Software Requirements Specification

Version 1.2

November 22, 2020

Social Networking Webapp for Hobbies

Prepared by:

Aditi Pandey (189302065)

Krishna Modi (189301009)

Khushal Agarwal (189301167)

Oorja Garg(189302037)

Ashish Raj Shekhar (189301231)

Manipal University Jaipur

Table of Contents

1.0 Introduction

1.1 Purpose

The purpose of this document is to build a progressive web application to provide a social networking platform for connecting people with similar hobbies and/or interests at a low cost.

1.2 Document Conventions

Term	Definition
SNI	Social Networking Interface
PWA	Progressive Web App
SNS	Social Networking Site

1.3 Intended Audience and Reading Suggestions

This project is a prototype for a social networking platform to connect people with similar hobbies. It has no restriction over its use as of now, anyone who wishes to connect can use this PWA. This has been implemented under the guidance of college professor. This project is useful to everyone and anyone with hobbies.

1.4 Product Scope

The purpose of the SNI is to ease the connection of people with similar taste in hobbies and to create a convenient and easy-to-use PWA for people to view and discuss all sorts of ideas through individual and communities. SNS connects and presents people based on info gathered about them as stored in user profile.

1.5 References

- [IEEE] The applicable IEEE standards are published in “IEEE Standards Collection”
- How to write a Software Requirements Specifications https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document
- Software Requirement Specification For Social Networking with Advertisement. https://www.academia.edu/27811487/Software_Requirements_Specification_For_Social_Networking_with_Advertisement
- SRS of Social Networking https://www.slideshare.net/maaano786/srs-of-social-networking

2.0 Overall Description

2.1 Product Perspective

The social networking website allows users to connect with people of similar hobbies, helping one maintain existing relationships with such users and share ideas and ones creations to establish new connections with those who share similar interests by reaching out to them. This is an independent product which may or may not be released.

2.2 Product Functions

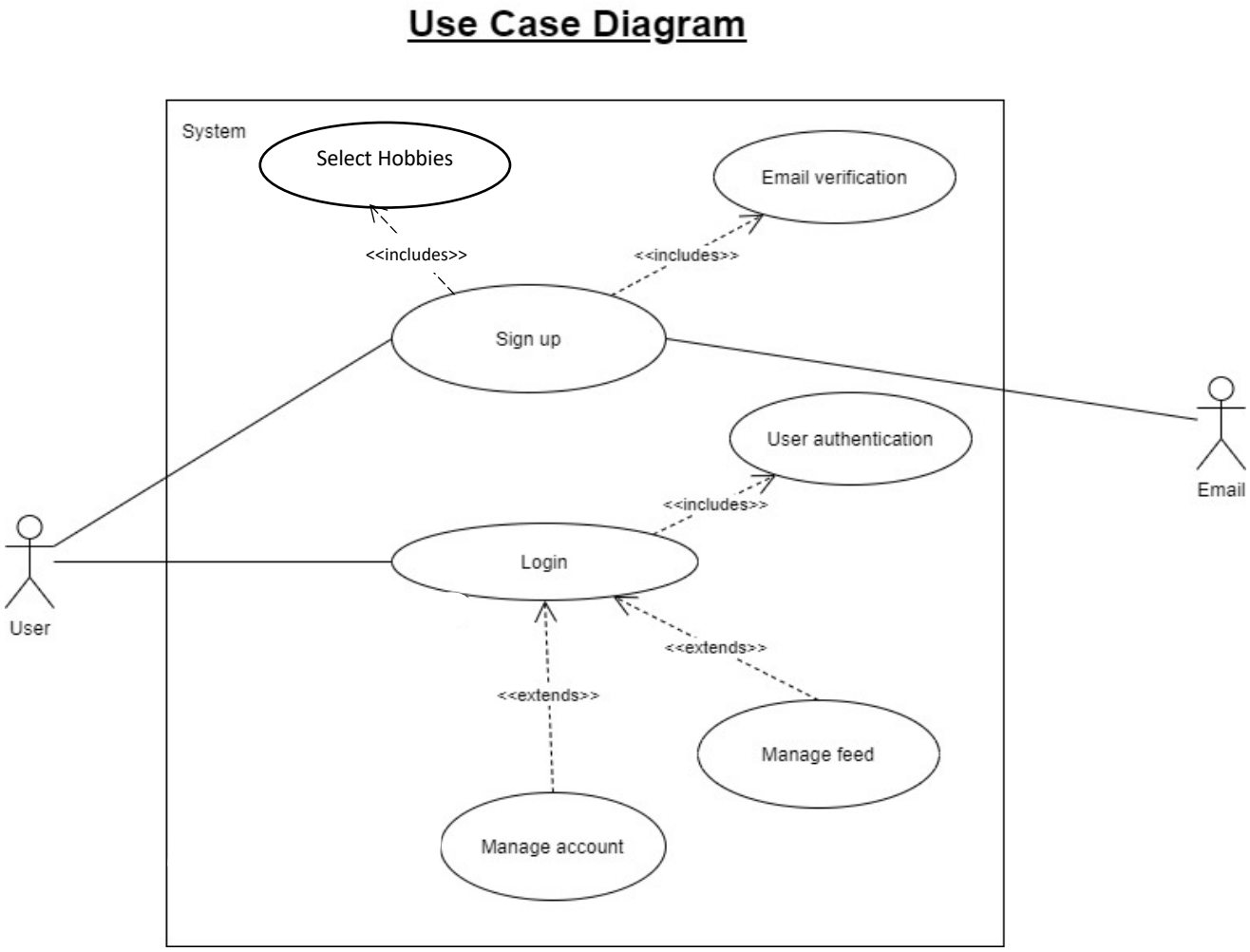
User can register to the PWA and login whenever he wishes to do so, create his profile with username, add and edit information, set visibility to various profile sections on his/her own profile and upload profile picture to his/her profile.

