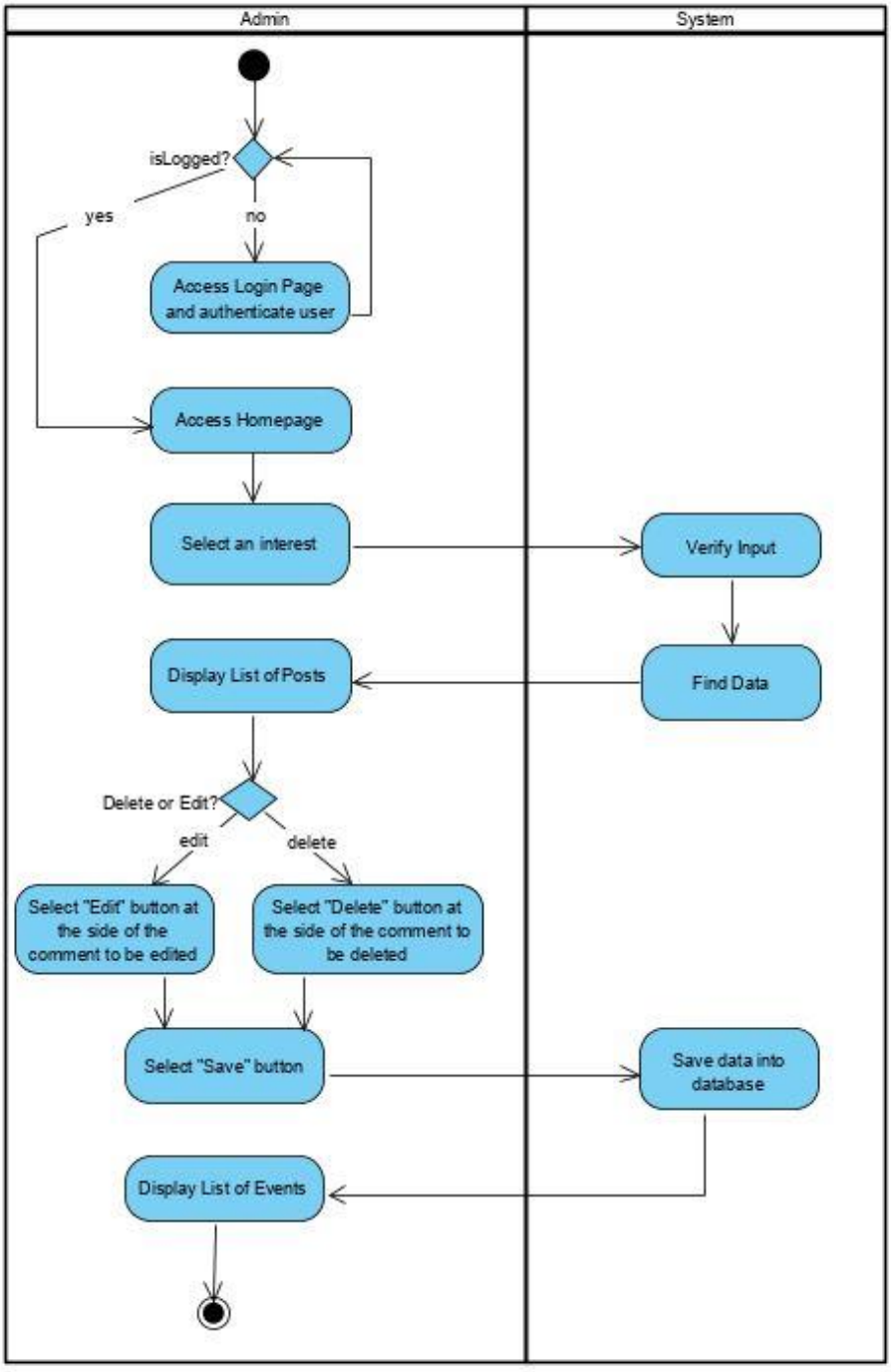


Figure 11- Post management

- F.R. 12 – Create Event

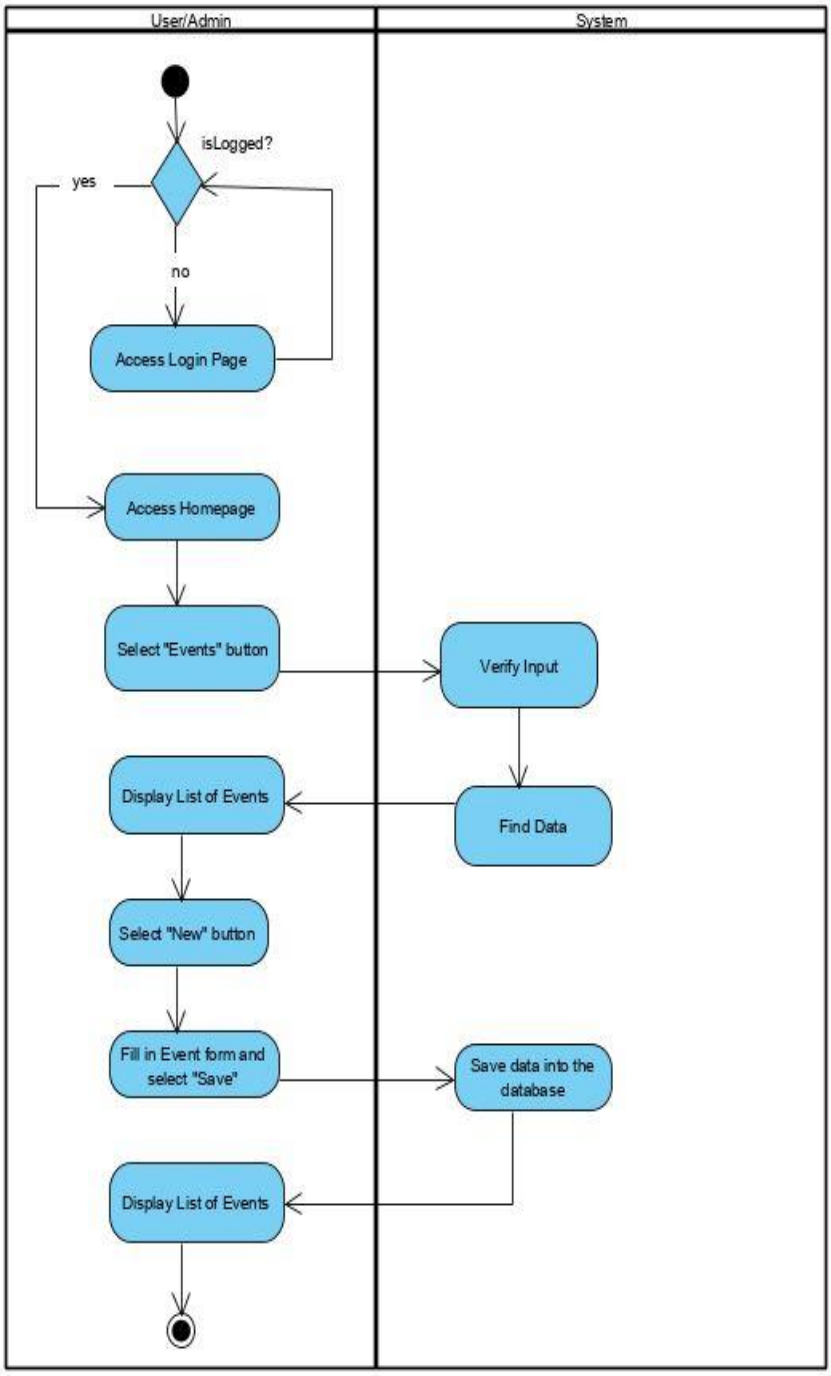


Figure 12 - Create Event

- F.R. 13 – Event Management

