

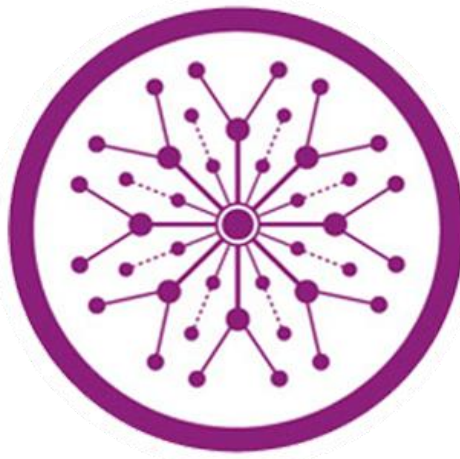
Developer-Connector

Final Year Project

Session 2018-2022

A project submitted in partial fulfillment of the degree of

BS in Computer Science




Department of Computer Sciences

Faculty of Computer Science & Information Technology

The Superior University Lahore

Fall 2021

Type (Nature of project)	<input checked="" type="checkbox"/> Development <input type="checkbox"/> Research <input type="checkbox"/> R&D			
Area of specialization	MERN Stack			
FYP ID	FYP-BCSM-S21-004			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	BCSM-S18-014	Majid Hussain	Majidkhokhar010@gmail.com	
(ii)	BCSM-S18-007	Usama Gulzar	Bcsm-s18-007@superior.edu.pk	
(iii)	BCSM-S18-039	Qaium Saleem	Bcsm-s18-039@superior.edu.pk	


*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I **Majid Hussain** S/D of **Muhammad Shahbaz**, group leader of FYP under registration no. **BCSM-S18-014** at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: **05-01-2021**

Name of Group Leader: **Majid Hussain**

Signature: 

Name of Supervisor: **Mr. Hafiz Naeem**

Co-Supervisor: **Mr. Muddasir Hussain**

Designation: **Lecturer**

Designation: **Lecturer**

Signature: _____

Signature: _____

HoD: Irfan-ud-Din

Signature: _____

Developer-Connector

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		Original Draft	
			Changes Based on Feedback from Supervisor	
			Changes Based on Feedback From Faculty	
			Added Project Plan	
			Changes Based on Feedback from Supervisor	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____

Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____

Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____

Signature: _____

Dedication

We want to dedicate this work to those students who wants to learn something even though they have no resources. We also want to dedicate this project to all of my friends who have crossed this stage without any resources and faced these types of difficulties.

Acknowledgements

We would like to pay our gratitude to the almighty ALLAH who created us with not only the ability to design and program this system but also the power of patience.

We are obliged and thankful to our project supervisor **Sir Hafiz Naeem** Lecturer, Superior University, for his continuous encouragement, motivation and professional guidance during the work of this project which has proven to be an integral part of it. Without his valuable support and guidance, this project could not elevate up this level of development from our point of view.

We would like to express our sincere and cordial gratitude to the people those who have supported us directly, evaluated and criticized our work in several phases during the development of this project and for preparing this dissertation indirectly

Executive Summary

Developer-Connector is web based social application that will allow connecting all types of developers like Software House developers, Freelancers, University IT teachers, IT students and researchers to approach with each other according to their taste of field. They can set up their profiles accordingly with profile photos to complete related information. It is a small social network for developers and persons like them as I mentioned above, user can register and can create a pretty extensive profiles with many fields including education and job experience etc. User can also post and comment on them also like and unlike them. People can buy different tech related products i.e. laptops, processors, new tech gadgets and many more, can be able to generate their invoice to track the order.

Looking toward the problem, we purposed a solution accordingly, we'll going to design a platform that will allow different users to get connect with each other, basically this platform will connect communities as above mentioned like Software houses, Universities, Freelancers, Researchers and Students easily. There they can set up their profiles accordingly with profile photos to complete related information. It is a small social network for developers and persons like them as I mentioned above, user can register and can create a pretty extensive profiles with many fields including education and job experience etc. User can also post and comment on them also like and unlike them.

Table of Contents

PLAGIARISM FREE CERTIFICATE	II
DEDICATION	V
ACKNOWLEDGEMENTS	VI
EXECUTIVE SUMMARY	VII
TABLE OF CONTENTS.....	VIII
LIST OF FIGURES.....	XI
LIST OF TABLES	XII
CHAPTER 1	1
INTRODUCTION	1
1.1. BACKGROUND.....	2
1.2. MOTIVATIONS AND CHALLENGES.....	2
1.3. GOALS AND OBJECTIVES.....	3
1.4. EXISTING SOLUTIONS.....	3
1.5. GAP ANALYSIS	3
1.6. PROPOSED SOLUTION.....	3
1.7. PROJECT PLAN	4
1.7.1. WORK BREAKDOWN STRUCTURE.....	4
1.7.2. ROLES & RESPONSIBILITY MATRIX.....	4
1.7.3. GANTT CHART.....	6
1.8. EMPATHY MAP	7
CHAPTER 2	8
SOFTWARE REQUIREMENT	8
SPECIFICATIONS.....	8
2.1. INTRODUCTION	9
2.1.1. PURPOSE	9
2.1.2. DOCUMENT CONVENTIONS	9
2.1.3. INTENDED AUDIENCE AND READING SUGGESTIONS	10
2.1.4. PRODUCT SCOPE	10
2.2. OVERALL DESCRIPTION	10
2.2.1. PRODUCT PERSPECTIVE.....	10
2.2.2. USER CLASSES AND CHARACTERISTICS.....	10
2.2.3. OPERATING ENVIRONMENT	11
2.2.4. DESIGN AND IMPLEMENTATION CONSTRAINTS	11
2.2.5. ASSUMPTIONS AND DEPENDENCIES.....	11
2.3. EXTERNAL INTERFACE REQUIREMENTS	12
2.3.1. USER INTERFACES	12
THE APPLICATION GUI PROVIDES MENUS, TOOLBARS, BUTTONS, PANES, CONTAINERS, GRIDS ALLOWING FOR EASY CONTROL BY A KEYBOARD AND A MOUSE.	12
2.3.2. HARDWARE INTERFACES	12
2.3.3. SOFTWARE INTERFACES	12
2.3.4. COMMUNICATIONS INTERFACES	12
2.4. SYSTEM FEATURES	12
2.4.1. SIGN-UP	12
2.4.1.1. DESCRIPTION AND PRIORITY.....	12
2.4.1.2. RESPONSE SEQUENCES.....	12
2.4.1.3. FUNCTIONAL REQUIREMENTS.....	12

2.4.2.	LOGIN	13
2.4.2.1.	DESCRIPTION AND PRIORITY	13
2.4.2.2.	STIMULUS/RESPONSE SEQUENCES	13
2.4.2.3.	FUNCTIONAL REQUIREMENTS	13
2.4.3.	POST	13
2.4.3.1.	DESCRIPTION AND PRIORITY	13
2.4.3.2.	STIMULUS/RESPONSE SEQUENCES	13
2.4.3.3.	FUNCTIONAL REQUIREMENTS	13
2.4.4.	UPDATE POST	13
2.4.4.1.	DESCRIPTION AND PRIORITY	13
2.4.4.2.	RESPONSE SEQUENCES.....	13
2.4.4.3.	FUNCTIONAL REQUIREMENT	13
2.4.5.	SEARCH POST	14
2.4.5.1.	DESCRIPTION AND PRIORITY.....	14
2.4.5.2.	RESPONSE SEQUENCES.....	14
2.4.5.3.	FUNCTIONAL REQUIREMENT	14
2.4.6.	COMMENTING POST	14
2.4.6.1.	DESCRIPTION AND PRIORITY.....	14
2.4.6.2.	RESPONSE SEQUENCES.....	14
2.4.6.3.	FUNCTIONAL REQUIREMENT	14
2.4.7.	LIKE POST	14
2.4.7.1.	DESCRIPTION AND PRIORITY.....	14
2.4.7.2.	RESPONSE SEQUENCES.....	14
2.4.7.3.	FUNCTIONAL REQUIREMENT	15
2.4.8.	SAVE POST	15
2.4.8.1.	DESCRIPTION AND PRIORITY.....	15
2.4.8.2.	RESPONSE SEQUENCES.....	15
2.4.8.3.	FUNCTIONAL REQUIREMENT	15
2.4.9.	BUY PRODUCT	15
2.4.9.1.	DESCRIPTION AND PRIORITY.....	15
2.4.9.2.	RESPONSE SEQUENCES.....	15
2.4.9.3.	FUNCTIONAL REQUIREMENT	15
2.4.10.	DOWNLOAD INVOICE.....	16
2.4.10.1.	DESCRIPTION AND PRIORITY.....	16
2.4.10.2.	RESPONSE SEQUENCES.....	16
2.4.10.3.	FUNCTIONAL REQUIREMENT	16
2.4.11.	USER UPDATE	16
2.4.11.1.	DESCRIPTION AND PRIORITY.....	16
2.4.11.2.	RESPONSE SEQUENCES.....	16
2.4.11.3.	FUNCTIONAL REQUIREMENT	16
2.4.12.	CHANGE PASSWORD	16
2.4.12.1.	DESCRIPTION AND PRIORITY.....	16
	RESPONSE SEQUENCES.....	16
2.4.12.2.	FUNCTIONAL REQUIREMENT	16
2.4.13.	DELETE ACCOUNT	17
2.4.13.1.	DESCRIPTION AND PRIORITY.....	17
2.4.13.2.	RESPONSE SEQUENCES.....	17
2.4.13.3.	FUNCTIONAL REQUIREMENT	17
2.5.	NONFUNCTIONAL REQUIREMENTS	17
REFERENCES	17
2.5.1.	PERFORMANCE REQUIREMENTS	17
2.5.2.	SAFETY REQUIREMENTS	17
2.5.3.	SECURITY REQUIREMENTS	17
2.5.4.	USABILITY REQUIREMENTS	17
2.5.5.	RELIABILITY REQUIREMENTS	17
2.5.6.	MAINTAINABILITY/SUPPORTABILITY REQUIREMENTS	18
2.6.	DOMAIN REQUIREMENTS.....	18

CHAPTER 3	19
USE CASE ANALYSIS	19
3.1. USE CASE MODEL.....	21
CHAPTER 4	31
SYSTEM DESIGN	31
4.1. ARCHITECTURE DIAGRAM.....	32
4.2. ENTITY RELATIONSHIP DIAGRAM WITH DATA DICTIONARY	33
4.3. CLASS DIAGRAM.....	35
WHAT IS A CLASS DIAGRAMS USED FOR?.....	36
4.4. SEQUENCE / COLLABORATION DIAGRAM	37
4.5. OPERATION CONTRACTS.....	38
4.6. ACTIVITY DIAGRAM	39
4.7. STATE TRANSITION DIAGRAM.....	40
4.8. COMPONENT DIAGRAM	41
4.9. DEPLOYMENT DIAGRAM	41
4.10. DOMAIN DIAGRAM	42
4.11. DATA FLOW DIAGRAM.....	43
CHAPTER 5	44
IMPLEMENTATION	44
5.1. IMPORTANT FLOW CONTROL/PSEUDO CODES	46
5.2. COMPONENTS, LIBRARIES, WEB SERVICES AND STUBS	47
5.3. DEPLOYMENT ENVIRONMENT	47
5.4. TOOLS AND TECHNIQUES.....	47
5.5. BEST PRACTICES / CODING STANDARDS	47
5.6. VERSION CONTROL	48
CHAPTER 6	49
TESTING AND EVALUATION	49
6.1. USE CASE TESTING.....	50
6.2. EQUIVALENCE PARTITIONING.....	60
6.3. BOUNDARY VALUE ANALYSIS	60
6.4. DATA FLOW TESTING	60
6.5. UNIT TESTING	60
6.6. INTEGRATION TESTING	61
6.7. PERFORMANCE TESTING.....	61
6.8. STRESS TESTING	62
CHAPTER 7	63
SUMMARY, CONCLUSION AND FUTURE ENHANCEMENTS	63
7.1. PROJECT SUMMARY	64
7.2. ACHIEVEMENTS AND IMPROVEMENTS.....	64
7.3. CRITICAL REVIEW.....	64
7.4. LESSONS LEARNT	65
7.5. FUTURE ENHANCEMENTS/RECOMMENDATIONS	65
APPENDICES	66
APPENDIX A: USER MANUAL	67

List of Figures

1.7.3	Gantt Chart	6
3.1	Use Case Diagrams	22
4.1	Architecture Diagram	33
4.2	Entity Relation Diagram	34
4.3	Class Diagram	36
4.4	Sequence Diagram	38
4.6	Activity Diagram	40
4.7	State Transition Diagram	41
4.8	Component Diagram	42
4.9	Deployment Diagram	42
4.10	Domain Diagram	43
4.11	Data Flow Diagram	44

List of Tables

1.7.2	Rolls and responsibility matrix	4
3.2	Use Case Descriptions	24
4.5	Operations Contracts	39

Chapter 1

Introduction

Chapter 1: Introduction

Developer-Connector is web based social application that will allow to connect all types of developers like Software House developers, Freelancers, University IT teachers, IT students and researchers to approach with each other according to their taste of field. It's a web based application that is made to build a strong IT environment between all the people from all over the world no matters either they are students, teachers, developers or researchers. They can find out the best solutions against their queries moreover if they have done with any kind of research they can also share the research work over there with the people. They can set up their profiles accordingly with profile photos to complete related information. It is a small social network for developers and persons like them as I mentioned above, user can register and can create a pretty extensive profiles with many fields including education and job experience etc. User can also post and comment on them also like and unlike them.

1.1. Background

Life is full of science and knowledge produced by the minds of men, The God-given all these minds of inspirations, sciences and the rule, where the science is the light of life, and you know the rights of the creator, and how to communicate with users of the community in the field of engineering, medicine, modern technology and etc. Nowadays IT is becoming an important tool, it is no longer as limited source, but exceeds its position to provide modern techniques and many services. But there is one problem software house developers, freelancer and others could not find the proper way to approach with each other, according to their field of interest. That is why they feel to have such a platform that provide them easy way to approach or get connected with each other. there are other social platforms by that they can interact with each other but they don't know either he/she is a developer other person, after contacting with they get to know he/she is a developer, freelancer, researcher, teacher etc.

1.2. Motivations and Challenges

Motivation to doing this is that, there is no such platform exists that is why we decided to take this project which will help people in future, and the challenge behind this project is as

there is no related project exists so that we were not have a single idea how to start and where to get started. And which things or components should be added in it.

1.3. Goals and Objectives

Goal behind this web based application “Developer-Connector” is to easy approach to the required person like, if someone wants to contact with a particular researcher to discuss some of their articles or others, is to approach him/her easily just by registering on our website. Also they can post to get connect with required person.

1.4. Existing Solutions

There is no such a proper existing platform or solution regarding this problem, but there is a platform known as “**LinkedIn**” although it connects but this is not exactly for **IT Persons** there is a lot more other fields also it is basically for getting a job.

1.5. Gap Analysis

If we talk about the gap analysis, I would like to say nothing is hundred percent perfect at start with the passage of time things are getting better, and our plan is to launch this web based application with such convenient way that everyone would like it and show some eager to use it. We launched this in the competition of “**LinkedIn**” although it is not our direct competitor.