The application provides a common platform for those with similar interest to connect. Users can look up for profiles with similar hobbies and will be shown appropriate results if matching profiles are found. They can upload their work in the format of pictures and videos on the specific hobby board. The user has the choice to delete his account if needed.

2.3 User Classes and Characteristics

Standard users may belong to any demographic group including any gender, nationality above the age of 13 that can use any active device with internet access. User interface is in English language so users should have a basic English reading knowledge.

2.4 System Environment



2.5 Design and Implementation Constraints

- The PWA will support the following browsers: Microsoft Edge, Chrome and Firefox.
- The PWA supports English language only.
- The application is planned to be mobile responsive.

2.6 User Documentation

The remainder of this document is two chapters the first providing a full description of the project for the owners. It lists all the functions performed by the system. The final chapter concerns the details of each system function and actions in full for software developers' assistance. These two sections are cross-referenced by the topic to increase understanding by both groups.

2.7 Assumptions and Dependencies

- There will be availability of Internet via 3G,4G,5G or Wi-Fi.
- The user should have a valid email address in order to register into the website.
- The Central server of the system must be able to handle all the incoming requests simultaneously.

3.0 External Interface Requirements

3.1 User Interfaces:

Standard users will use the web browser to use the website. Hence, it shall have a login page for users to login into the site. For the new users, they will have to register. After logging into the system, the user will be shown his profile and the feed containing posts related to his/her interests. User is given options to log out, search people and change any of his personal settings on his home page.

3.2 Software Interfaces

3.3.1 E-mail Interface: This interface uses the SMTP services to send emails to the users.

3.3 Communication Interfaces

The social networking website is based on HTTP.

4.0 System Features

4.1 Sign Up

4.1.1 Description: User will create an account if there is no existing account associated with user email-id. User should enter account details such as name, email-id and password. All these are required fields and account creation is not possible unless these are filled. User is registered after this information is validated and agree to terms and conditions,must be 13 years old or more. Password should be at least 8 characters long including at least one capital and small letter, a special character and a number.

4.1.2 Dependency: None.

4.2 Login

4.2.1 Description: User must be registered in order to Login. The email id and password can be used for login. The hash of password should match to the hash of password stored in the database. If user clicks on forgot password then reset password link should be emailed to the user. Input to each field must be validated.

4.2.2 Dependency: Sign Up.

4.3 Select Hobbies

4.3.1 Description: The user is provided with a list of interests from which they can choose. The option will be provided as tags that the user can select. The user can choose multiple tags according to their preferences.

4.3.2 Dependency: Sign-Up .

4.4 Manage Account

4.4.1 Description: User should be able to add and edit the personal details like language, gender, upload the profile picture and give an introduction about themselves . These fields can remain empty. User must be logged in and his account should be verified to be able to add personal information. Input to each field must be validated. User information will be cleared when the account is deleted.

4.4.2 Dependency: Login.

4.5 Manage Feed

4.5.1 Description: Creation of a feed where the user can view content related to their chosen hobbies. The user can like, comment and save others' posts as well as upload content on their profiles. A user can remove a post that has been posted which would remove all the comments associated with that post. A user can delete a comment written by him or a comment on his posts.

4.5.2 Dependency: Login.

5.0 Other Non-Functional Requirements

5.1 Performance Requirements:

5.1.1 Speed: The application should be fast, it should not slow down with increase in the number of users. Search functionality should be fast to enable better end-user experience. The system should be quick enough to be able to respond to the user actions within a short period of time.

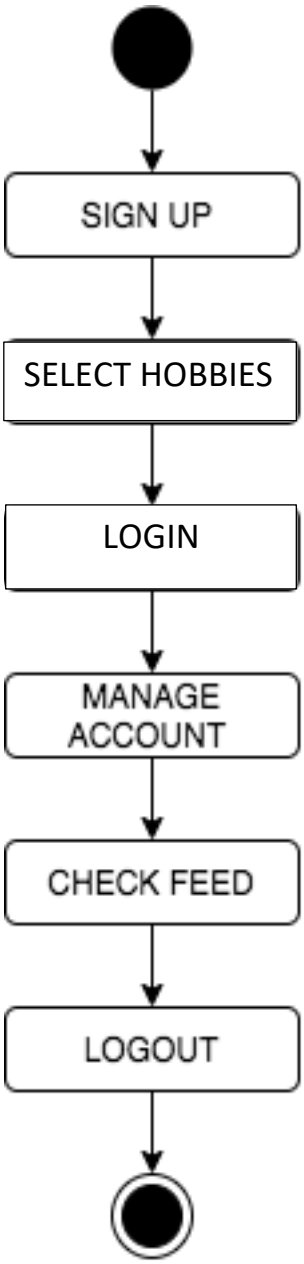
5.2 Software Quality Attributes:

5.2.1 Usability: User interface should be simple and understandable by any user.

5.2.2 Availability: The user will be able to use the application when connected to the internet with minimum to no downtime to ensure a better experience. The system should be reliable and should yield expected result if an account is searched.