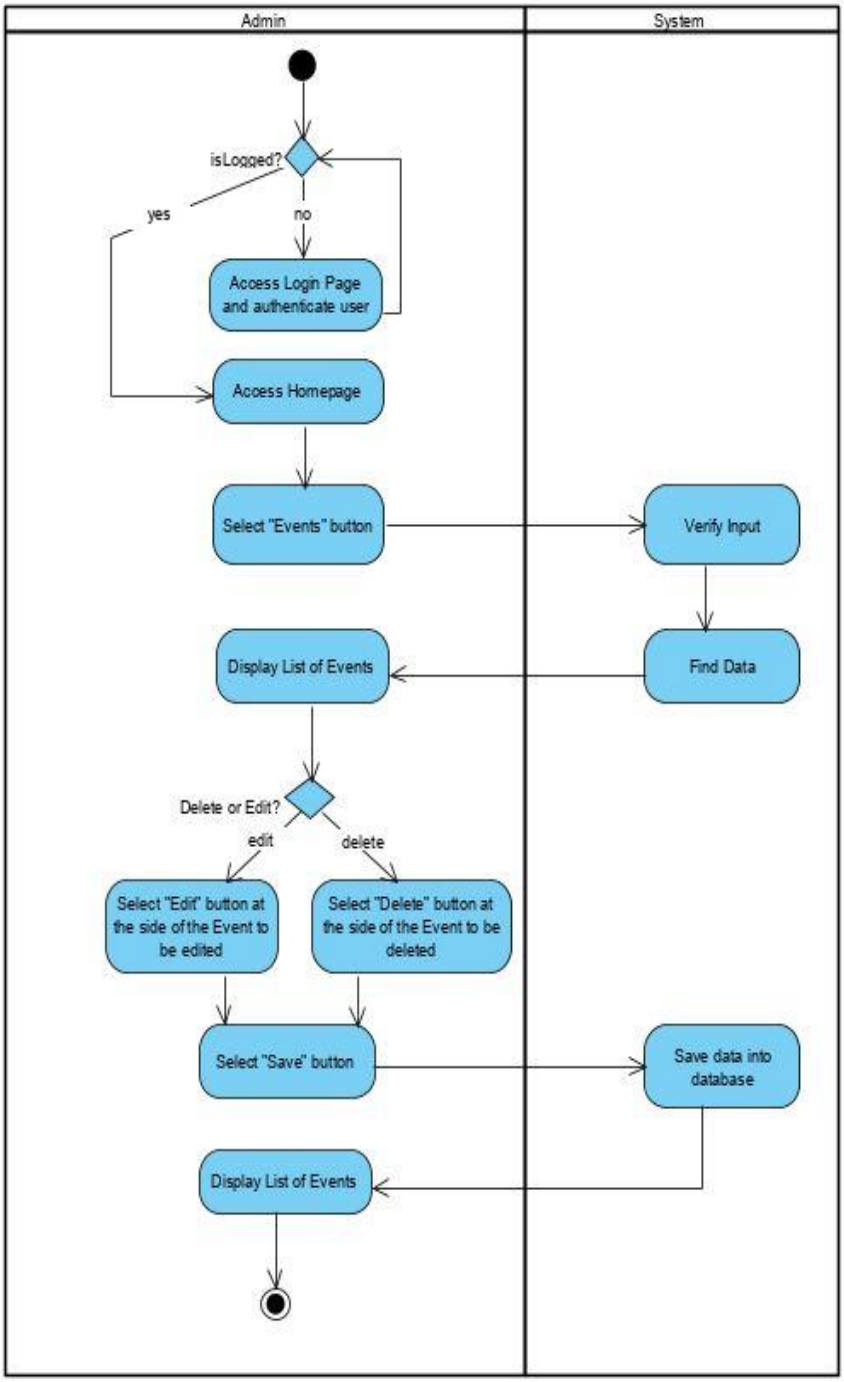


Figure 13 - Event management

- F.R. 14 – Remove User

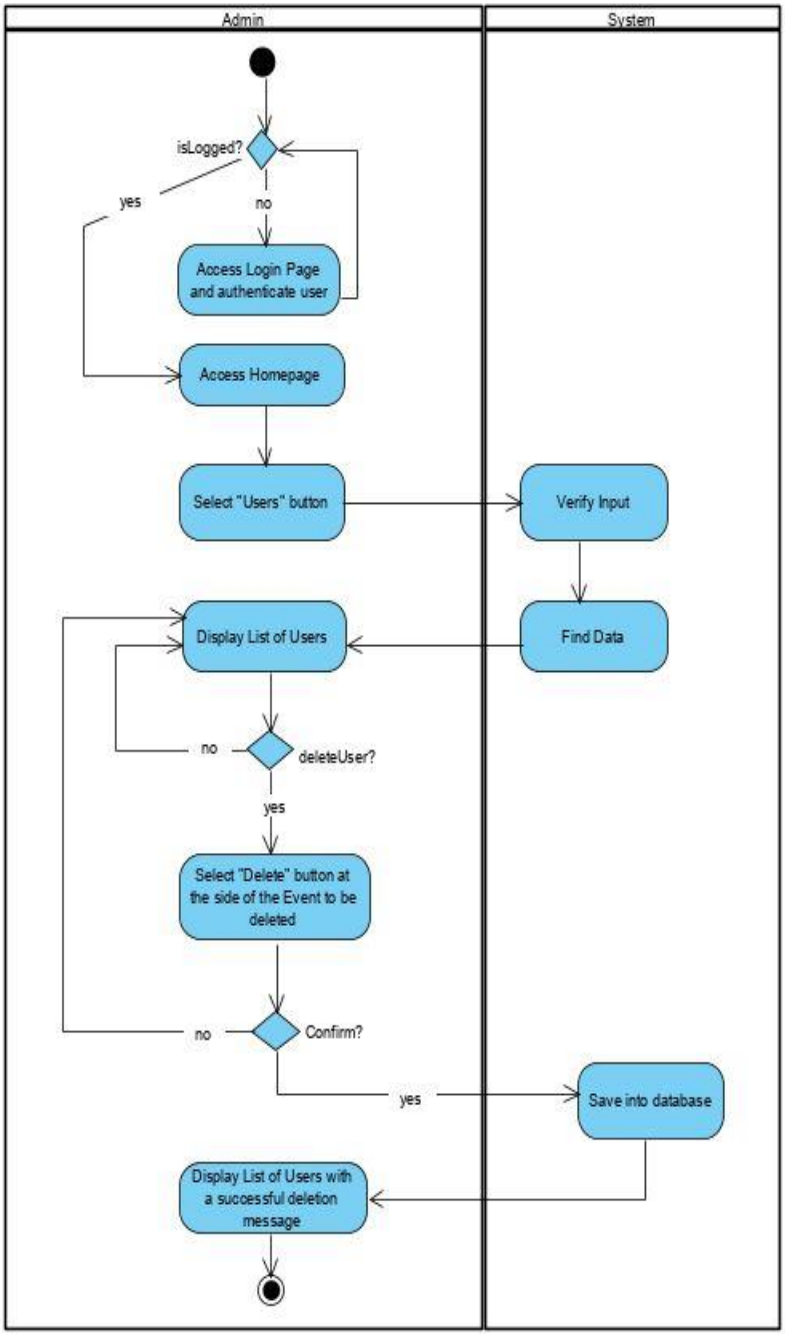


Figure 14 - Remove User