1.6. Proposed Solution

Looking toward the problem, we purposed a solution accordingly, we'll going to design a platform that will allow different users to get connect with each other, basically this platform will connect communities as above mentioned like Software houses, Universities, Freelancers, Researchers and Students easily. There they can set up their profiles accordingly with profile photos to complete related information. It is a small social network for developers and persons like them as I mentioned above, user can register and can create a pretty extensive profiles with many fields including education and job experience etc. User can also post and comment on them also like and unlike them.

1.7. Project Plan

Divided the project into various parts to work on it, in easy and understandable way, different tasks and parts are assigned to the group members to perform their tasks which is well explained in the next section.

1.7.1. Work Breakdown Structure

Work breakdown structure will be clearly and well explained in the next field.

In next section it is mention how and who performs which task.

1.7.2. Roles & Responsibility Matrix

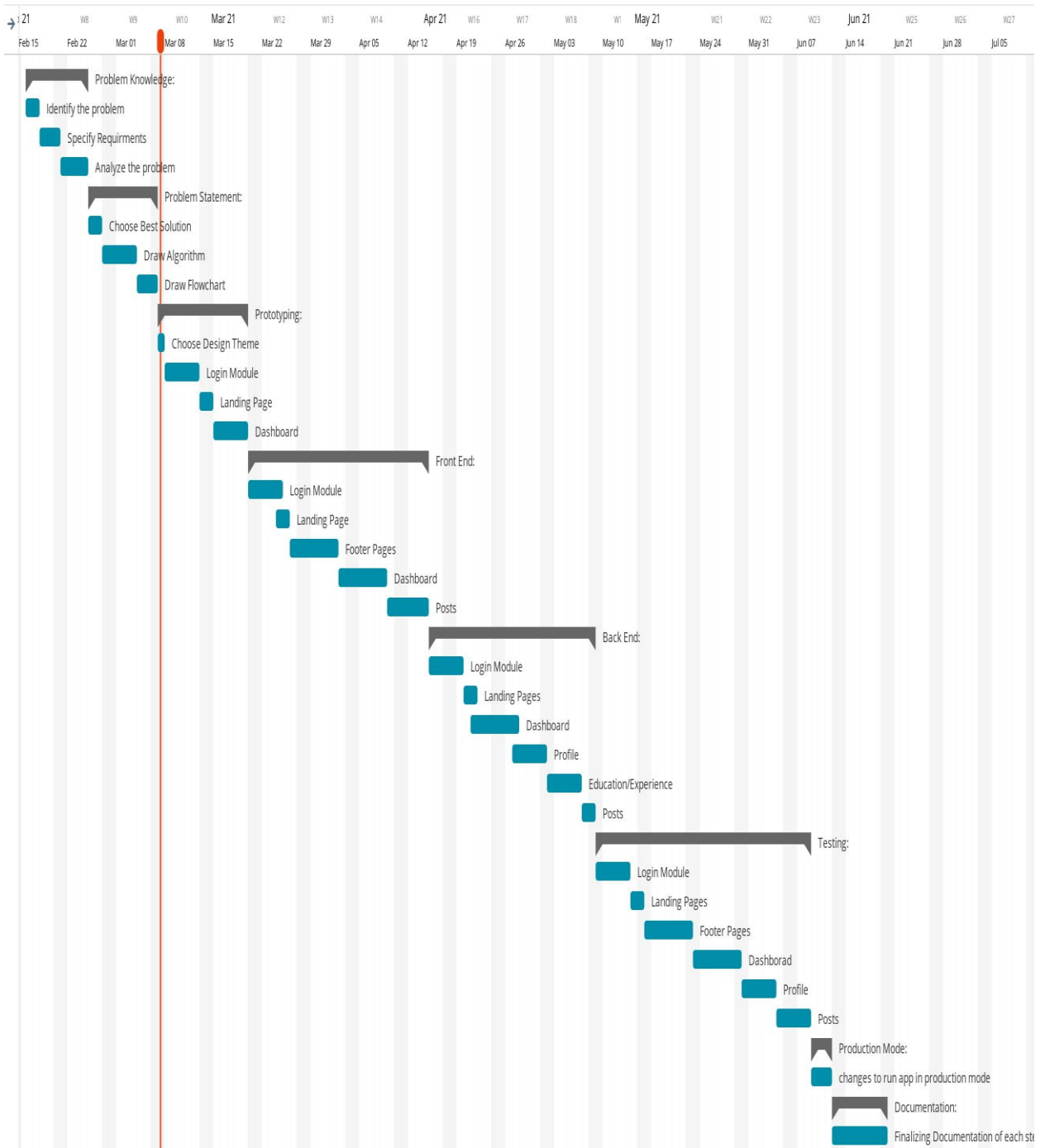
WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Problem Knowledge	1.1	Identify the problem	1	Majid
		1.2	Specify requirements	2	Usama
		1.3	Analyze the problem	4	Majid
2	Problem Solution	2.1	Choose best Solution	2	Majid/Usama
		2.2	Draw Algorithm	5	Usama
		2.3	Draw Flowchart	3	Majid/Qaium
3	Prototyping	3.1	Choose Design theme	1	Majid
		3.2	Login Module	5	Majid/Qaium
		3.3	Landing Page	2	Usama
		3.4	Dashboard	5	Majid
4	Front-End	4.1	Login Module	5	Usama
		4.2	Landing Page	2	Majid
		4.3	Footer Pages	7	Majid/Qaium
		4.4	Dashboard	7	Majid
		4.5	Posts	6	Majid/Usama
5	Back-End	5.1	Login Module	5	Majid
		5.2	Landing Page	2	Majid
		5.3	Dashboard	7	Majid

		5.4	Profile	5	Majid
		5.5	Education/Experience	5	Majid
		5.6	Posts	6	Majid
6	Testing	6.1	Login Module	5	Usama
		6.2	Landing Page	2	Usama
		6.3	Footer Pages	7	Qaium
		6.4	Dashboard	7	Majid
		6.5	Profile	5	Majid
		6.6	Posts	5	Majid
7	Production Mode	7.1	changes to run app in production mode	2	Majid/Usama
8	Documentation	8.1	Finalizing Documentation of each step	6	Majid/Usama/Qaium

1.7.3. Gantt Chart

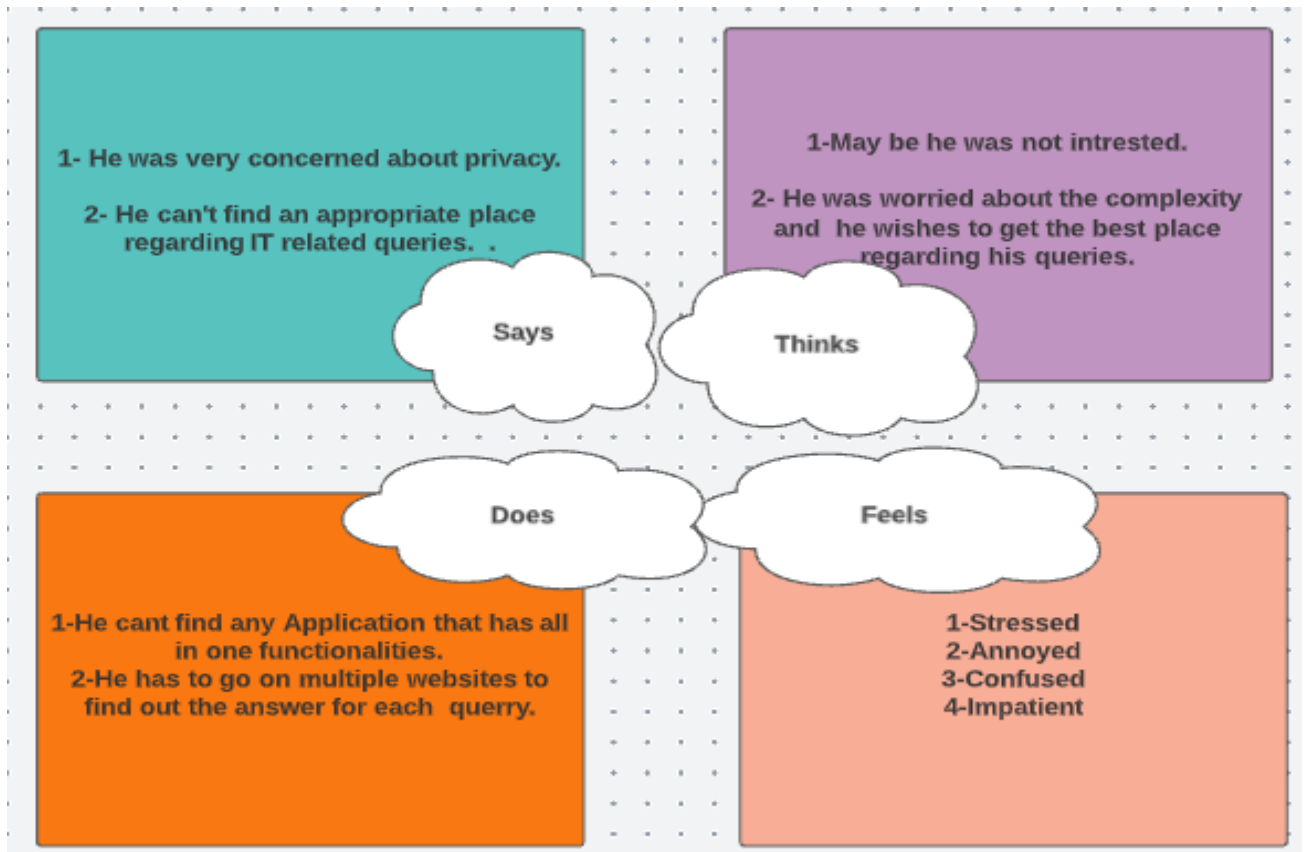
FYP

Read-only view, generated on 07 Mar 2021



1.8. Empathy Map

Empathy Map:



Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

Developer-Connector is web based social application that will allow to connect all types of developers like Software House developers, Freelancers, University IT teachers, IT students and researchers to approach with each other according to their taste of field. It's a web based application that is made to build a strong IT environment between all the people from all over the world no matters either they are students, teachers, developers or researchers. They can find out the best solutions against their queries moreover if they have done with any kind of research they can also share the research work over there with the people. They can set up their profiles accordingly with profile photos to complete related information. It is a small social network for developers and persons like them as I mentioned above, user can register and can create a pretty extensive profiles with many fields including education and job experience etc. User can also post and comment on them also like and unlike them.

2.1.1. Purpose

The problem is, software house developers, freelancer and others could not find the proper way to approach with each other, according to their field of interest. That is why we feel that, we have such a platform that connects all of the above mentioned persons in a convenient way.

2.1.2. Document Conventions

Conventions	Meanings & Explanation
1. DB	Database
2. SRS	Software Requirements Specifications
3.ERD	Entity Relationship Diagram
4. UI	User Interface
5. HI	Hardware Interface
6. SI	Software Interface
7. CI	Communications Interface
8. OS	Operating System

2.1.3. Intended Audience and Reading Suggestions

This document is written for the researchers, project managers, programmers, designers, developers, documentation writers, users involved in the website modification of online website of Developer-Connector to connect all developers, IT teachers, freelancers etc.

2.1.4. Product Scope

The problem is, software house developers, freelancer and others could not find the proper way to approach with each other, according to their field of interest. That is why we feel that, we have such a platform that connects all of the above mentioned persons in a convenient way.

Goal behind this web based application “Developer-Connector” is to easy approach to the required person like, if someone wants to contact with a particular researcher to discuss some of their articles or others, is to approach him/her easily just by registering on our website. Also they can post to get connect with required person.

If we talk about the acceptance of the website, it will be accepted by the users because we saw users do not prefer the verifications to use any website so that in our website before getting registered there is an option where random user can see all the users which are already registered on website and by clicking on the specific persons profile to show his/her complete profile, and the other thing to acceptance is this is the platform especially designed for IT person rather they belong to software houses, students, researchers, freelancers or university teachers.

2.2. Overall Description

2.2.1. Product Perspective

The product is supposed to be an open source, under the GNU general Public License. It is a web based system implementing client-server model.

The system allows the user to create their accounts in the system and provide features of updating and viewing profiles.

2.2.2. User Classes and Characteristics

Users of the system should be able to see posts of different users who posts on the web application, and comment on the post, like and dislike the post as well. User should be able

to signup, sign in, sign out, if user is logged in in that only case he is able to see the posts, means authorization.

2.2.3. Operating Environment

This is a web based system and hence will require the operating environment for a client and server GUI. This will be operating in the following operating environment:

2.2.4. Design and Implementation Constraints

- Information of all the user should be stored in the database.
- This application is running 24 hours.
- User can access this from anywhere from any computer
- User must have their correct email addresses and passwords matching to it to use this application.

2.2.5. Assumptions and Dependencies

This software highly depends on type and version of browser being installed in the system i.e. browser version should be used which have HTML5 support.

Various dependencies will be used throughout in this project, but for now we used and install some of them which are,

- Body-Parser
- Cookie-Parser
- Dotenv
- Express
- JWT
- Express jwt
- Express-router
- Express-validator
- Nodemon
- Uuid
- Morgan
- Mongoose and will be used many more.

2.3. External Interface Requirements

2.3.1. User Interfaces

The application GUI provides menus, toolbars, buttons, panes, containers, grids allowing for easy control by a keyboard and a mouse.

2.3.2. Hardware Interfaces

Device should be enabled

2.3.3. Software Interfaces

The user's browser should be HTML5 compatible for a satisfactory user experience.

2.3.4. Communications Interfaces

For the communication protocols the program needs these protocols to be installed:

Tcp for the client to connect the server in online mode

Storing devices (flash, optical disks etc.) for the client to take a test in offline mode

2.4. System Features

Main features are included yet which are given below.

2.4.1. SIGN-UP

2.4.1.1. Description and Priority

To use the application user should must sign up before using this application, because without signing up user is unable to Login which costs user cannot see posts, submit post, follow or unfollow anyone all that I am saying is to perform various function user must be signup.

2.4.1.2. Response Sequences

User must go to web page and click on register or signup button to sign up for application use, then user must provide his/her Name, Email, Password according to the requirements, after signing up user will be able to login and join the family.

2.4.1.3. Functional Requirements

User must have to enter his/her full name, Valid E-mail ID must contain with "@", for password user must set password with the length of 8 characters along with at least one numerical number.

2.4.2. Login

2.4.2.1. Description and Priority

Login is must to get into the application or get connected by other people.

2.4.2.2. Stimulus/Response Sequences

First you have to put your email and password to get signed up to use this application. Secondly email and password should be valid, means it meets the required criteria of them.

2.4.2.3. Functional Requirements

A user should be authorized to perform other functions of the application, like user profile, posts, post react like comment, like, dislike etc.

2.4.3. Post

2.4.3.1. Description and Priority

User can post after he/she signed up to the application. And post will be signed a unique id.

2.4.3.2. Stimulus/Response Sequences

To make a post user must add the post title and the post body, with some restrictions

2.4.3.3. Functional Requirements

like for title there must be a 4 to 150 characters otherwise error occurs that “title should be between 4 to 150 characters”, and same process will go for the body but in that case restriction is 10 to 2000 characters.

2.4.4. Update Post

2.4.4.1. Description and Priority

If any user have posted any post, and afterwards he/she wants to update any of information he/she can update easily.