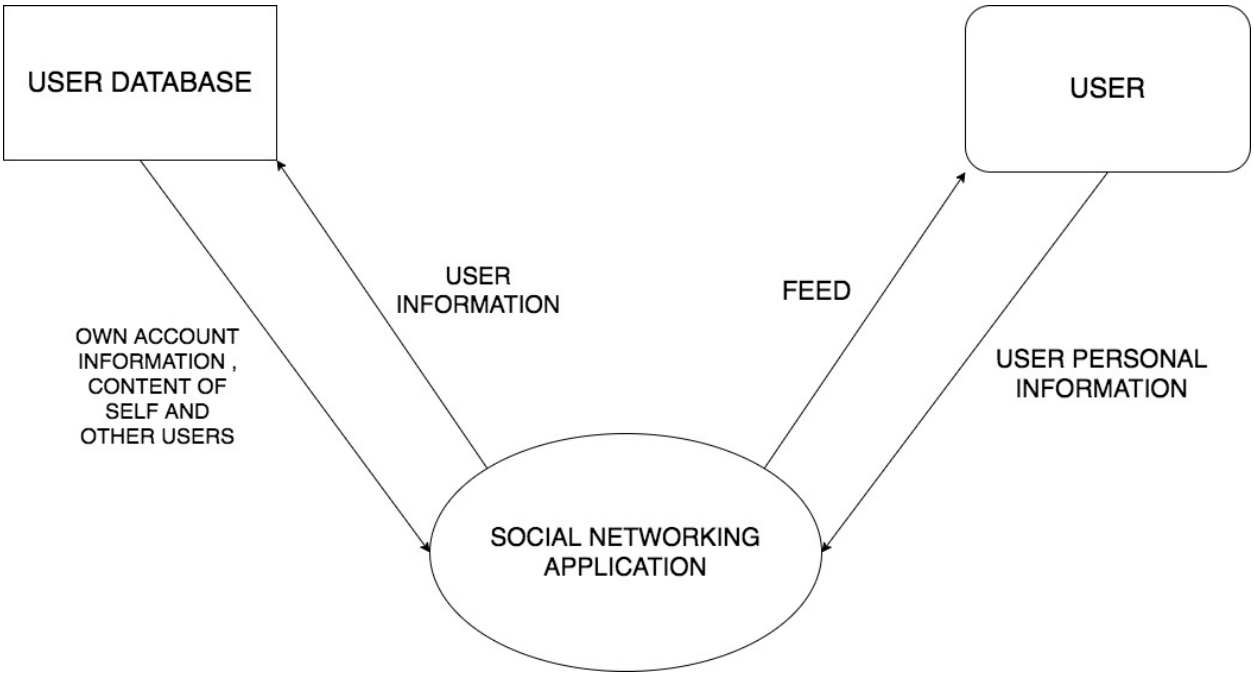
6.0 Design Model

6.1 Activity Diagram:

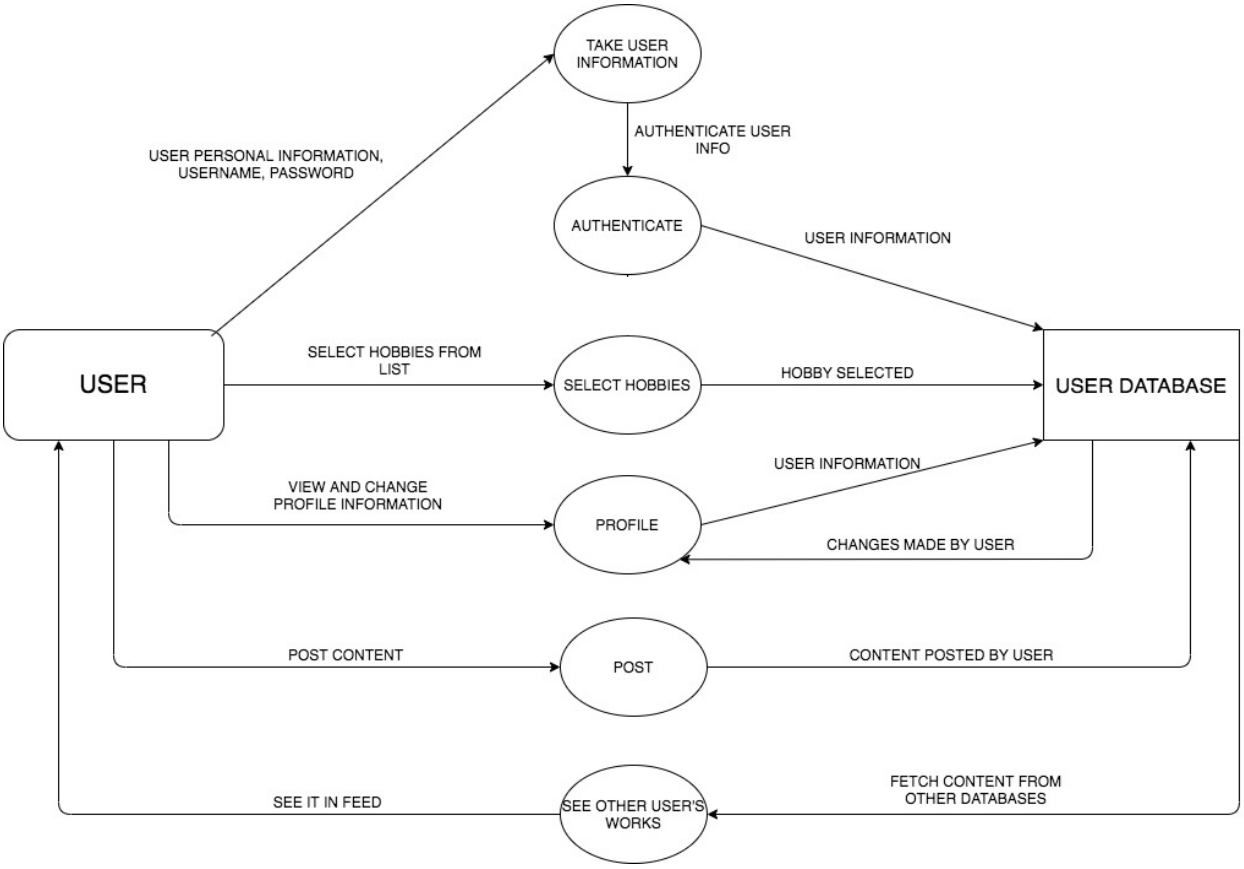


6.2 Data Flow Diagram:

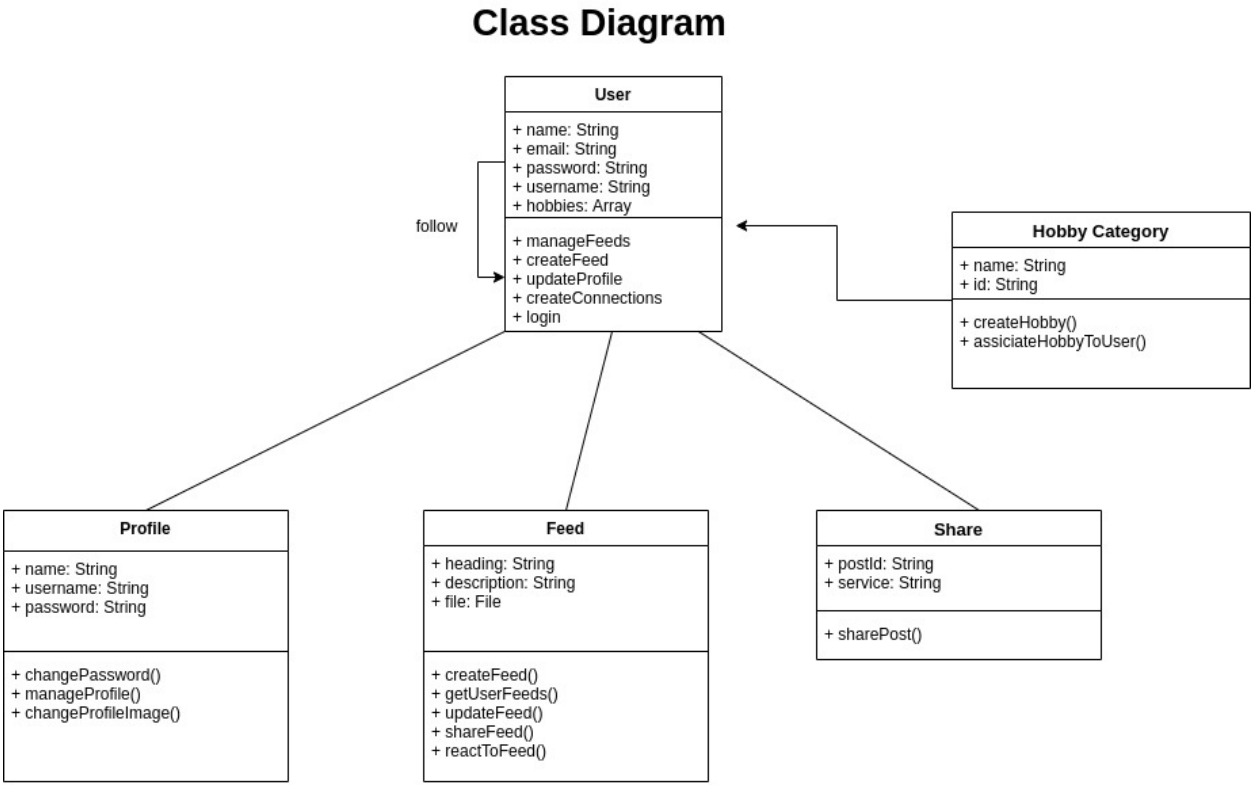
Level 0