- F.R. 15 – User Questionnaire

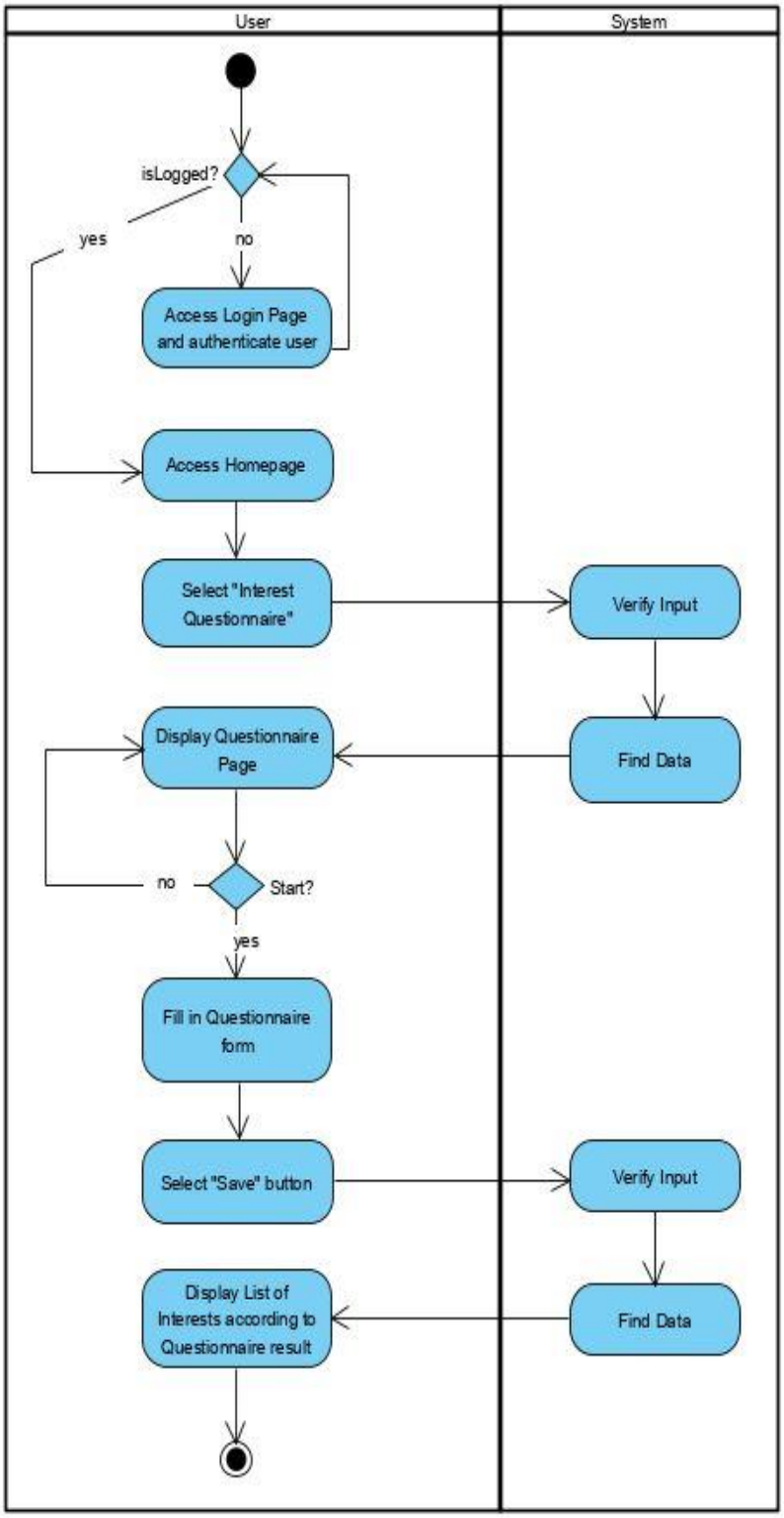


Figure 15 - User questionnaire

- F.R. 16 – Badge

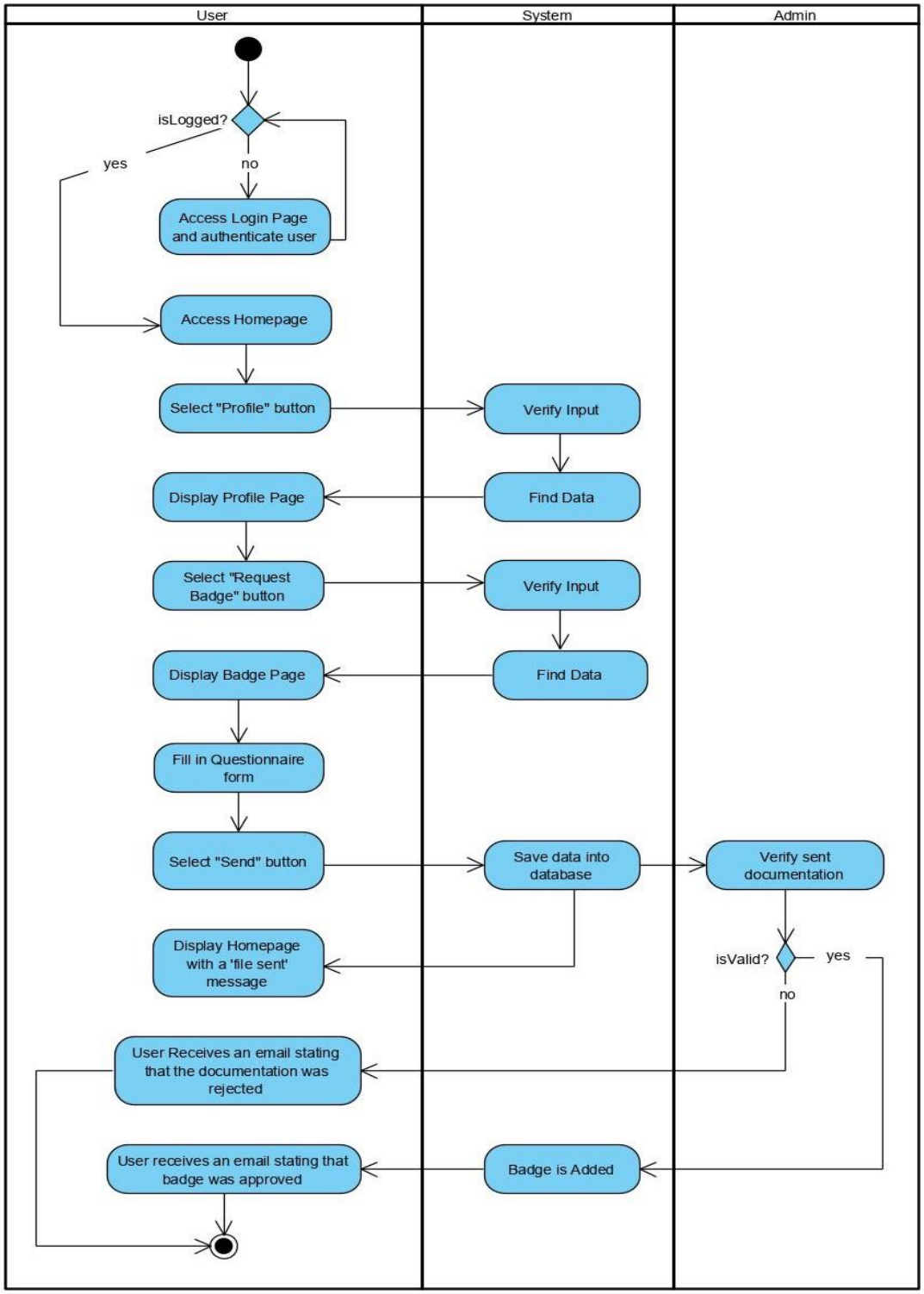


Figure 16 - Badges