2.4.4.2. Response Sequences

User must go to web page and click on update post to make changes in it user will be verified either he/she is the same user, then user can see the changes he/she made.

2.4.4.3. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.4.5. Search Post

2.4.5.1. Description and Priority

If any user have posted any post, and someone wants to search any particular post, so that he/she can go to the search bar to find that post.

2.4.5.2. Response Sequences

User must go to web page and click on the search bar where he/she can enter the keywords to find that particular post for their ease.

2.4.5.3. Functional Requirement

User can be logged in and should be authorized to perform this function. This is not compulsory.

2.4.6. Commenting Post

2.4.6.1. Description and Priority

If any user have posted any post, and afterwards he, or someone (user- he/she) else on the platform can comment on any post/blog, to convey his/her feelings or opinion to the owner of the post.

2.4.6.2. Response Sequences

User must go to web page and click on the post he/she wants to comment on, then user can type his/her thoughts about the post in the comment section and hit the comment button to post the comment.

2.4.6.3. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.4.7. Like Post

2.4.7.1. Description and Priority

If any user have posted any post, and afterwards he, or someone (user- he/she) else on the platform can smash a like on any post/blog.

2.4.7.2. Response Sequences

User must go to web page where all the post are showing up, you can hit the like button showing up on the post to like that post.

2.4.7.3. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.4.8. Save Post

2.4.8.1. Description and Priority

If any user wants to save anyone's post/blog, he/she simply needs to go to the platform and see the posts where he/she can smash the save button to save that post into his/her saved section, where he/she can easily access to those posts without wasting their precious time in search of that post.

2.4.8.2. Response Sequences

Once a user hit the save button, mentioned on the post that post will be listed into the saved section for that particular account.

2.4.8.3. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.4.9. Buy Product

2.4.9.1. Description and Priority

If any user wants to buy any tech gadget, from our platform, he/she can easily buy any product with lower rates than the online market.

2.4.9.2. Response Sequences

User must go to valid web page and click shop button mentioned in the side bar to see the available products that are listed on the website, he/she can select any product, to add that product into the cart, after adding into the cart user must have to fill the required details to purchase it. After placing the order it shows in the dashboard's orders list.

2.4.9.3. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.4.10. Download Invoice

2.4.10.1. Description and Priority

If a user has ordered any product, and he/she wants to have the record for that product he/she can download that product's invoice for the record.

2.4.10.2. Response Sequences

User must go to web page and click on dashboard where he/she will be shown by two options, user must select the orders section to see the status of his/her orders and to download the invoice on any order, user just needs to click on eye button to view the details and then hit a download button which will leads to the download the invoice for that order.

2.4.10.3. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.4.11. User Update

2.4.11.1. Description and Priority

If any user wants to update any of information like name, email he/she can update easily.

2.4.11.2. Response Sequences

User must go to web page and click on update post to make changes in it user will be verified either he/she is the same user, then user can see the changes he/she made.

2.4.11.3. Functional Requirement

User must be logged in and should be authorized in order to update his/her name or email.

2.4.12. Change Password

2.4.12.1. Description and Priority

If any user wants to change his/her accounts' password they can easily change their passwords.

Response Sequences

User must go to web page and click on setting button, there user can be able to change their password, by putting their old password, and new password and then hit save to update.

2.4.12.2. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.4.13. Delete Account

2.4.13.1. Description and Priority

If any user wants to delete his/her account he/she can delete account immediately.

2.4.13.2. Response Sequences

User must go to web page and click on setting, there he/she can see the button named "Delete Account", simply hit that button and your account will be deleted instantly.

2.4.13.3. Functional Requirement

User must be logged in and should be authorized to perform this function.

2.5. Nonfunctional Requirements

References

IEEE standard document for software requirement specification.

2.5.1. Performance Requirements

The system is interactive and the delays involved are less. So in every action-response of the system, there are no immediate delays. Popping error messages and saving the settings there is delay much below 2 seconds.

2.5.2. Safety Requirements

Information transmission will be securely transmitted to server without any changes in information.

2.5.3. Security Requirements

The main security concern is for users account hence proper login mechanism should be used to avoid hacking. The tablet id registration is way to spam check for increasing the security.

2.5.4. Usability Requirements

As the system is easy to handle and navigates in the most expected way with no delays. In that case the system program reacts accordingly and transverses quickly between its states.

2.5.5. Reliability Requirements

The system provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

2.5.6. Maintainability/Supportability Requirements

If we talk about the maintainability of the application it can be maintainable.

2.6. Domain Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Chapter 3

Use Case Analysis

Chapter 3: Use Case Analysis

Functional requirements of the system are described using use case view. The Following use case diagrams summarize the functional requirements of the system and explain it in more convenient way.

Chapter 3: Use Case Analysis

List of Actors:

- Administrator
- Authenticated User
- Un authenticated User
- Help line staff

List of Use Cases:

Admin:

- Login
- Forget Password
- Manage User
- Add User
- Delete User
- Update User

User (Authenticated):

- Create Profile
- Login
- Forget Password
- Update Profile
- Edit post
- Search post

Search

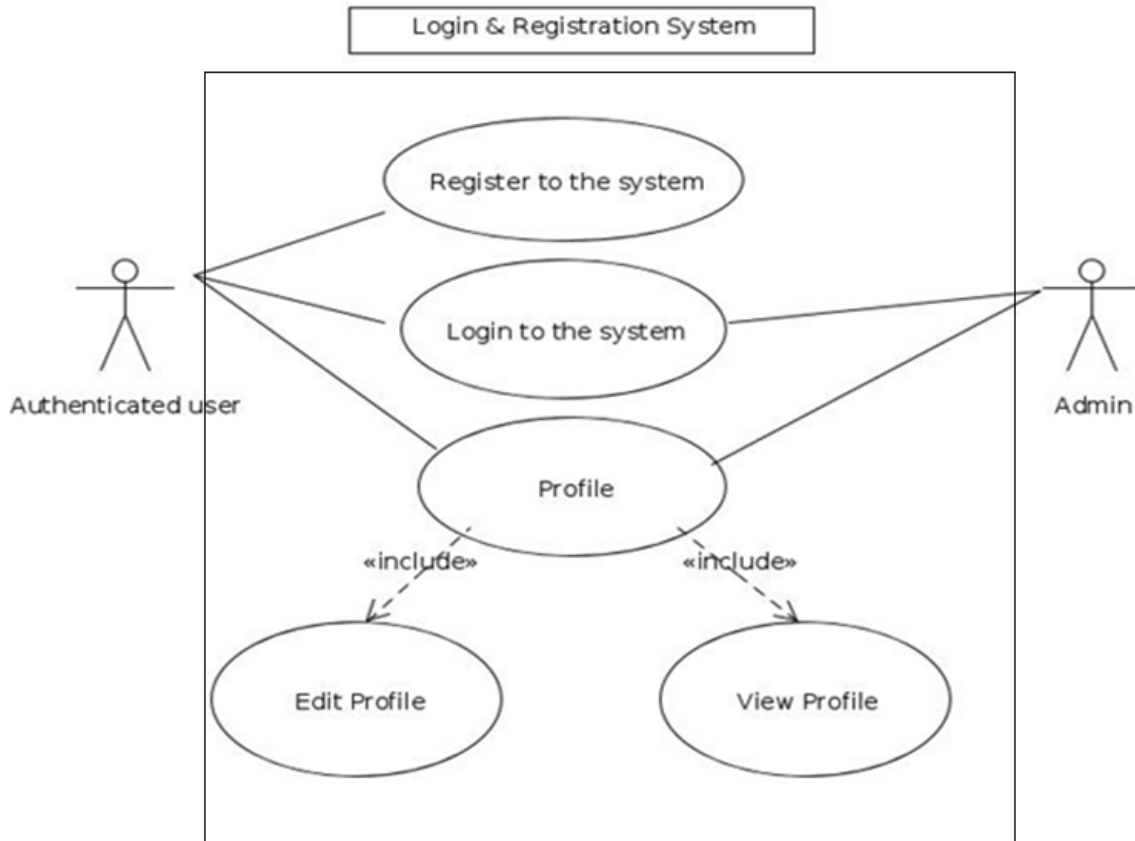
- By Name
- By designation

Customer Support:

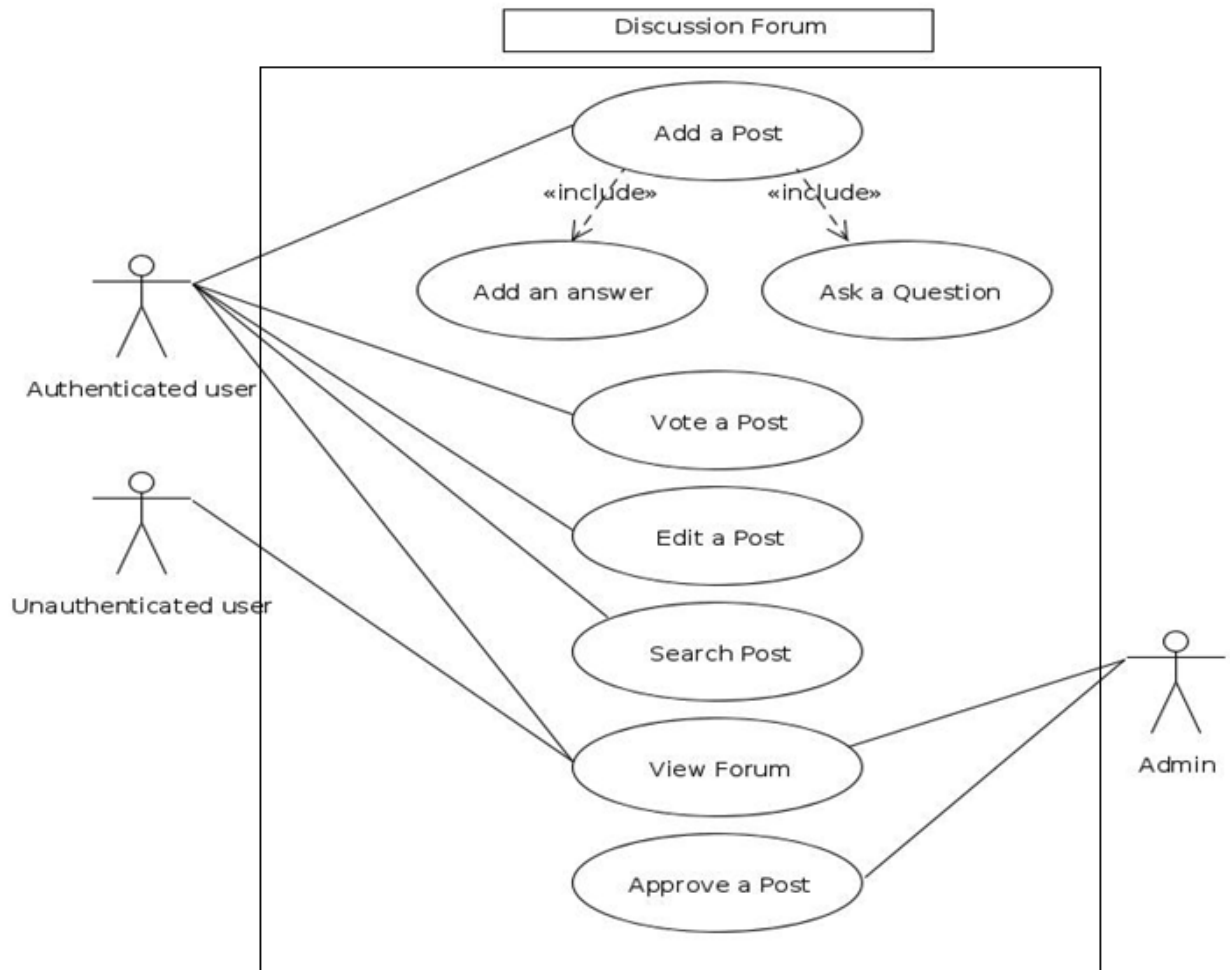
- Login
- Forget Password
- Via Email System

3.1. Use Case Model

Use case diagram of Login and signup or registration system.

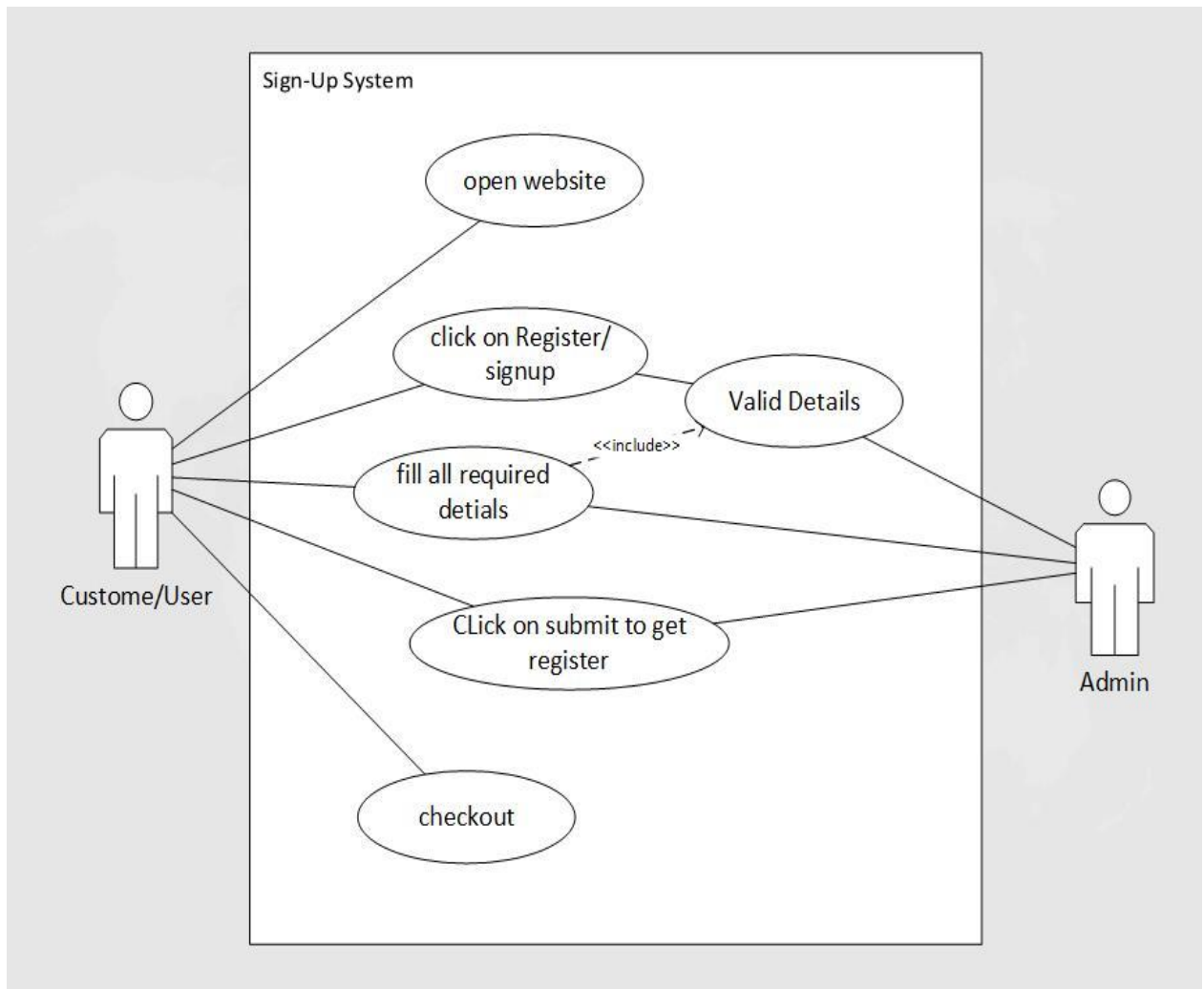


Use case diagram for communication System:



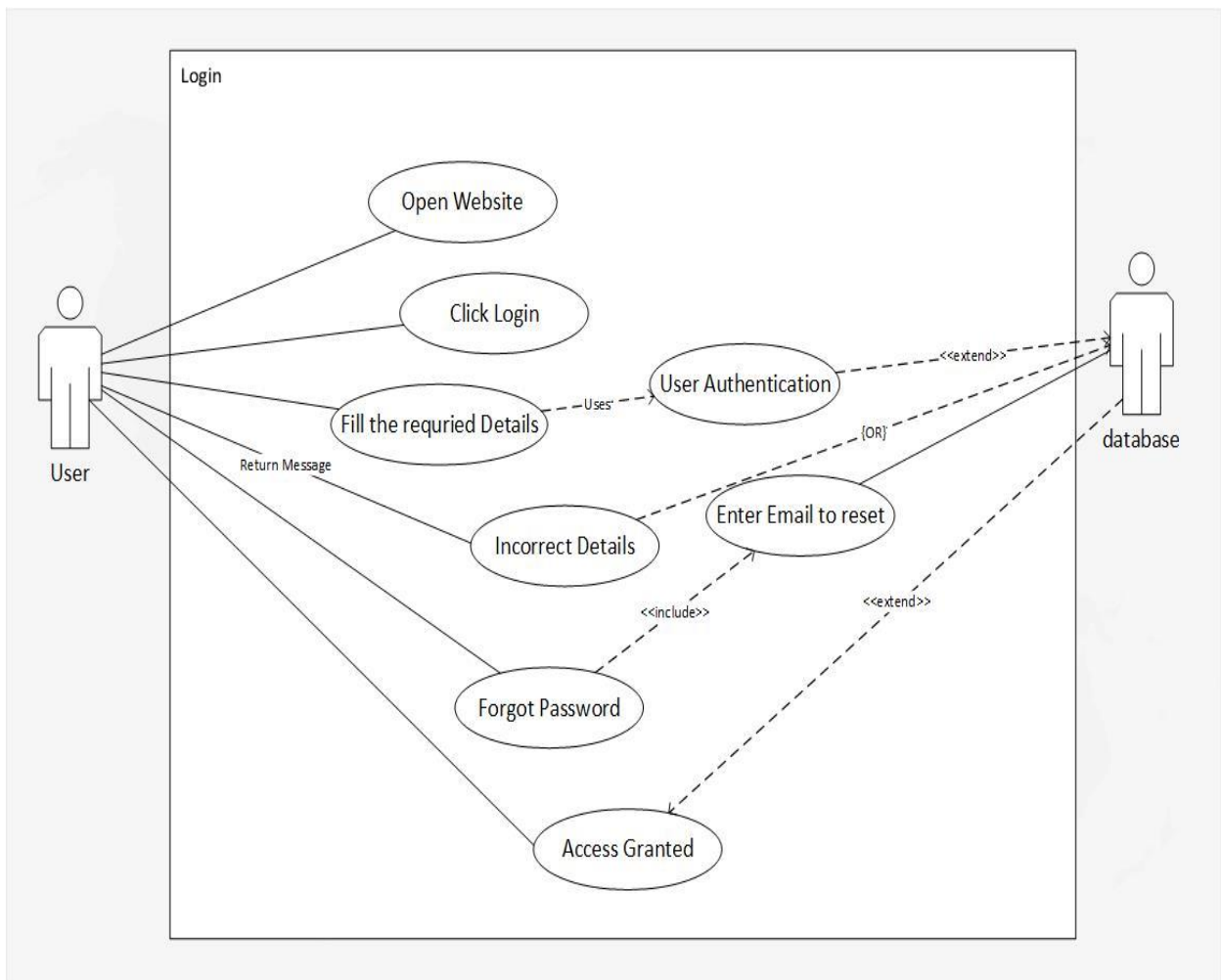
3.1

Use case Name	Signup
Description	User can make his personal account to access the functionalities of the system.
Actor	User
Scope	User can get his own profile to access the system.
Pre-condition	User must have to put valid information.
Post-condition	After signup, user get the notification of successful signup.



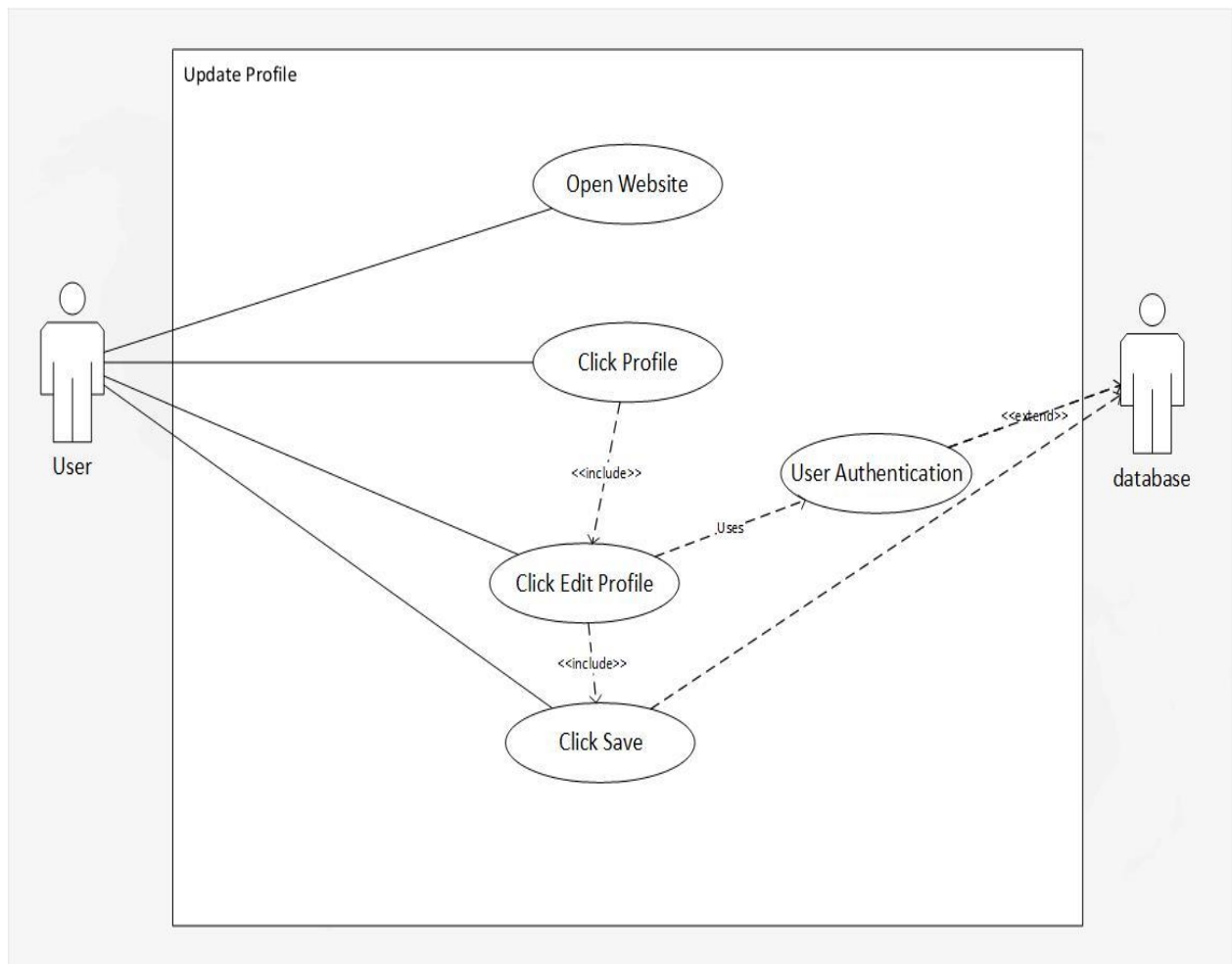
3.2

Use case Name	Login
Description	User can enter his personal account to access the functionalities of the system.
Actor	Authenticated User, Admin
Scope	User can login to his personal account.
Pre-condition	User must have to create an account first with valid information.
Post-condition	User data stored in DB will be shown against his login ID.



3.3

Use case Name	Profile Update
Description	User can enter his personal account to updates his/her profile.
Actor	Authenticated User
Scope	User will update the data according to his needs.
Pre-condition	User must have an account to login to system, with valid URL
Post-condition	Data will be stored and allocated in the system's database. Also get notified that data has been updated.



3.4

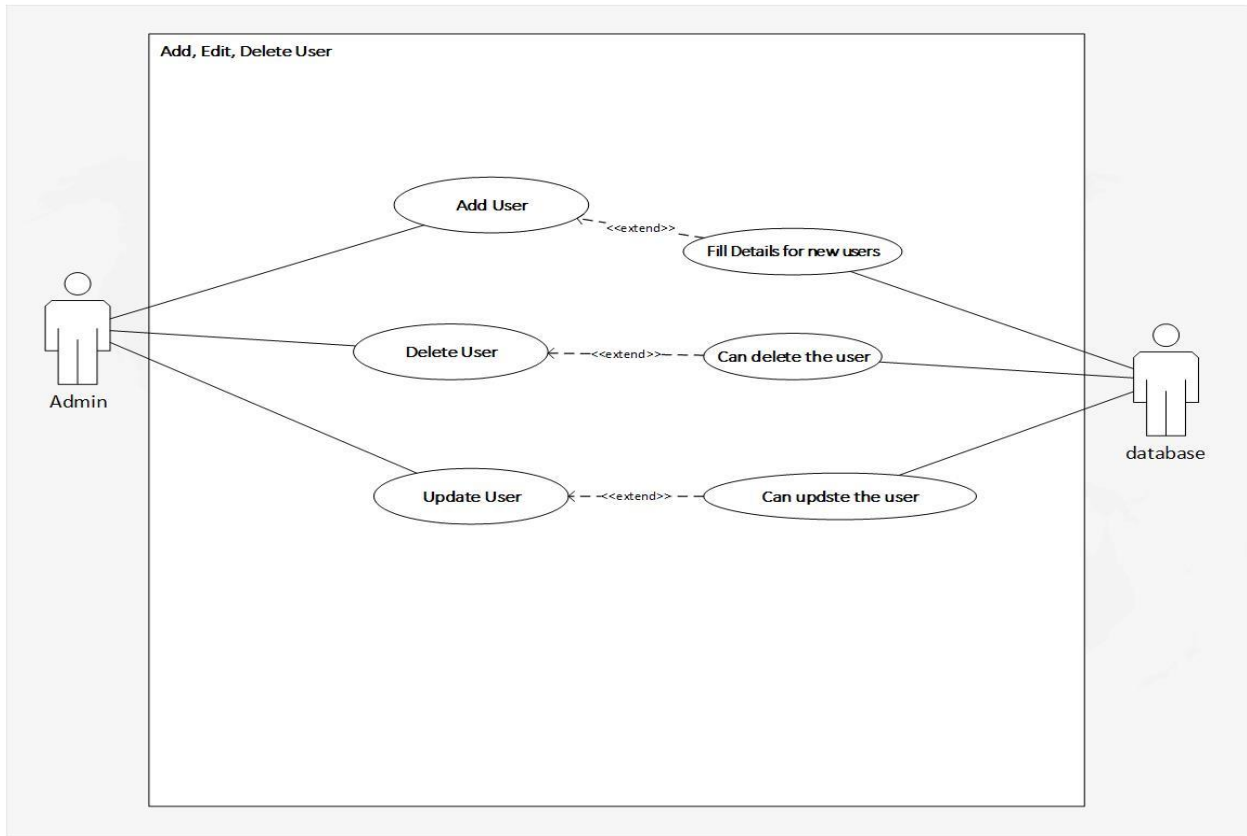
Use case Name	Add User
Description	Admin can be able to add any user into the system.
Actor	Admin
Scope	User will be available to the platform after added.
Pre-condition	Admin must have fill valid information in order to add any user.
Post-condition	User can access the system through his profile, Admin will be notified

3.5

Use case Name	Delete User
Description	Admin can be able to delete any user from the system.
Actor	Admin
Scope	User will be removed from the platform after getting deleted.
Pre-condition	Admin must access to the valid URL in order to delete the user.
Post-condition	Admin will be notified that, user has been removed.

3.6

Use case Name	Update User
Description	Admin can be able to update any user from the system.
Actor	Admin
Scope	User will be updated after updating user's information.
Pre-condition	Admin must access to the valid URL in order to update any user.
Post-condition	Admin will be notified that, user has been updated.



3.7

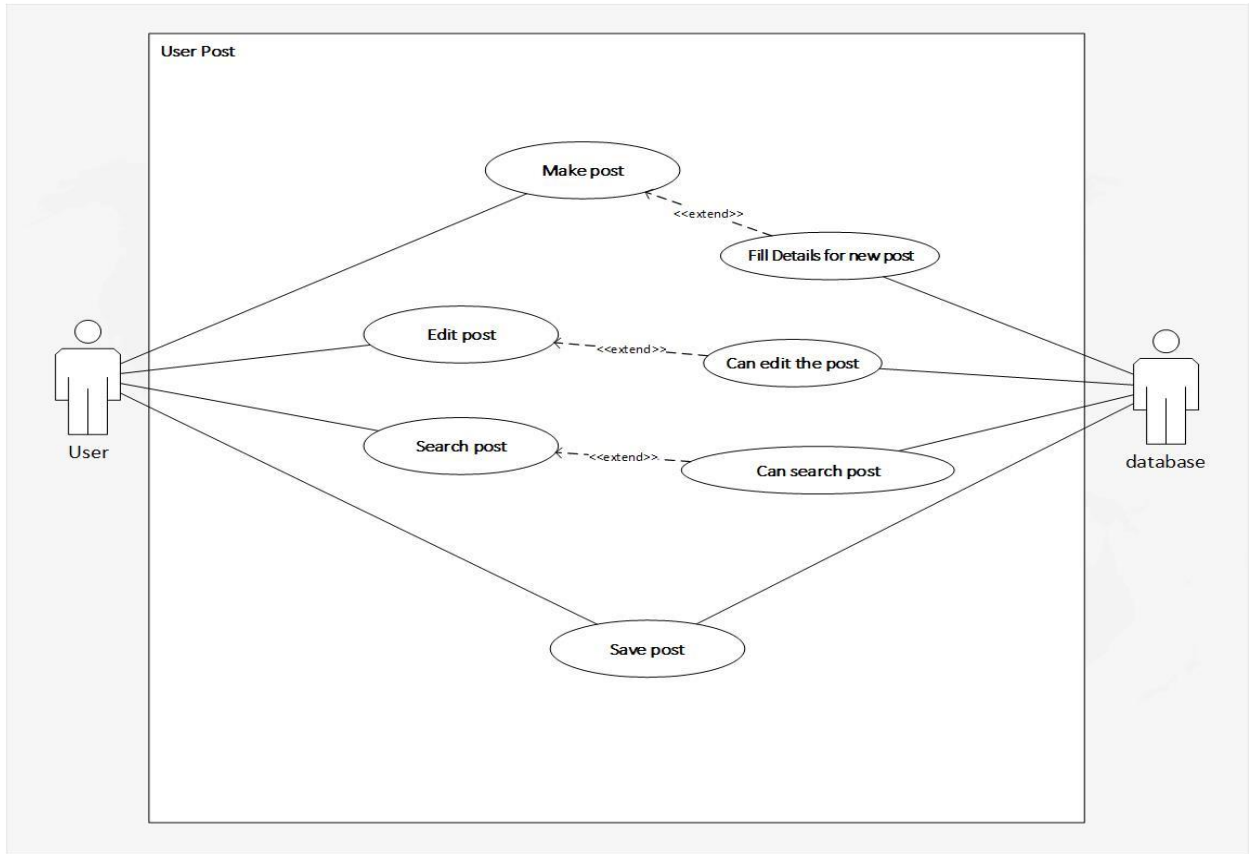
Use case Name	Post
Description	User can post anything on his / her feed.
Actor	User
Scope	User can login to the system to make a post visible to everyone.
Pre-condition	User’s system must be connected to network while using application.
Post-condition	User get notified after any operation is done by him/her.