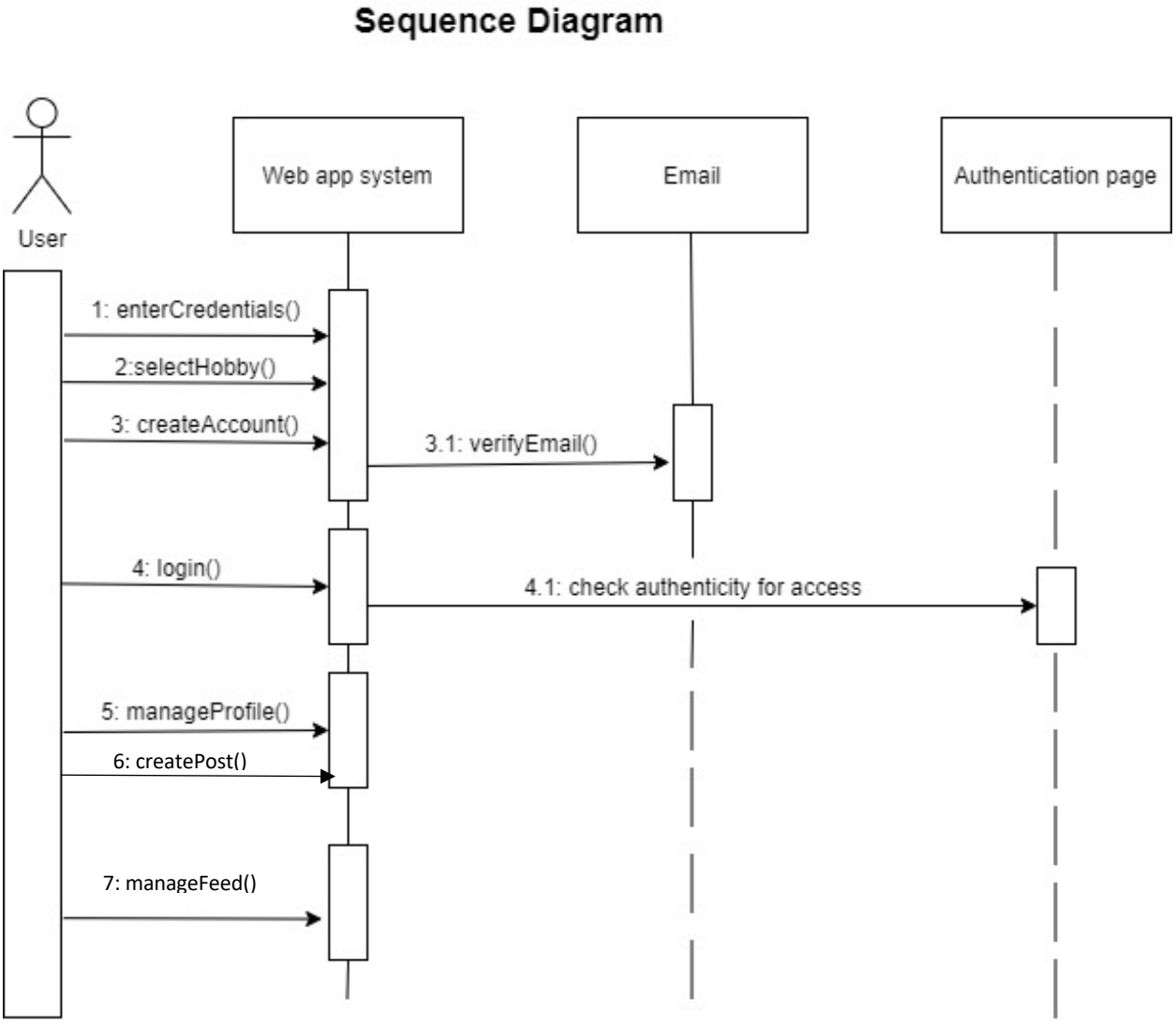
Level 1 :


TAKE USER INFORMATION
AUTHENTICATE USER INFO
USER PERSONAL INFORMATION, USERNAME, PASSWORD
AUTHENTICATE
USER INFORMATION
USER
SELECT HOBBIES FROM LIST
SELECT HOBBIES
HOBBY SELECTED
USER DATABASE
USER INFORMATION
VIEW AND CHANGE PROFILE INFORMATION
PROFILE
CHANGES MADE BY USER
POST CONTENT
POST
CONTENT POSTED BY USER
FETCH CONTENT FROM OTHER DATABASES
SEE IT IN FEED
SEE OTHER USER'S WORKS