3.5 Sequence Diagrams

- *F.R. 1 and F.R. 2 : Registration and Account Activation (no admin)*

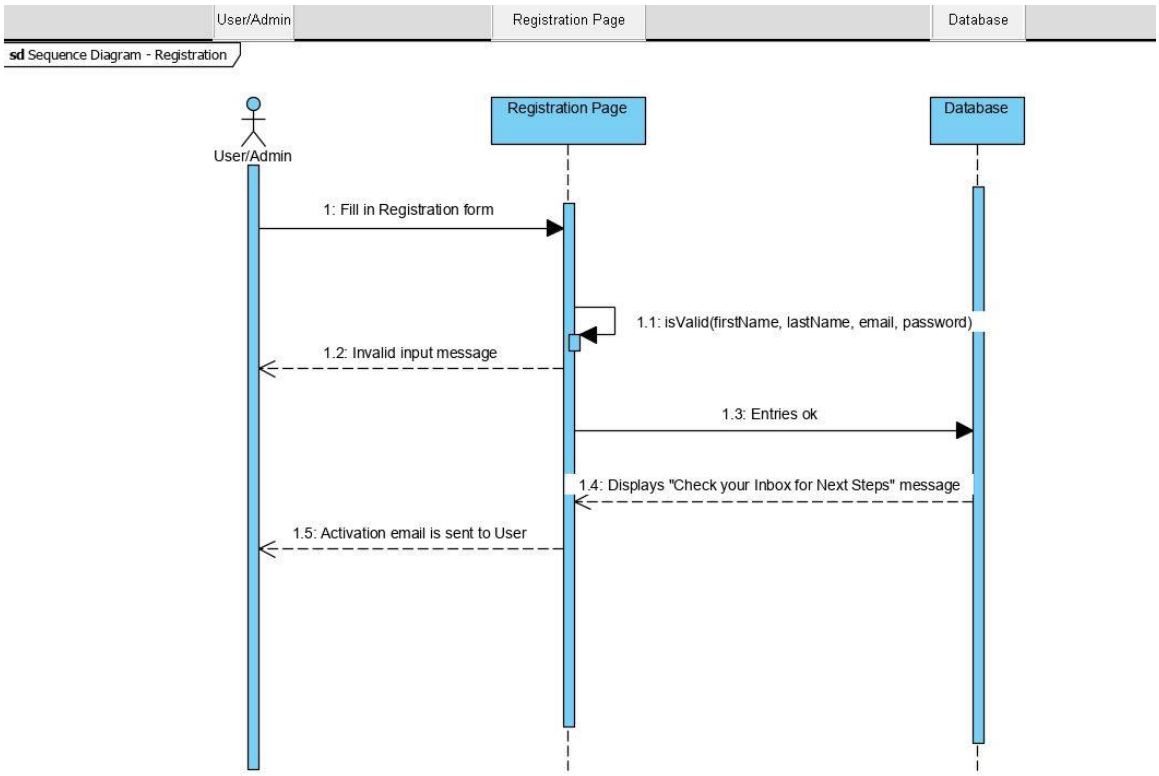


Figure 17 - Sequence Diagram for Registration