3.8

Use case Name	Edit Post
Description	User can edit his / her post
Actor	User
Scope	User can update his/her posts that he/she made.
Pre-condition	User enters the required data to edit post, with valid URL.
Post-condition	Post edited and data stored/allocated in DB, and updated to the wall.

3.9

Use case Name	Search Post
Description	User can search any post
Actor	User
Scope	Post searched according his / her interest, using related keywords.
Pre-condition	User enters the required data to search post.
Post-condition	Post searched against the data user provided.

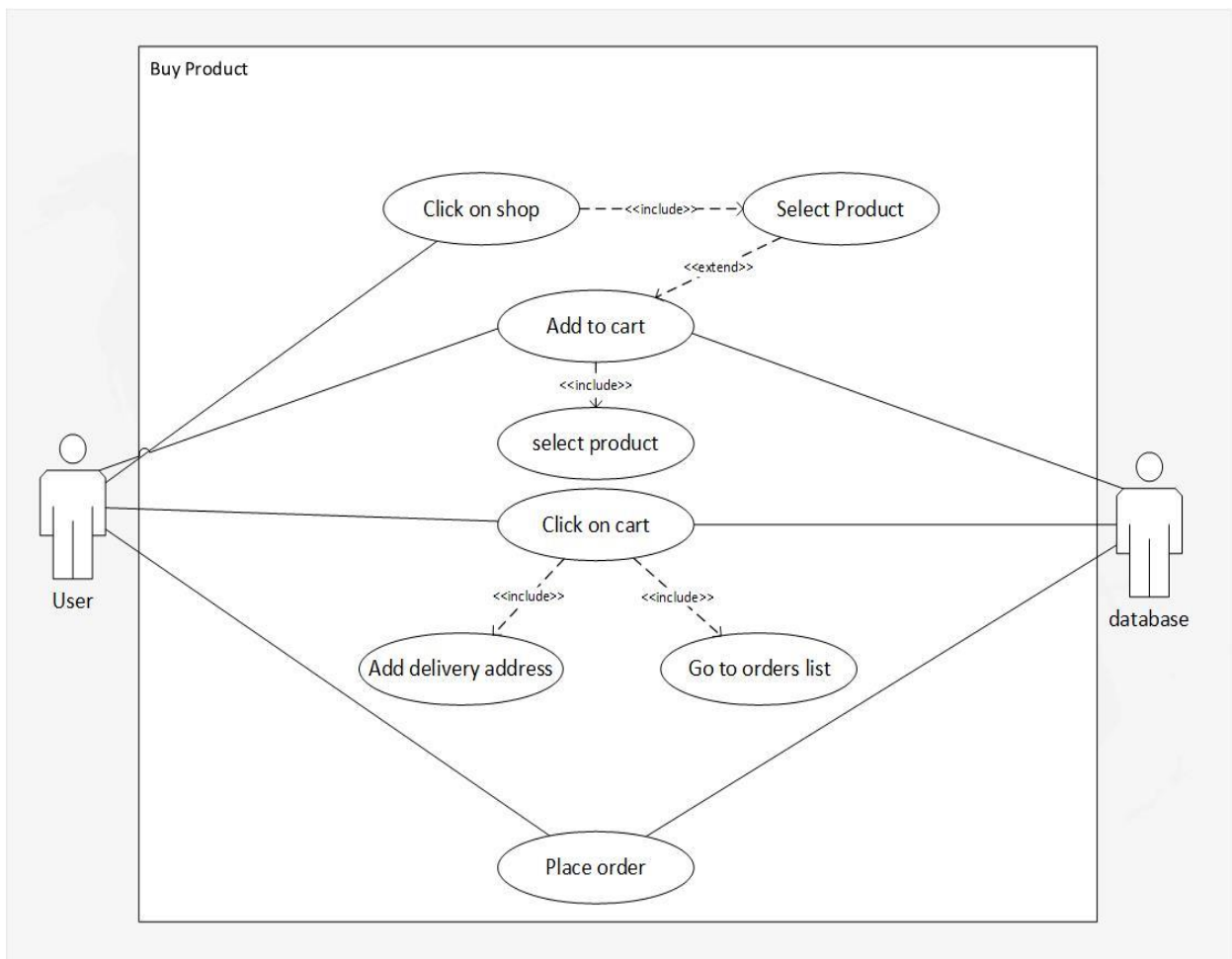


3.10

Use case Name	Update Password
Use case Description	User can update his / her password
Actors	Authenticated User
Pre-Condition	User must have to enter valid old password to set new one.
Post-Condition	User get notified after updating password, and will be stored in DB

3.11

Use case Name	Buy Product
Use case Description	A user can go to the shop and select any product whatever he/she wants, then proceed to the buying process.
Actors	Users
Pre-Condition	User's system must be connected to network along with the valid URL so that user can proceed further easily.
Post-Condition	User get notified after adding product, and make purchase from platform.



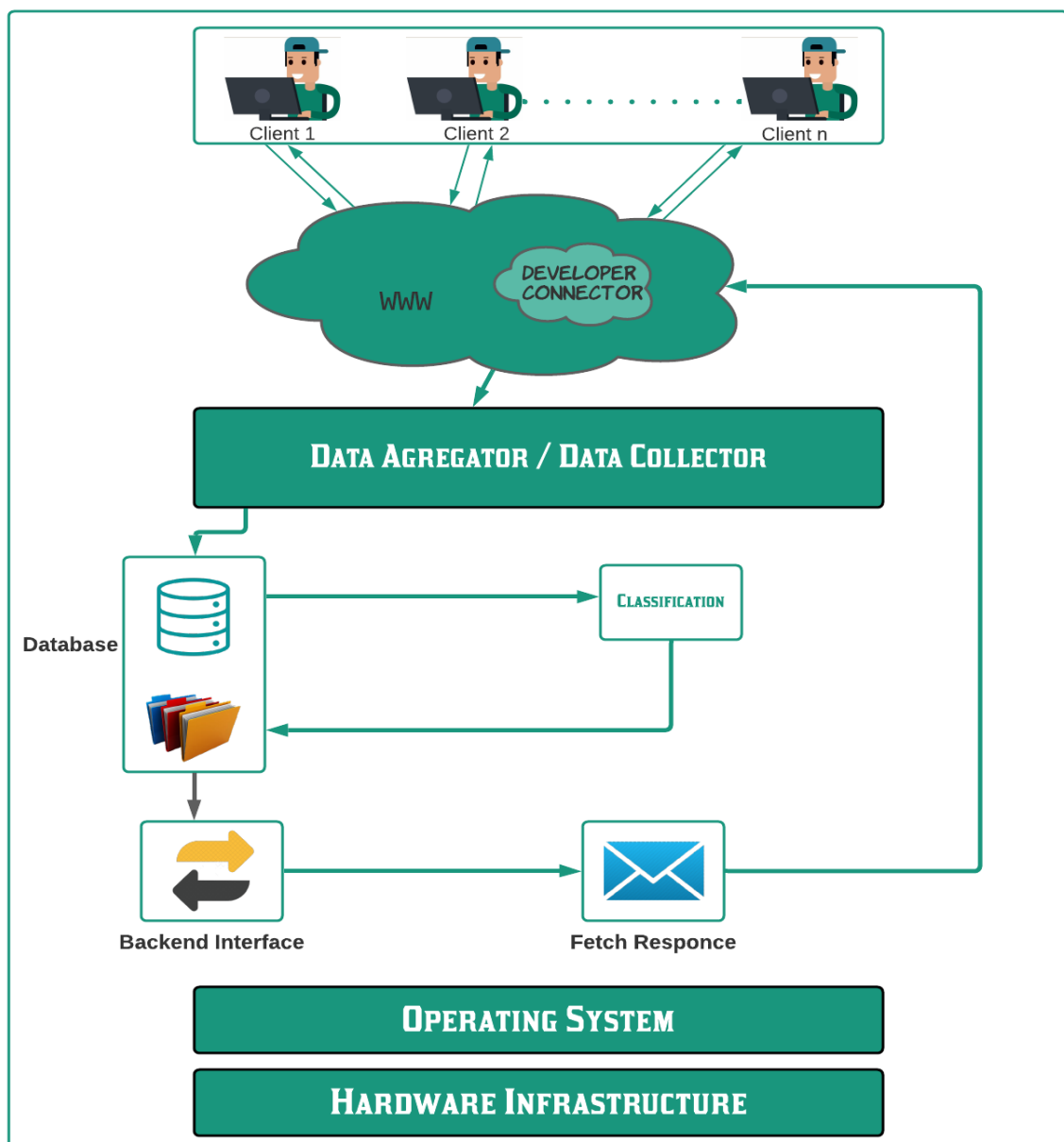
Chapter 4

System Design

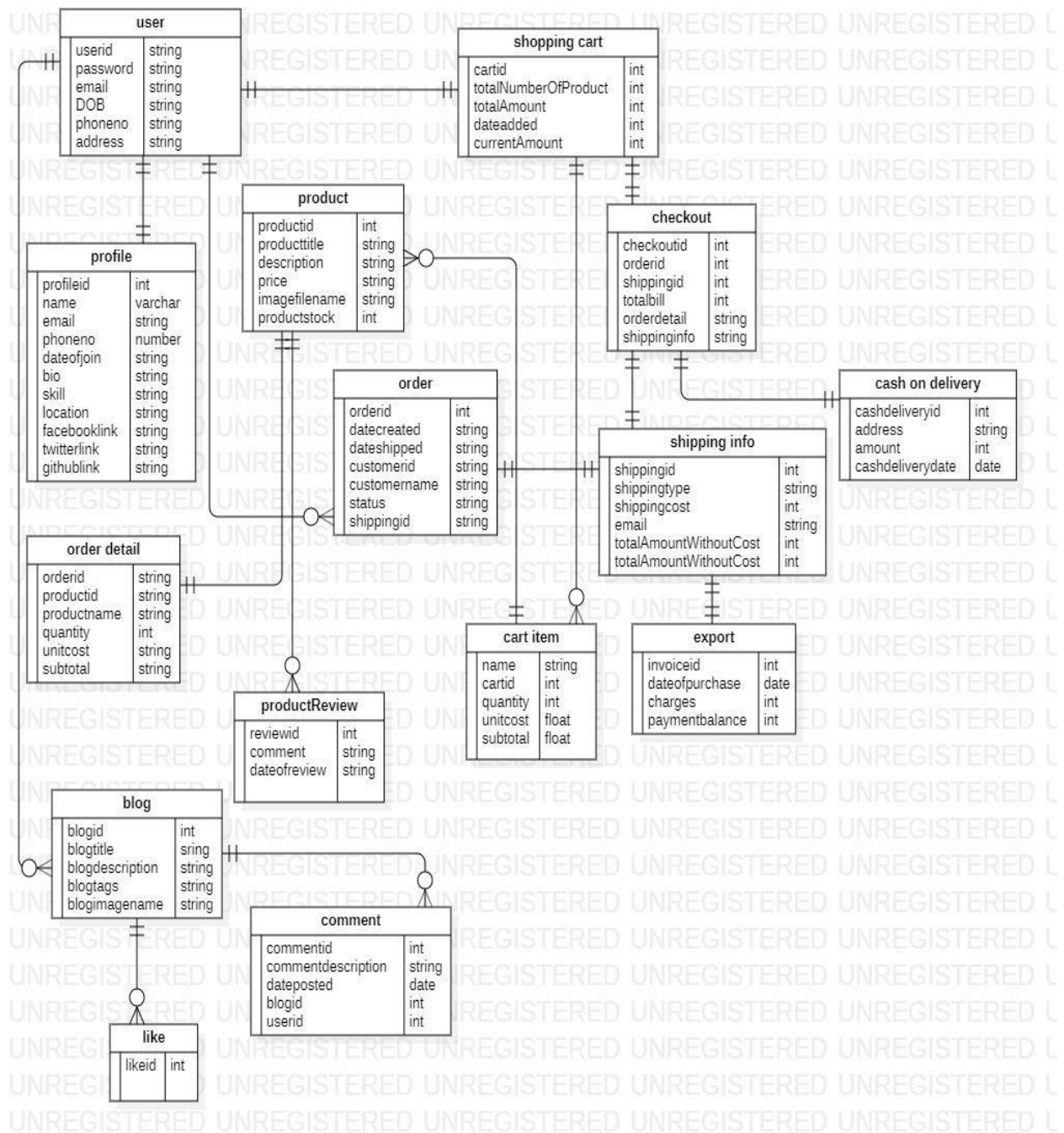
Chapter 4: System Design

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

4.1. Architecture Diagram



4.2. Entity Relationship Diagram with data dictionary







An entity-relationship diagram, or ER diagram, is important for demonstrating the data stored in a database. It is the basic design upon which a database is built. ER diagrams specify what data we will store: the entities and their attributes. They also show how entities communicate to other entities.

Another advantage of ERDs is that they represent the data in a graphical manner. This makes it easier for business folks to understand.

Why we need to make ERD?

Like anything else, the life of a database has its ups and downs. In the future, we might need to make updates or any sort of changes to the database. If we have properly documented the database with an ER diagram, it is much easier (and I am telling you this from experience) to make updates or changes into the database.