6.3

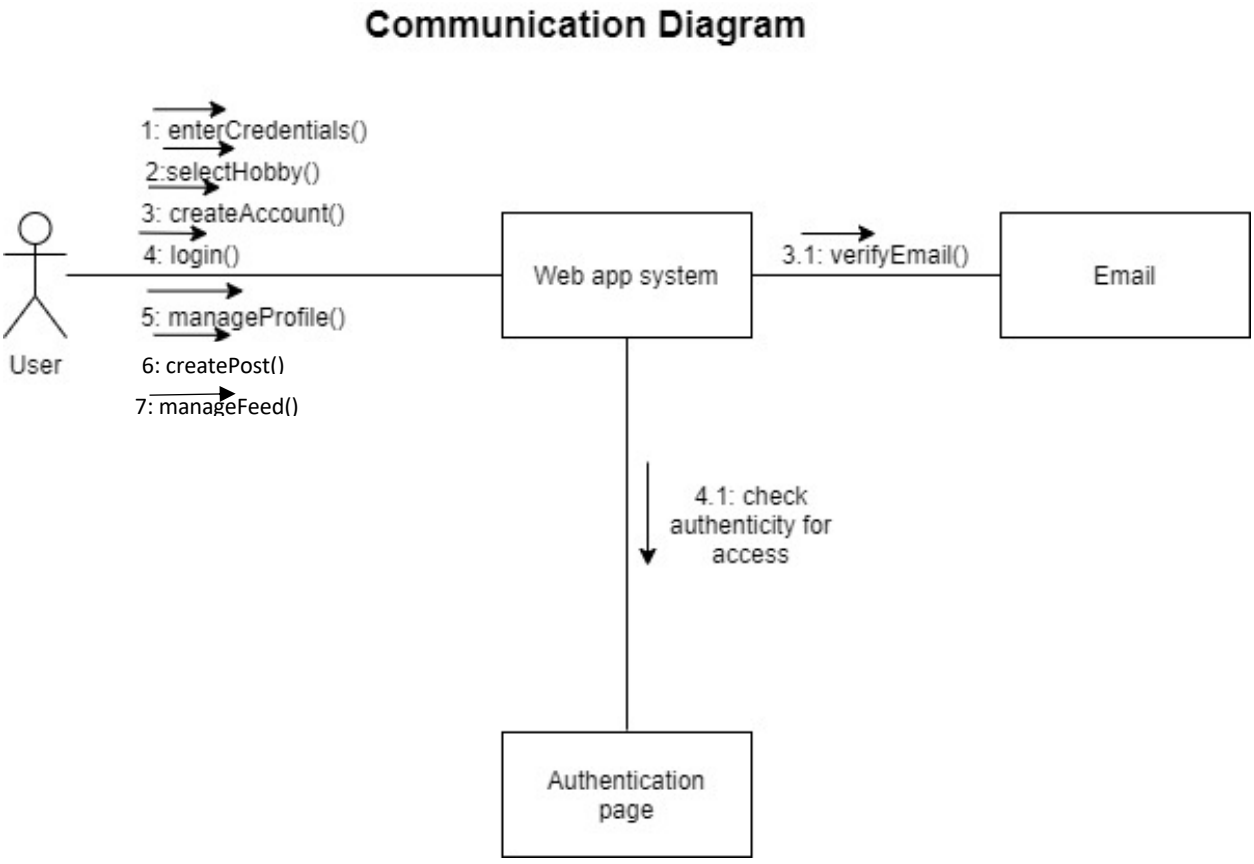

Class Diagram
User
+ name: String
+ email: String
+ password: String
+ username: String
+ hobbies: Array
+ manageFeeds
+ createFeed
+ updateProfile
+ createConnections
+ login
follow
Hobby Category
+ name: String
+ id: String
+ createHobby()
+ assiciateHobbyToUser()
Profile
+ name: String
+ username: String
+ password: String
+ changePassword()
+ manageProfile()
+ changeProfileImage()
Feed
+ heading: String
+ description: String
+ file: File
+ createFeed()
+ getUserFeeds()
+ updateFeed()
+ shareFeed()
+ reactToFeed()
Share
+ postId: String
+ service: String
+ sharePost()