- **F.R. 3 – Login**

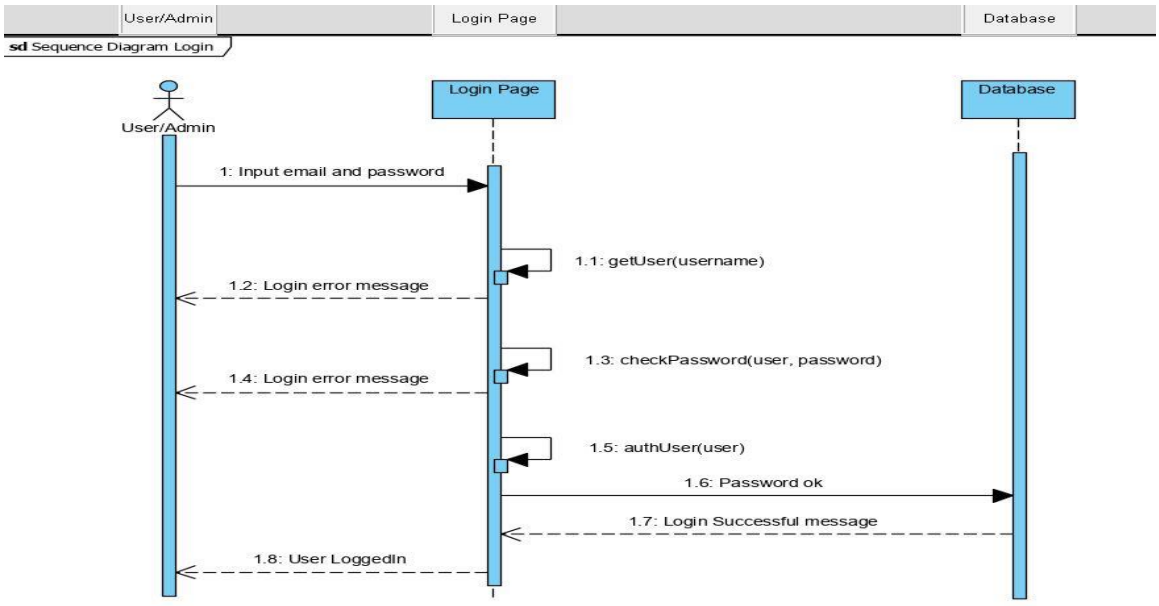


Figure 18 - Sequence Diagram for Login

- **F.R. 4 : Logout**

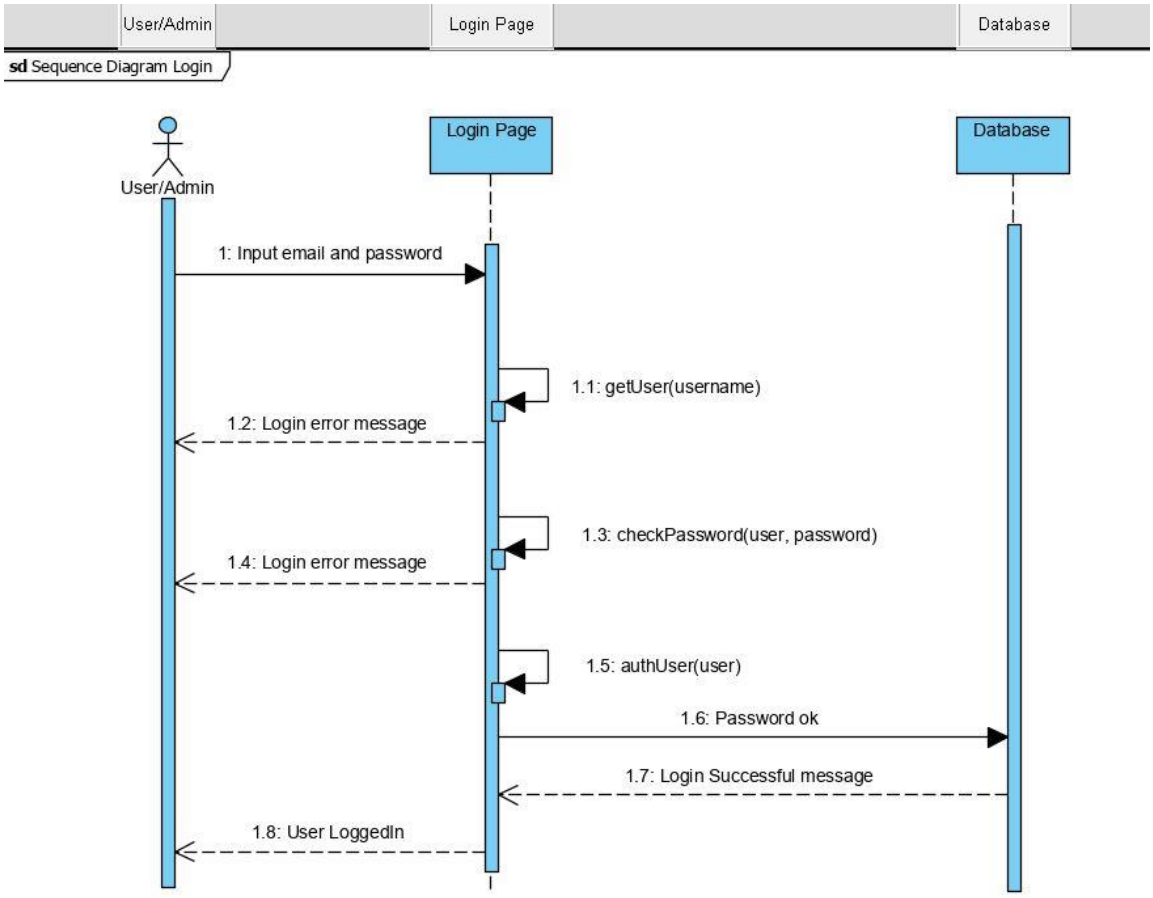