Symbol	Meaning
Relationships (Cardinality and Modality)	
	Zero or One
	One or More
	One and only One
	Zero or More



Entity
(with no attributes)



Entity
(with attributes field)

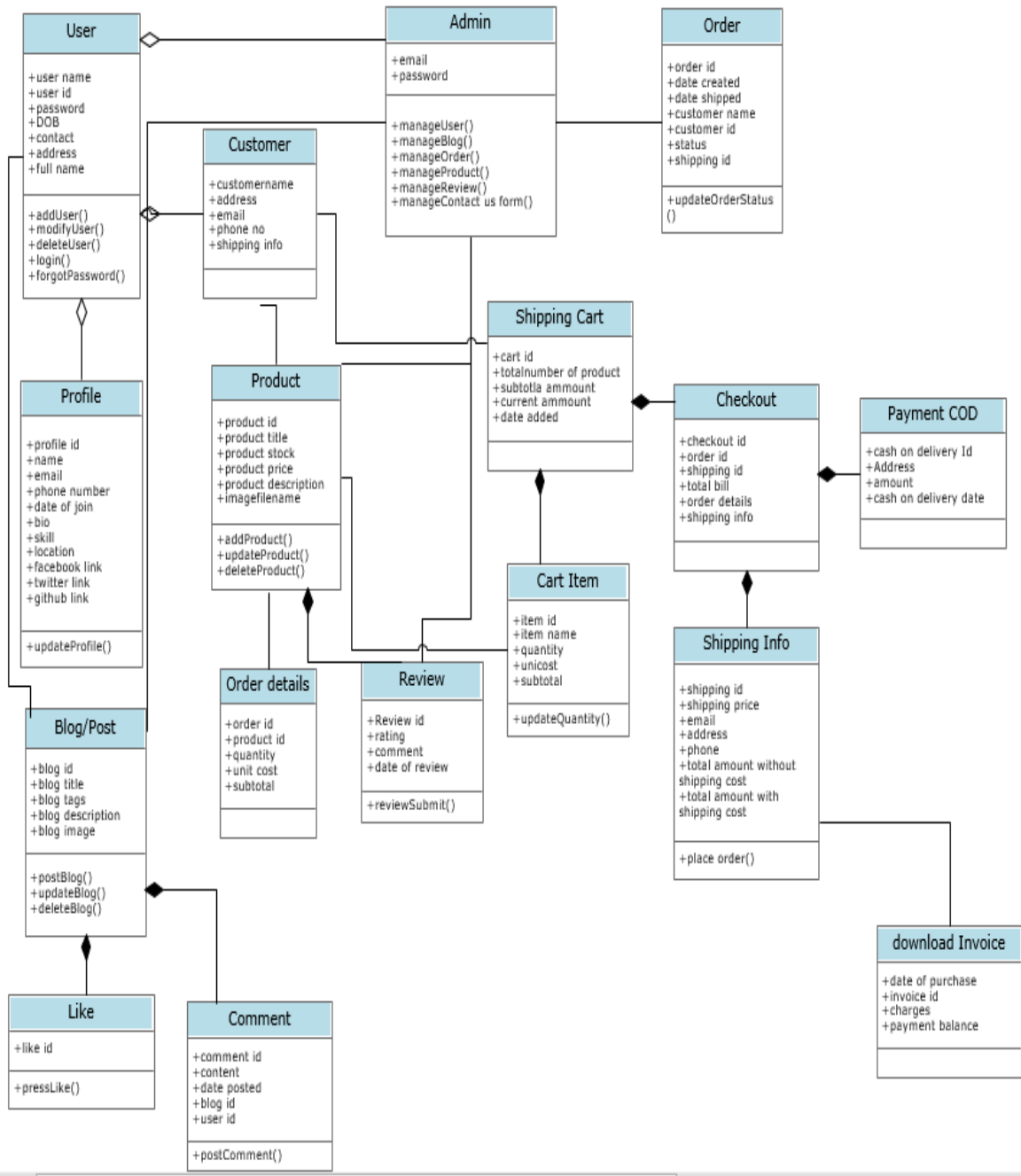


Entity
(attributes field with columns)



Entity
(attributes field with columns and
variable number of rows)

4.3. Class Diagram



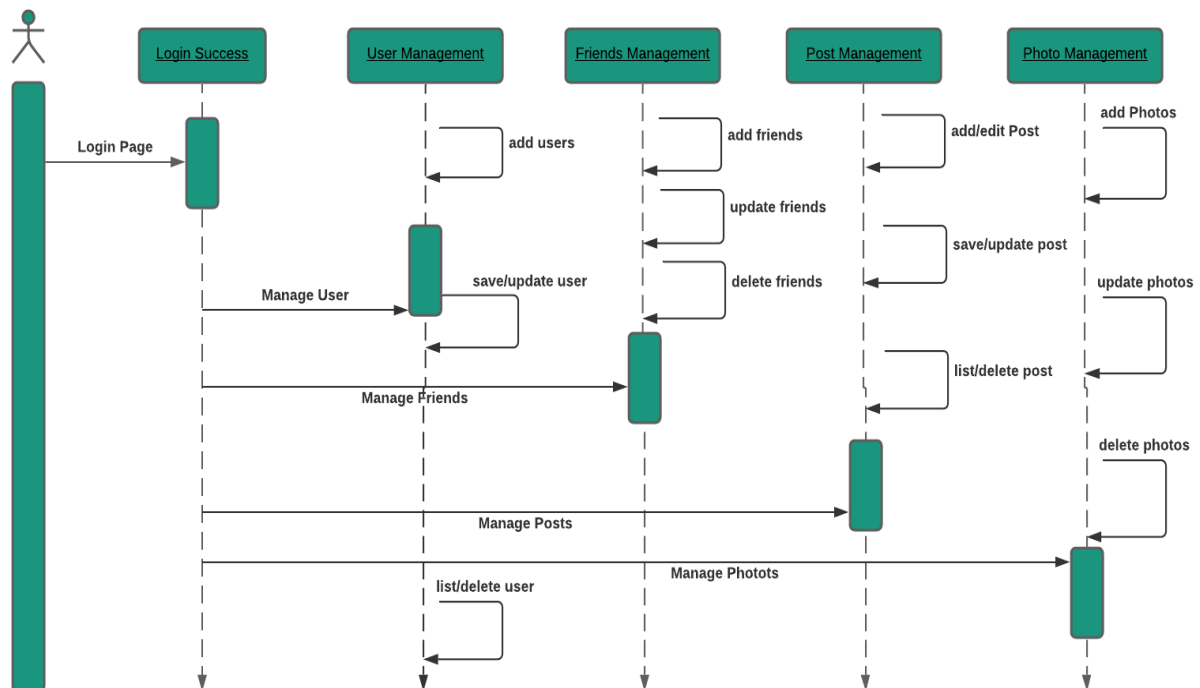
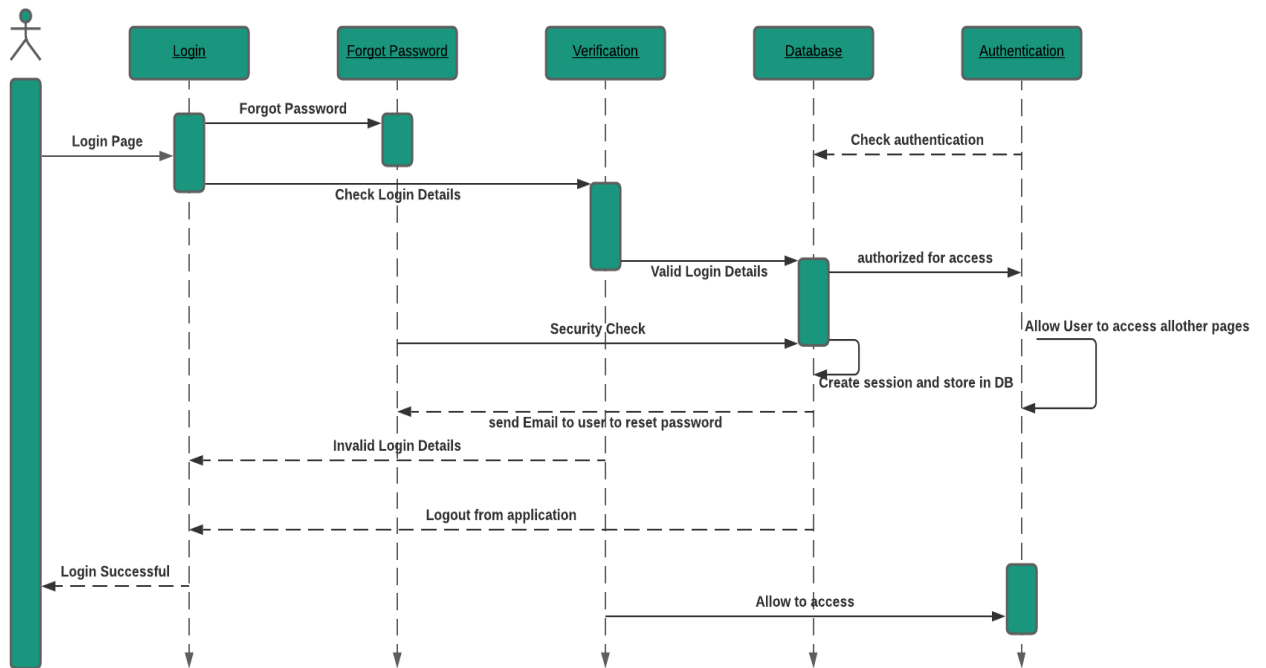
Class diagrams describe systems by demonstrating characteristics, operations and relationships between classes. Unified Modeling Language (UML) calls them structure diagrams.

What is a class diagrams used for?

Class diagrams give you the aptitude to create models with the help of UML using attributes, relationships, operations and intersections. A class diagram visualizes the paths between classes in the form of aggregations and associations as well as through the passing on of belongings and behavior between classes. These take the form of generalizations.

The class diagram represents the static structure of systems. It demonstrates classes with their attributes, operations as well as their statistical and logical inter-relationships.

4.4. Sequence / Collaboration Diagram



A sequence diagram is a type of collaboration diagram because it describes how and in what order a group of objects works together. Sequence diagrams are sometimes known as event diagrams or event scenarios or collaboration diagram as well.

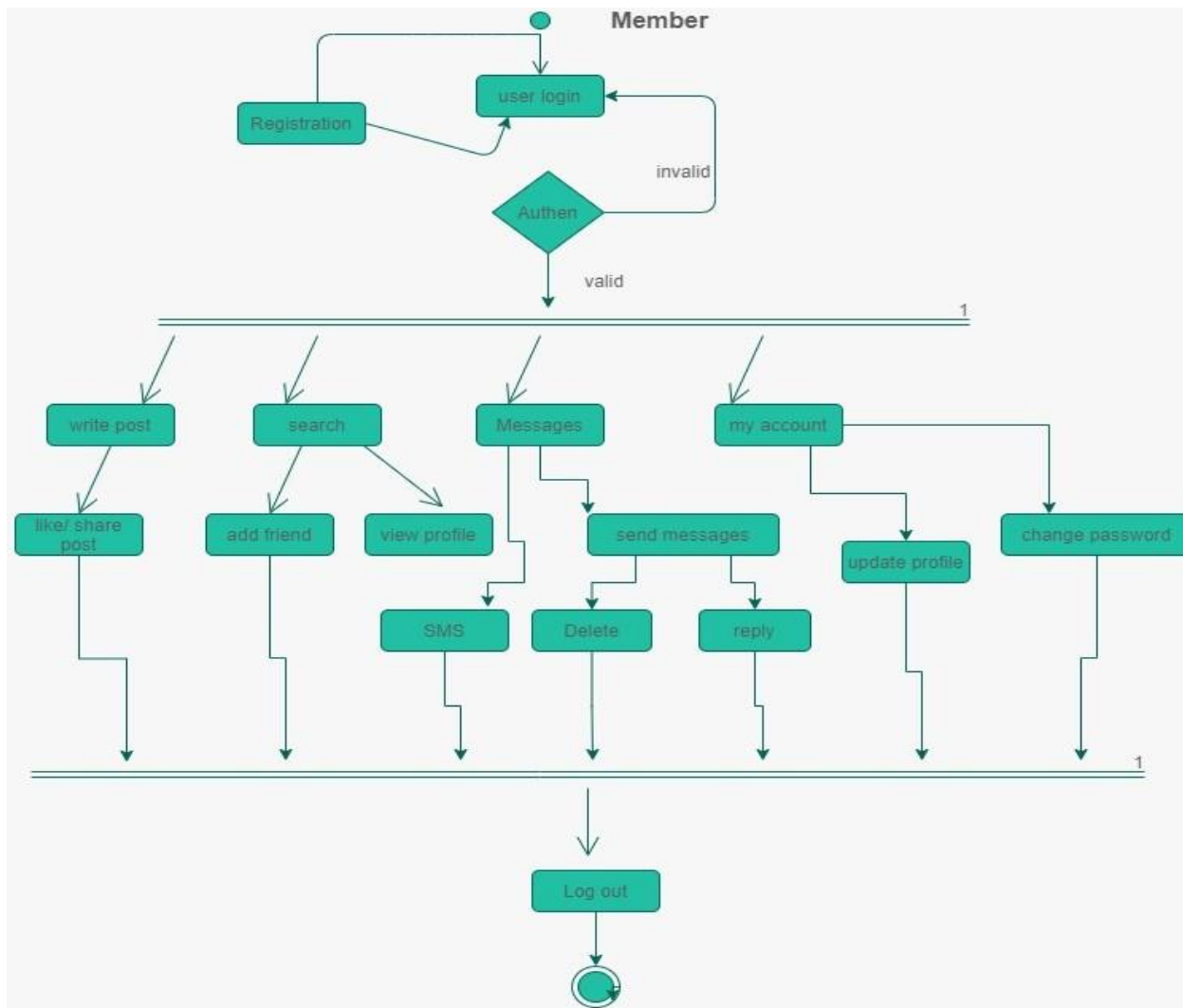
4.5. Operation contracts

Operation	Login/Signup
Description	A user login to the system or web application to access the functionality of the system.
Actors	User and Admin
Pre-Condition	System must be connected to the network.
Post-Condition	After got login, user get notification of successful login.

Operation	Post
Description	A user can post, update and delete any post he/she want to.
Actors	Users, admin
Pre-Condition	User's system must be connected to network while using application.
Post-Condition	User get notified after any operation is done by him/her.