6.4

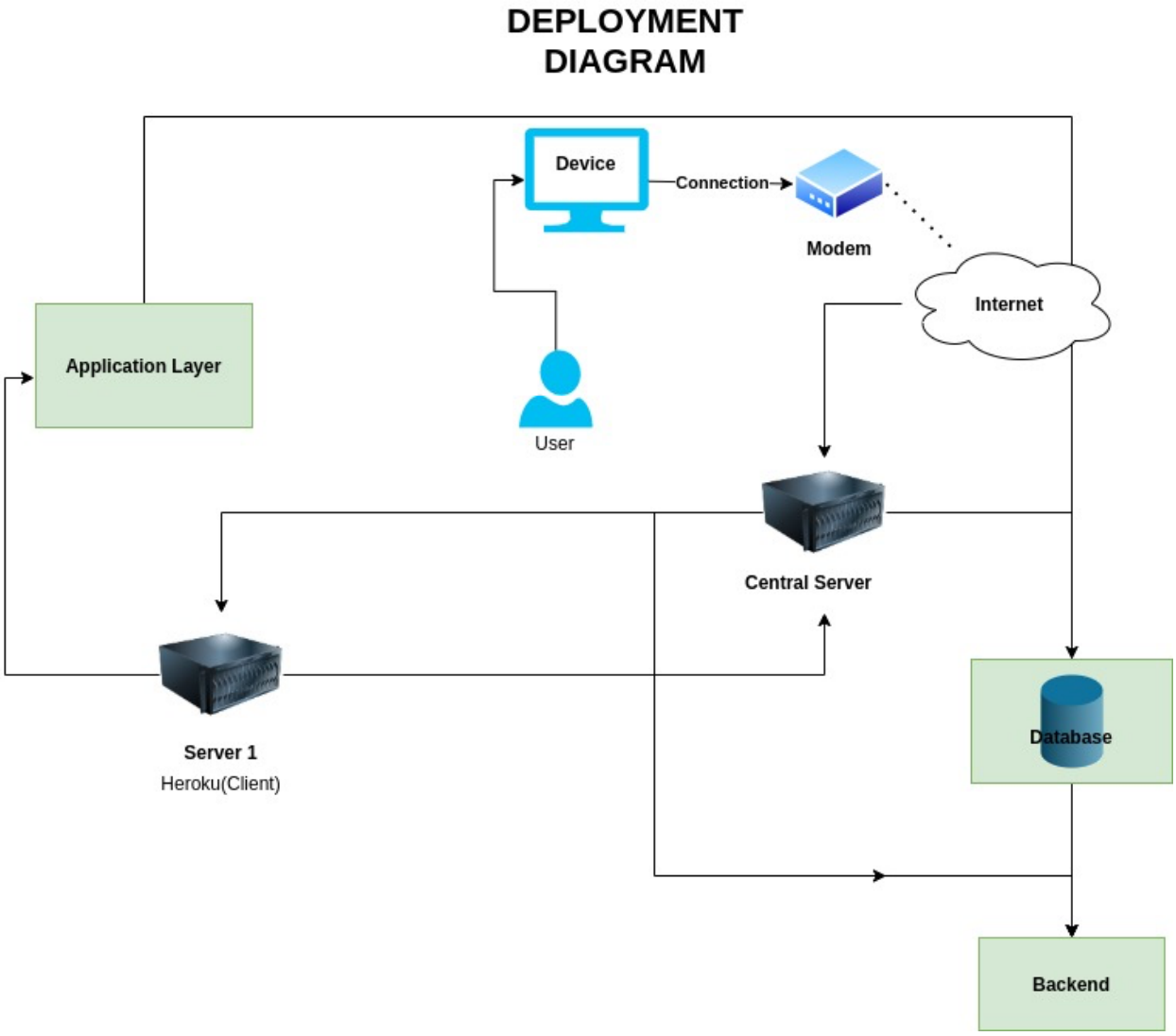

Sequence Diagram
User
Web app system
Email
Authentication page
1: enterCredentials()
2:selectHobby()
3: createAccount()
3.1: verifyEmail()
4: login()
4.1: check authenticity for access
5: manageProfile()
6: createPost()
7: manageFeed()

6: createPost()

6.5



6.6


DEPLOYMENT
DIAGRAM
Device
Connection
Modem
Internet
Application Layer
User
Central Server
Server 1
Heroku(Client)
Database
Backend

7.0 Test Cases

7.1 Test Scenario: Checking Sign Up Functionality

Test Steps:

- Enter Email
- Enter Username
- Enter Password
- Agree with Terms and Conditions
- Click Next

7.1.1 **Test Case 1:** Check result on entering valid email and password

Test Data	Result Required	Actual Result	Pass/ Fail
Email: xyz@gmail.com Username: Aliza321 Password: Xyz@1234 T&C : Agree	Moves to the next page	Moves to the next page	Pass

7.1.2 **Test Case 2:** Check result on entering invalid email or password

Test Data	Result Required	Actual Result	Pass/ Fail
Email: xyz@gmail.com Username: Aliza321 Password: x@14 T&C: Agree	Moves to the next page	Enter valid password	Fail

Test Data	Result Required	Actual Result	Pass/ Fail
Email: xyz@ Username: Aliza321 Password: Xyz@1234 T&C: Agree	Moves to the next page	Enter valid email	Fail

Test Data	Result Required	Actual Result	Pass/ Fail
Email: xyz@gmail.com Username: Aliza Password: Xyz@1234 T&C: Agree	Moves to the next page	Username already exists.	Fail

7.1.3 Test Case 3: When account already exists

Test Data	Result Required	Actual Result	Pass/ Fail
Email: xyz@gmail.com Username: Aliza321 Password: Xyz@1234 T&C: Agree	Moves to the next page	Account already exists	Fail

7.1.4 **Test Case 4:** Check result when not agreed to T&C

Test Data	Result Required	Actual Result	Pass/ Fail
Email: xyz@gmail.com Username: Aliza321 Password: Xyz@1234 T&C: Disagree	Moves to the next page	Agree with the T&C	Fail

7.1.5 **Test Case 5:** Check result when the email or password is empty

Test Data	Result Required	Actual Result	Pass/ Fail
Email: Username: Password: T&C:	Moves to the next page	Enter valid username or password	Fail

7.2 Test Scenario : Checking Select Hobbies Functionality

Test Step:

- Select Hobbies
- Click Sign Up

7.2.1 **Test Case 1:** Checks result on selecting hobbies

Test Data	Result required	Actual Result	Pass/ Fail
Hobbies(List): Acting	Account has been created	Account has been created	Pass