Figure 19- Sequence Diagram for Login

- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

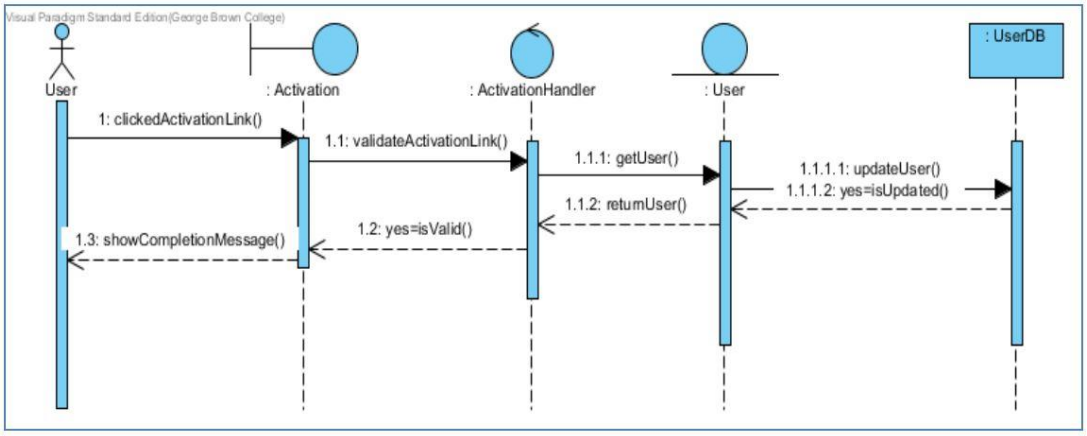
- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