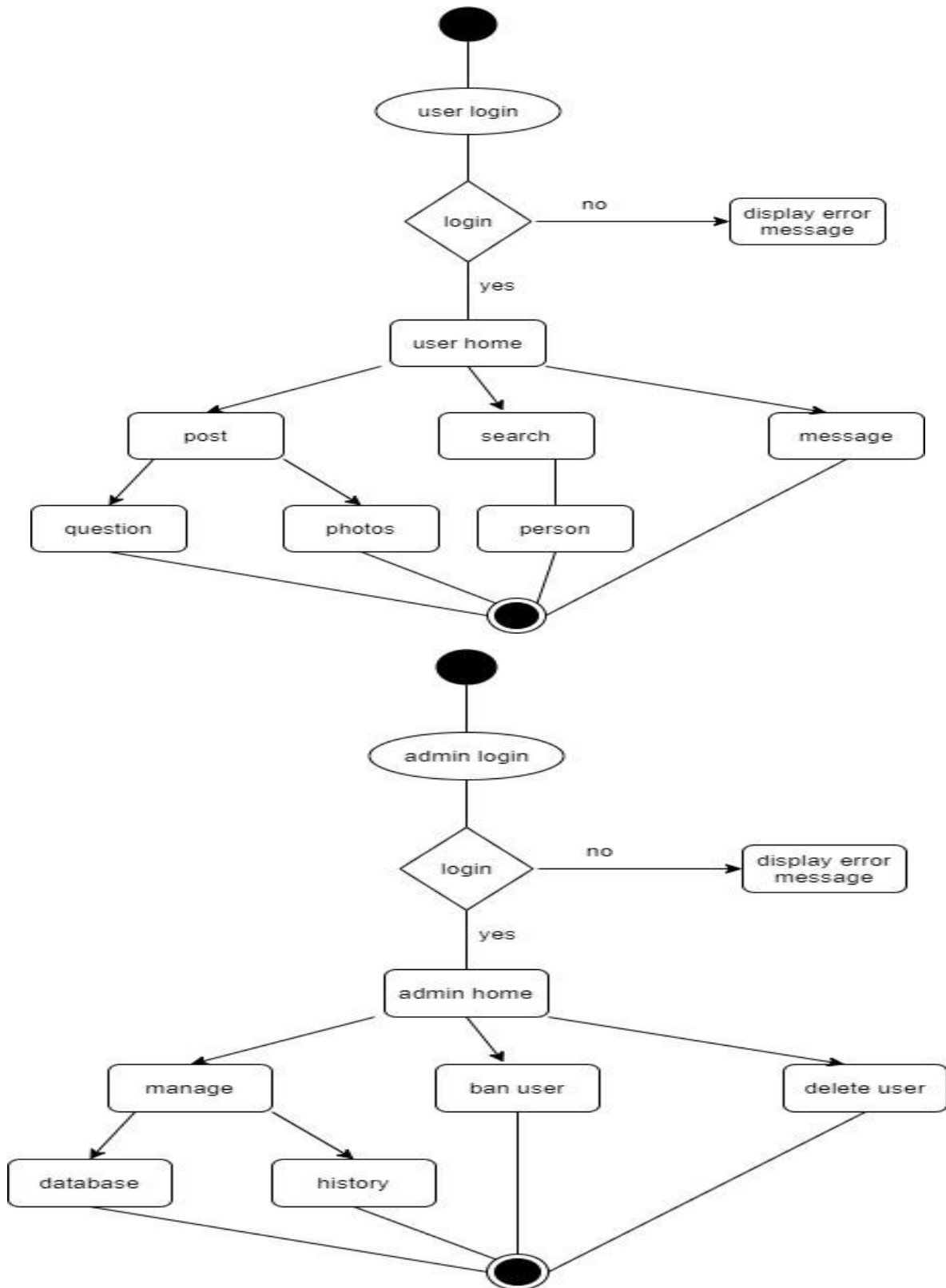
Operation	User
Description	A user can update and delete his/her account if they want to.
Actors	Users, admin
Pre-Condition	User's system must be connected to network while using application.
Post-Condition	User get notified after any operation is done by him/her.

4.6. Activity Diagram

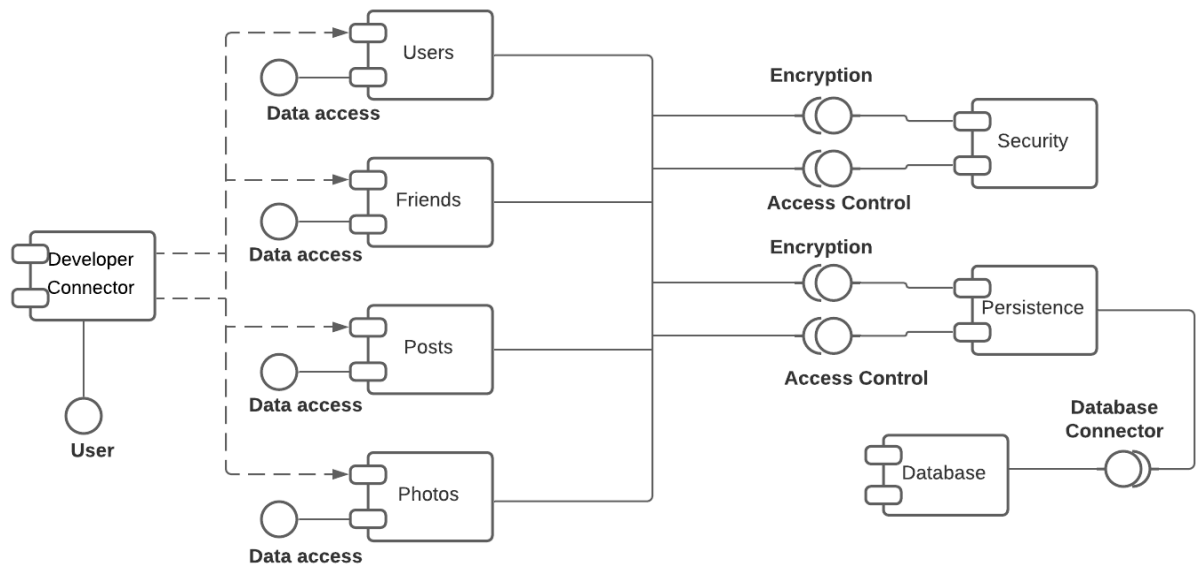


An activity diagram is used by developers to understand the flow of programs on a high level. It also allows them to figure out restrictions and conditions that cause particular events. A flow chart meets into being an activity diagram if complex decisions are being made.