7.2.2 **Test Case 2:** Checks result on not selecting hobbies

Test Data	Result required	Actual Result	Pass/ Fail
Hobbies(List) :	Account has been created	Select a hobby (Error message)	Fail

7.3 Test Scenario : Authentication of Email

Test Step:

- Click on verification link sent to email
- Email Verified

7.3.1 **Test Case 1:** Checks result if link received

Test Data	Result required	Actual Result	Pass/ Fail
Link sent to email	Account has been verified	Account has been verified	Pass

7.3.2 **Test Case 2:** Checks result if link not received

Test Data	Result required	Actual Result	Pass/ Fail
-	Account has been verified	Email not verified	Fail

7.4 Test Scenario: Checking Login Functionality

Test Step:

- Enter Username
- Enter Password
- Click on Login

7.4.1 **Test Case 1:** Checks result on entering valid username and password

Test Data	Result Required	Actual Result	Pass/ Fail
Username: Aliza321 Password: Xyz@1234	Login Successfully	Login Successfully	Pass

7.4.2 **Test Case 2:** Check result on entering invalid username or password

Test Data	Result Required	Actual Result	Pass/ Fail
Username: Aliza321 Password: Xyz@14	Login Successfully	Enter Valid Password	Fail

Test Data	Result Required	Actual Result	Pass/ Fail
Username: Alia1 Password: Xyz@1234	Login Successfully	Enter Valid Username	Fail

7.4.3 **Test Case 3:** Checks result on not entering username or password

Test Data	Result Required	Actual Result	Pass/ Fail
Username: Password:	Login Successfully	Enter Valid Username and Password	Fail

7.5 Test Scenario: Checking Post Functionality

Test Step:

- Enter Content for Post
- Click on Post

7.5.1 **Test Case 1:** Checks result on entering content for post

Test Data	Result Required	Actual Result	Pass/ Fail
Image / Video	Post Created	Post Created	Pass

7.5.2 **Test Case 2:** Checks result when content of post is empty

Test Data	Result Required	Actual Result	Pass/ Fail
-	Post Created	Enter Content	Fail

7.0 Estimation

7.1 Function Point Estimation

7.1.1 Unadjusted Function Point Calculation

Component Type	Complexity of Components									
	Low			Average			High			Total
External Input	1	X3	3	1	X4	4	4	X6	24	31
External Output	1	X4	4	3	X5	15	9	X7	63	82
External Query	1	X3	3	1	X4	4	2	X6	12	19
Internal Logic Files	0	X7	0	2	X10	20	2	X15	30	50
External Interface Files	4	X5	20	2	X7	14	2	X10	20	54
									Total Unadjusted Function Point	236

7.1.2 Value Adjustment Factor Calculation

General System Characteristic		Grade
1	Data communications	5
2	Distributed data processing	3
3	Performance	4
4	Heavily used configuration	2
5	Transaction rate	4
6	On-Line data entry	5
7	End-user efficiency	4
8	On-Line update	5
9	Complex processing	0
10	Reusability	5
11	Installation ease	5
12	Operational ease	4
13	Multiple sites	4
14	Facilitate change	3

VAF= 0.65 + [($\sum_{i=1}^{14}$ Di) X 0.01]

= 0.65+[53*0.01]

= 1.18

Function Point (FP)= UFP X VAF

= 236 * 1.18 = 278.48

7.2 COCOMO Model

Effort (E) = a_b * (KLOC)^{b_b} (in Person-months)

= 2.4 * (30^1.05)

= 85.35 Person-months

Development Time (D) = c_b *(E)^{d_b} (in month)

= 2.5 * (85.35 ^0.38)

= 13.55 months

Average staff size (SS) = E/D (in Person)

= 85.35/13.55

= 6.298 //approximately 6 person

Productivity (P) = KLOC/E (in KLOC/Person-month)

= 30/85.35

= 0.35 KLOC/Person-month

8.0 System Evolution

In the future this system will be able to accommodate a chatting environment and a feature where the users can perform transactions i.e. sell and buy works of their interests by other users.

Appendix A : Glossary

User	A registered person in this site.
Password	A word that enables one to gain admission into the system.
HTTP (Hyper Text Transfer Protocol)	It is a transaction oriented client/ server protocol between a web browser and a web server.
SMTP(Simple Mail Transfer Protocol)	It is a set of communication guidelines that allow software to transmit an electronic mail over the internet.
Actual result	The system status or behaviour after you conduct a test. An anomaly or deviation is when your actual results differ from the expected results.
Expected result	A description of the test object's expected status or behaviour after the test steps are completed. Part of the test case.
Failure	Deviation of the component or system under test from its expected result.
Maintainability	A measure of how easy a given piece of software code is to modify in order to correct defects, improve or add functionality.
Test case	A structured test script that describes how a function or feature should be tested, including test steps, expected results preconditions and postconditions.
Test data	Information that completes the test steps in a test case with e.g. what values to input. In a test case where you add a customer to the system the test data might be customer name and address. Test data might exist in a separate test data file or in a database.