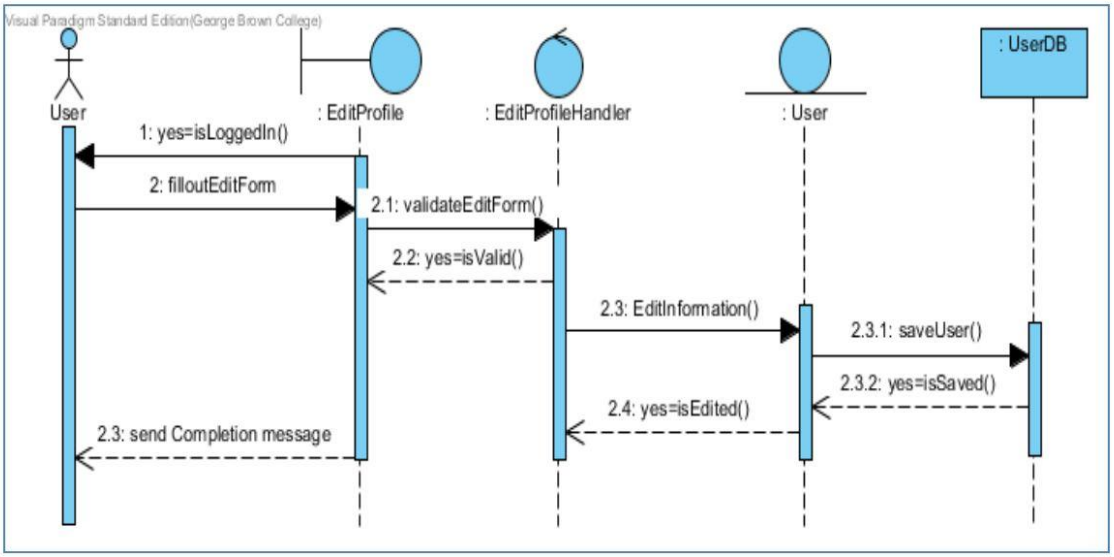
- **F.R. 4 : Logout**

- **F.R. 4 : Logout**

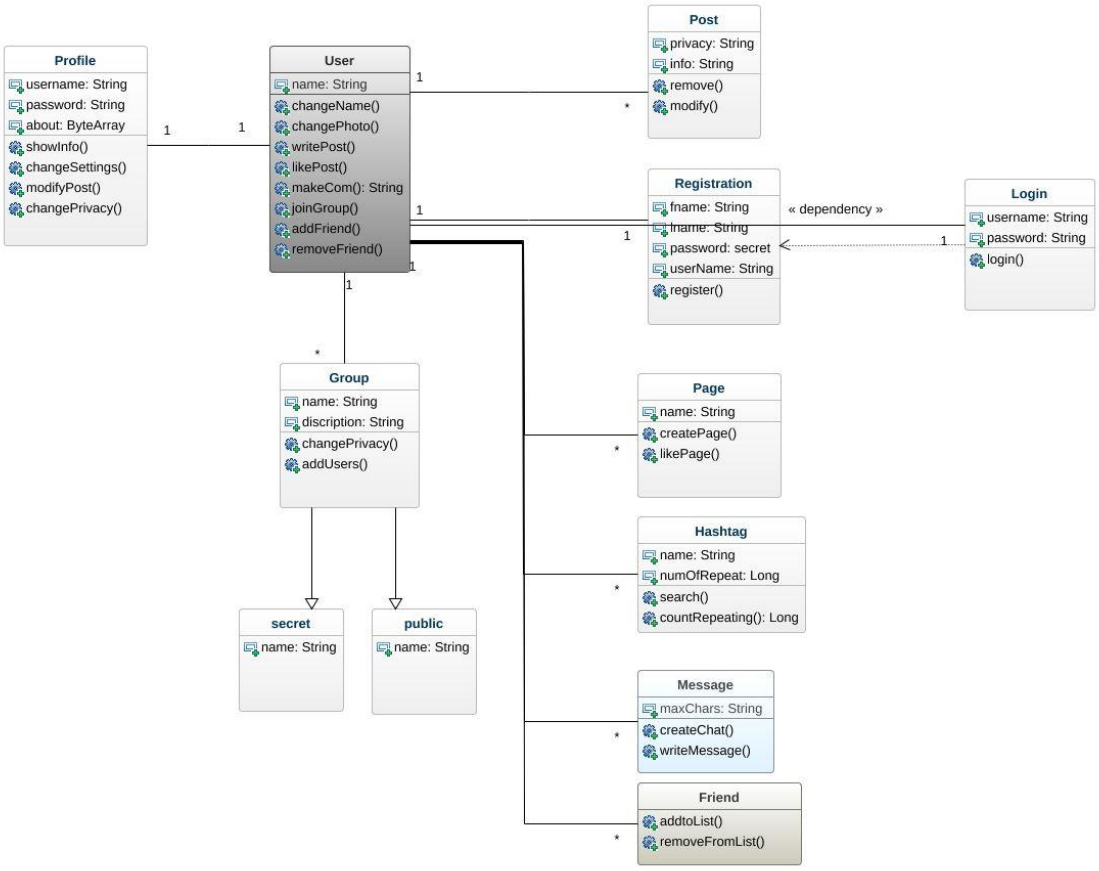
- Account activation



- Edit Profile

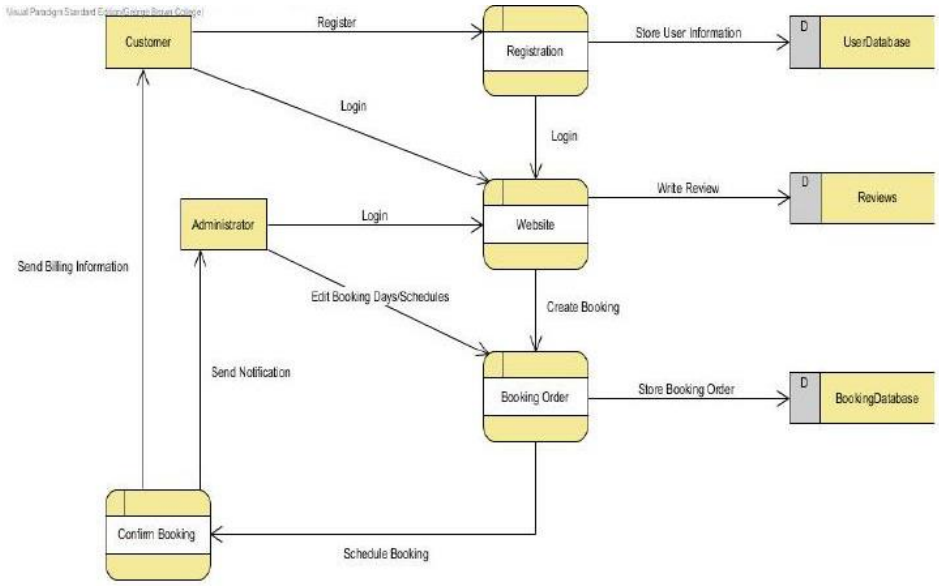


3.6 UML Class Diagram



3.7 Process Modelling

- Data Flow Diagram



4.0 Non-Functional Requirements

4.1 Operational Requirements

- A database is used for maintaining the database and the application server takes care of the site. In case of a failure, a re-initialization of the program will be done. Also, the software design is being done with modularity in mind so that its maintainability can be done efficiently.
- The application is HTML and scripting language based. So That end user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user is using this system on any browser. The system shall run on PC, Laptops.

4.2 Security Requirements

- The system should secure access of confidential data by username and password.
- This application is secure for every kind of its users, because if any user logout from any session then nobody will be able to access his profile without knowing his confidential password
- The system should not leave any cookies on the customer's computer containing the user's password, system's back-end servers shall only be accessible to authenticated administrators. Sensitive data will be encrypted before being sent.

4.3 Reliability and Availability Requirements

- The system provides storage of all databases on redundant computers with automatic switch over. The reliability of the overall program depends on the reliability of the separate components. The main pillar of reliability of the system is a database backup which is continuously maintained and updated to reflect the most recent changes. Thus, the overall stability of the system depends on the stability of container and its underlying operating system.
- The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

5.0 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

Yes, a database will be used in order to store information about customer's information such as their cars and address/phone numbers. The database will also hold images for certain pages. The repairs and services page will have images next to the service provided, in order to show what the service looks like. The data formats will be varchar, String, int, and date.

The website's database storage will be a limited size as we will be getting the database from the school to begin with. We will later use another domain with a database for the website that the company will purchase.

6.0 Other Requirements

Additional requirements, if any.

7.0 Approval

The signatures below indicate their approval of the contents of this document.

Project Role	Name	Signature	Date
Team Leader	Nathalia Silva	N.S.	Oct 30th, 2019
Tech Leader	Yuri Kusik	Y.K.	Oct 30th, 2019
Team Leader	**Nathalia Silva**	**N.S.**	**Jan 5th, 2020**