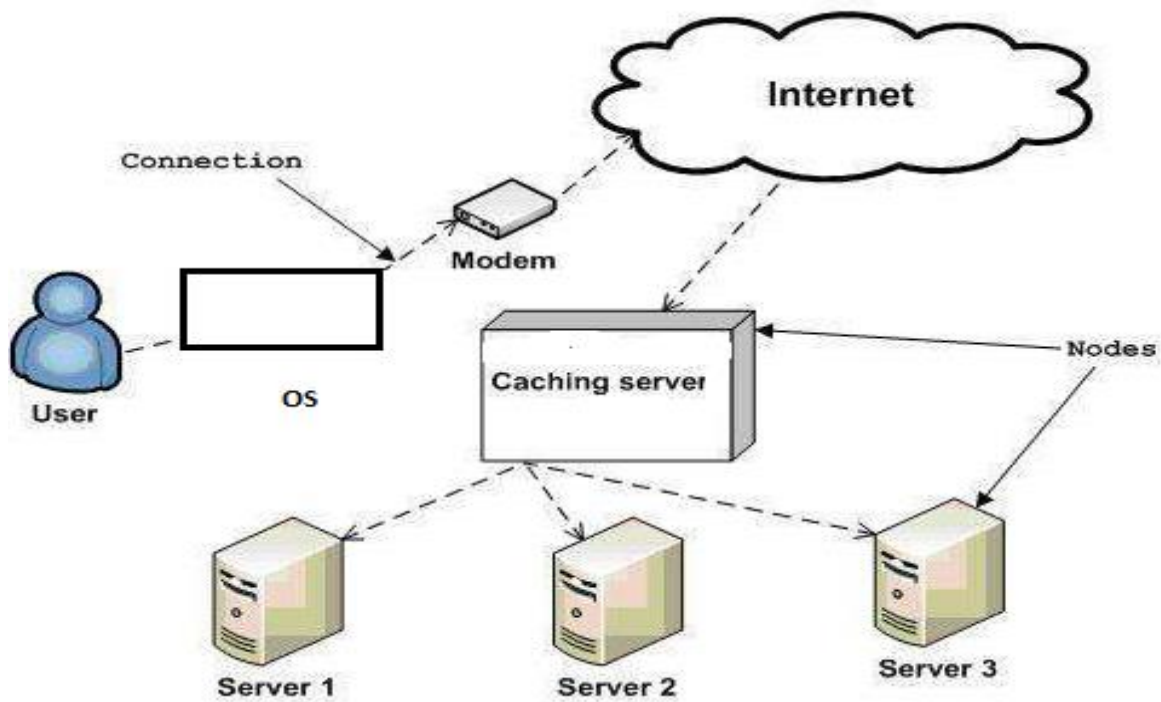
4.7. State Transition Diagram



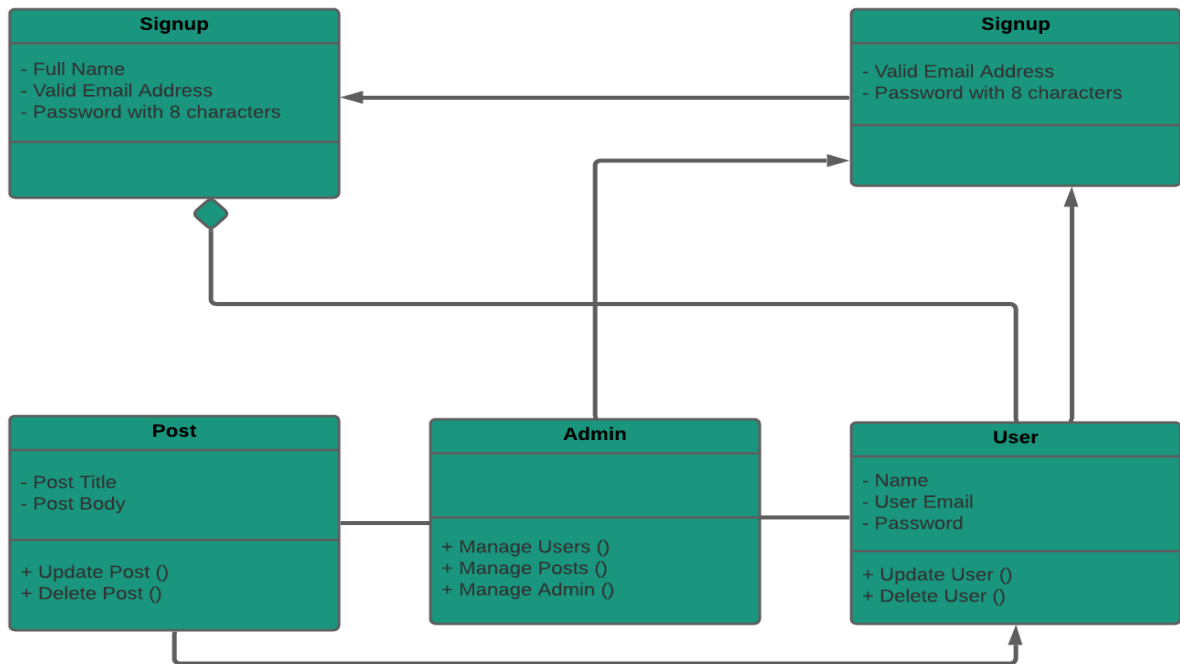
4.8. Component Diagram



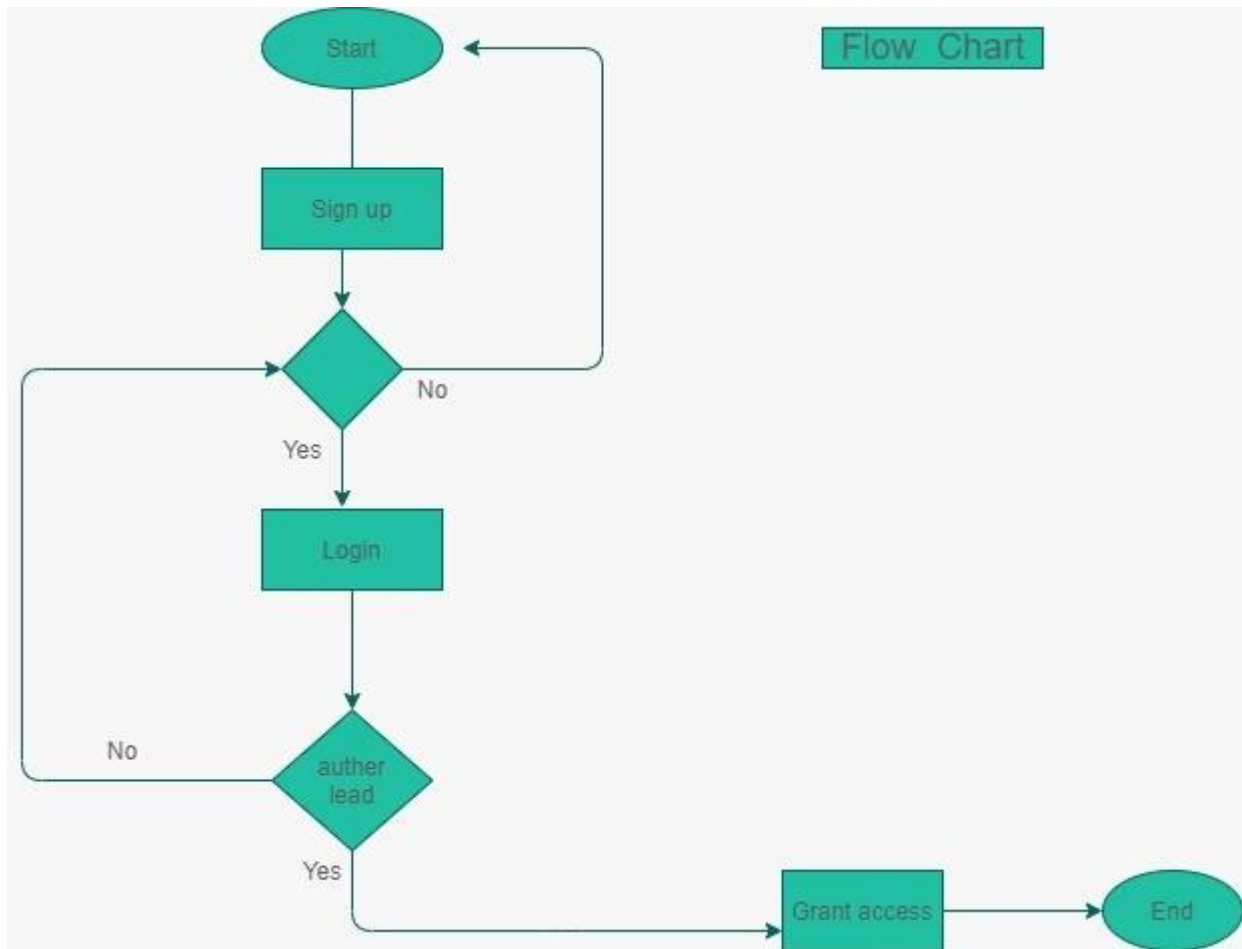
4.9. Deployment Diagram



4.10. Domain Diagram



4.11. Data Flow diagram



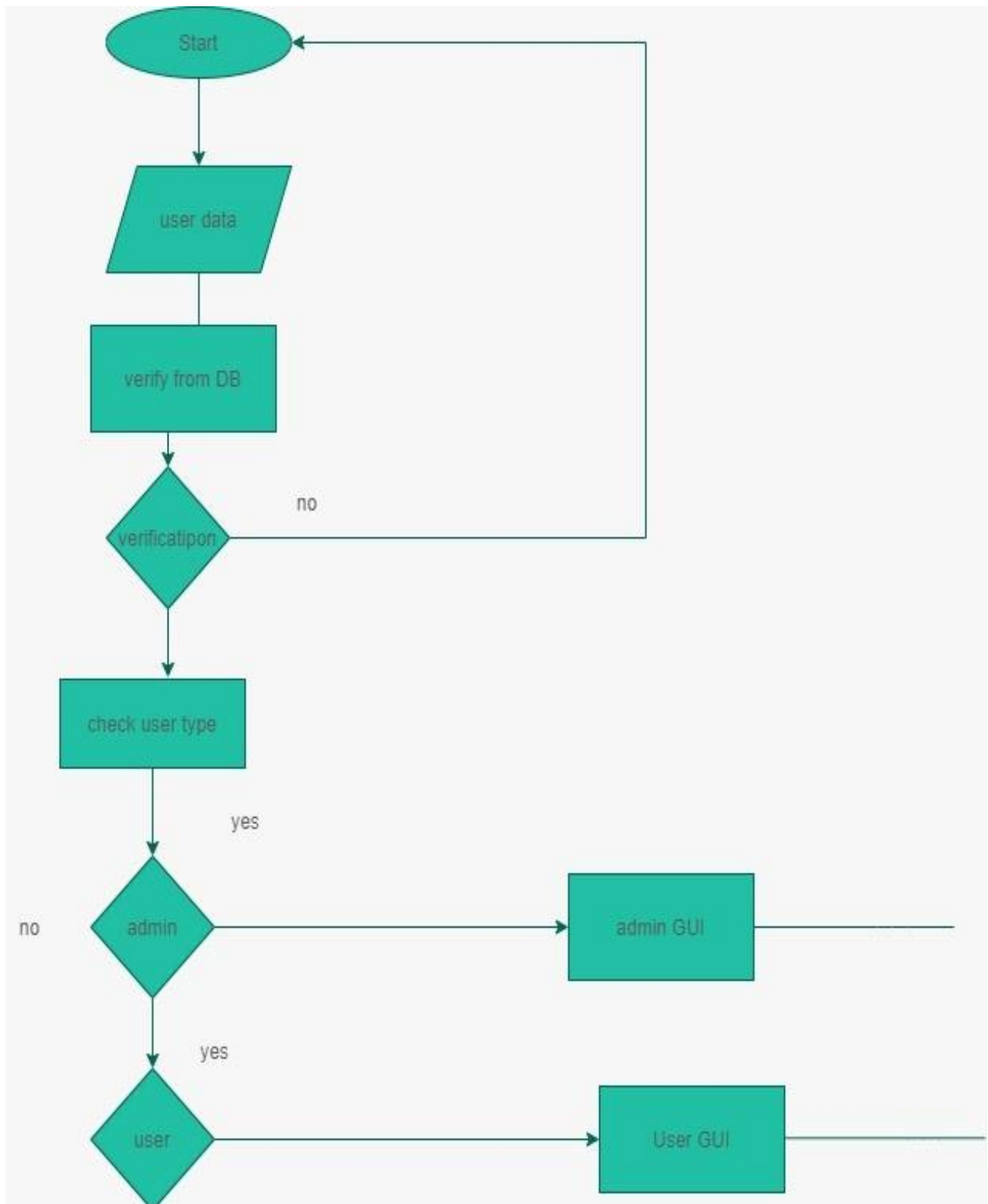
Chapter 5

Implementation

Chapter 5: Implementation

In this chapter we explained all about the application's components, libraries. Also about the deployment environment of the application. Tools and techniques used in the project. Code management, how to manage the code in term of easy access and factorization of the code.

5.1. Important Flow Control/Pseudo codes



5.2. Components, Libraries, Web Services and stubs

Libraries:

- JS
- React JS
- Express JS
- Node JS
- Mongo Db
- mongoose

5.3. Deployment Environment

- **Hardware with minimum Specs:** Core 2 duo, 80 GB Hard Drive, 1.9 GHZ, 4 GB RAM
Operating System: Window 7.
- **Application:** Google Chrome, Safari, Epic, Fire Fox
- **Network:** Modem, LAN WAN connections, Ethernet cables.

5.4. Tools and Techniques

Tools:

- Vs Code
- Mongo Db
- Postman

Technique:

- Agile Method

5.5. Best Practices / Coding Standards

Best code should be simple and divided into different chunks that will make ease to modify code. Factorization of code is good habit it makes your Application & website light and easily access able. That is why we divided our code into several chunks or I would say parts to easily make changes in it, can access it.

5.6. Version Control

We are using GIT version control system where we can handle our project with speed and efficiency. We can easily manage and share our code with other team members by using GIT.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

Test & Evaluation (T&E) is the process by which a system or components are compared against requirements and specifications through testing. The results are evaluated to assess progress of design, performance, supportability, etc.

6.1. Use Case Testing

User (Login)

Test Case ID	3.1	Test Case Description	Test the login functionality to enter system		
Created By	Usama Gulzar	Reviewed By	Majid Hussain	Test Module Name	User (Login)

Tester's Name	Qaium Saleem	Date Tested	17-dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	--------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Access to User Profile
2	
3	
4	

S #	Test Data
1	User id = mg12345
2	Pass = df12@434c
3	
4	

Test Scenario	Verify on entering valid user id and password, the customer can login
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to login page	Site should open	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click Submit	Customer is logged in	As Expected	Pass

6.2 User (Login)

Test Case ID	3.2	Test Case Description	Test the signup functionality		
Created By	Qaium Saleem	Reviewed By	Majid Hussain	Test Module Name	User (Sign up)

Tester's Name	Usama Gulzar	Date Tested	12-dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	--------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Enter the required data
2	
3	
4	

S #	Test Data
1	User id = mg12345
2	Pass = df12@434c
3	
4	

Test Scenario

Verify on entering valid user id and password, the customer can create account

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to main sign up page	Acc should be created	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click on create account	Customer is logged in	As Expected	Pass

6.3 User (update profile)

Test Case ID	3.2	Test Case Description	Test the function that can update profile		
Created By	Majid Hussain	Reviewed By	Usama Gulzar	Test Module Name	User (Sign up)

Tester's Name	Usama Gulzar	Date Tested	12-dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	--------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Must enter valid user id and password
2	
3	
4	

S #	Test Data
1	User id =
2	Pass = df12@434c
3	
4	

Test Scenario

Verify on entering valid user id and password, the customer can update profile

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to main sign up page	Site should be updated	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click on update account	Customer is logged in	As Expected	Pass

6.4 Admin (add user)

Test Case ID	3.4	Test Case Description	Test add user functionality either the user can be added or not		
Created By	Usama Gulzar	Reviewed By	Qaium Saleem	Test Module Name	User (Sign up)

Tester's Name	Majid Hussain	Date Tested	9-nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	------------	---	------

S #	Prerequisites:
1	User valid data is required
2	
3	
4	

S #	Test Data
1	User id = mg12345
2	Pass = df12@434c
3	
4	

Test Scenario

Verify the user can be added by the admin

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to add user page	Page should be opened	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click on add user	User added	As Expected	Pass

6.5 Admin (Delete User)

Test Case ID	3.5	Test Case Description	Test the user can be deleted or not		
Created By	Usama Gulzar	Reviewed By	Majid Hussain	Test Module Name	Admin (Delete User)

Tester's Name	Qaium Saleem	Date Tested	13-oct-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	--------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Enter valid id and password of the user
2	
3	
4	

S #	Test Data
1	User id = mg12345
2	Pass = df12@434c
3	
4	

Test Scenario

Verify the user can be deleted by admin

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to delete user page	Page should be opened	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click on delete user	Customer is logged in	As Expected	Pass

6.6 Admin (Update User)

Test Case ID	3.6	Test Case Description	Test either the user can be deleted or not		
Created By	Majid Hussain	Reviewed By	Usama Gulzar	Test Module Name	User (Sign up)

Tester's Name	Qaium Saleem	Date Tested	27-oct-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	--------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Must enter the valid information of user
2	
3	
4	

S #	Test Data
1	User id = mg12345
2	Pass = df12@434c
3	
4	

Test Scenario	Verify the user's information can be updated by the admin
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to main update user page	Page should open	As Expected	Pass
2	Enter User id & Password	Credential can be entered	As Expected	Pass
3	Click on update account	Customer updated	As Expected	Pass

6,7 User (Post)

Test Case ID	3.7	Test Case Description	Test the functionality that user can post something		
Created By	Majid Hussain	Reviewed By	Majid Hussain	Test Module Name	User (Post)

Tester's Name	Qaium Saleem	Date Tested	12-dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	--------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Must enter the required data
2	
3	
4	

S #	Test Data
1	Hy my name is Ahmed
2	selfie.png
3	
4	

Test Scenario Verify the user can post something or not

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to post something panel	Panel should open	As Expected	Pass
2	Enter the required data	Credential can be entered	As Expected	Pass
3	Click on post	Post uploaded	As Expected	Pass

6.8 User (Edit Post)

Test Case ID	3.8	Test Case Description	Test the functionality either the post can be edited or not		
Created By	Qaium Saleem	Reviewed By	Usama Gulzar	Test Module Name	User (Edit Post)

Tester's Name	Majid Hussain	Date Tested	15-nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Enter required data to edit
2	
3	
4	

S #	Test Data
1	Superior University
2	Gold Campus
3	
4	

Test Scenario Verify that edit post initialized by the user can be done or not

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to edit post panel	Panel should open	As Expected	Pass
2	Must enter the required data	Credential can be entered	As Expected	Pass
3	Click on create account	Post Edited	As Expected	Pass

6.9 User (Search Post)

Test Case ID	3.9	Test Case Description	Test the search button functionality		
Created By	Majid Hussain	Reviewed By	Qaium Saleem	Test Module Name	User (Search Post)

Tester's Name	Usama Gulzar	Date Tested	08-dec-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	--------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Must enter the required data
2	
3	
4	

S #	Test Data
1	Majid Hussain
2	Usama Gulzar
3	
4	

Test Scenario	Verify that on clicking search button other users can be searched
----------------------	---

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to search bar	Search bar should be open	As Expected	Pass
2	Enter valid username	Credential can be entered	As Expected	Pass
3	Click on search button	User displayed	As Expected	Pass

6.10 User (Update Password)

Test Case ID	3.10	Test Case Description	Test the functionality that password can be updated or not		
Created By	Qaium Saleem	Reviewed By	Usama Gulzar	Test Module Name	User (Update Password)

Tester's Name	Majid Hussain	Date Tested	28-nov-2021	Test Case (Pass/Fail/Not Executed)	Pass
----------------------	---------------	--------------------	-------------	---	------

S #	Prerequisites:
1	Enter the required data
2	
3	
4	

S #	Test Data
1	User id = mg12345
2	Pass = df12@434c
3	
4	

Test Scenario

Verify that by entering old password the user can update to new password or not

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to update password dialogue box	Site should open	As Expected	Pass
2	Enter old Password	Credential can be entered	As Expected	Pass
3	Click on update password button	Password updated	As Expected	Pass

6.2. Equivalence partitioning

1	User name is alphabetic.	valid
2	User name is not alphabetic.	invalid
3	Password is equal to 10 or greater than 15 characters in length.	valid
4	Password is 2 to 16 characters in length.	invalid
5	Password include one uppercase letter & one lowercase letter.	valid
6	Password include ' _ '.	invalid
7	Email without '@' and '.com'.	invalid
8	Email with '@' and '.com'.	valid
9	Price only be numeric.	valid
10	Price is alphabetic.	invalid
11	Search product with alphabetic.	valid
12	Search product with special character (/,%^,#,@).	invalid
13	No search product name entered	invalid
14	Brand name alphabetic	valid
15	Brand name with special character	invalid
16	Empty field	invalid

6.3. Boundary value analysis

		Partition 1	Partition 2	Partition 3
1.	Password	Less than 8 character	1 – 7 character	< 12
2.	Phone	<=0	1 - 11	< 11

6.4. Data flow testing

In data flow testing we tested the flow of data in program. Moreover, we tested all the relationships between different entities and variables. There is no conflict against any initialized variable in code.

6.5. Unit testing

We performed the execution of our system on different computers and we testified that the execution was successful and performed well.

6.6. Integration testing

Significant progress has been made to date in realizing our final project goal. As a group we have accomplished the following:

Designed overall user interface structure for Dev Connector

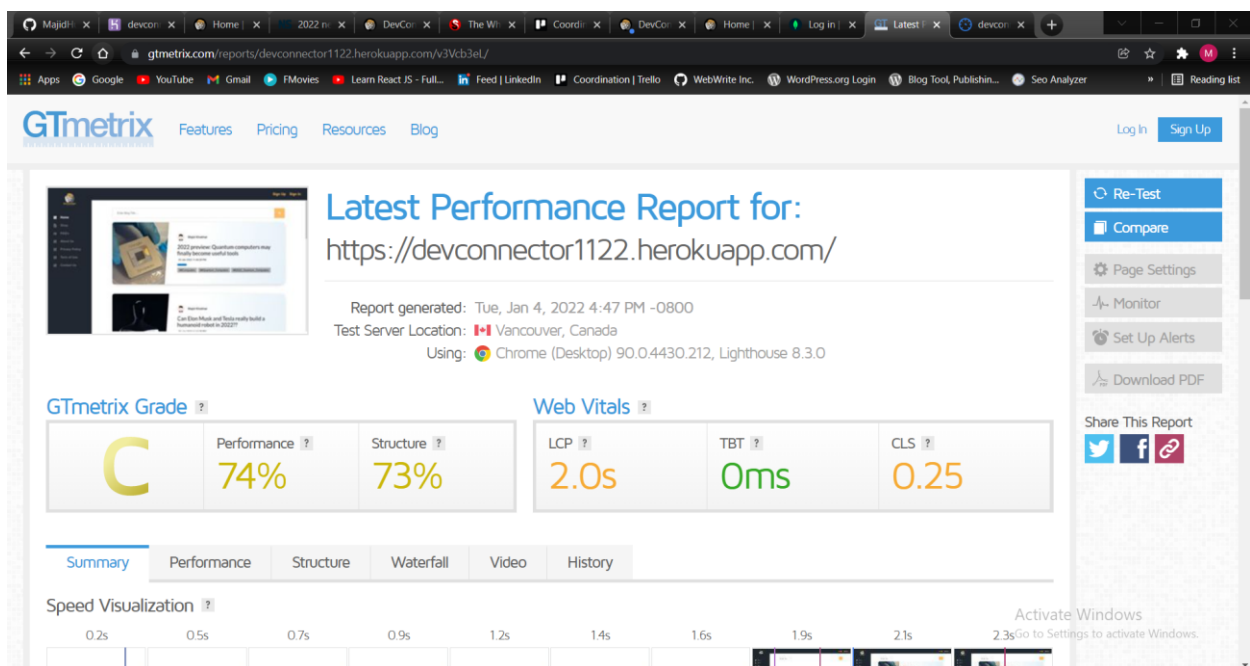
Made significant progress in integrating

We three were responsible to complete a number of the tasks, all of which are now completed.

We also conducted various tests like performance testing, Use Case Testing, Unit Testing etc. to The goal of our project remains the same that is, to design a user-friendly interface that makes use of our authentication and verification algorithms for a secure IT based environmental web application. We have also used the native integrations by JSON-like which is quite faster than other DB's. Moreover we integrated git hub as well users can link their github accounts to their profiles as well.

6.7. Performance testing

Performance testing from GTmetrix and the results are given below :



6.8. Stress Testing

We did a sample testing of traffic consist of 100 users and the performance of website was very good. It didn't stuck at any place. If we talk about its responsiveness, the app is very responsive. We did stress testing on BlazeMeter to check the overall performance of the app. There was no error occurred during the whole testing under all situations.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

We were very focused and dedicated towards our goals from very first day that's why we managed to complete our project in time. Our project is web based application especially made for IT students, Teachers, Researcher's because as a student we also know that we have to go on multiple websites to find out the answer against our each query. This project will help them all to find out the answers regarding their field of interest at one place. Each user has its own profile from where he can post blogs, ask questions, and get projects. There is an IT mart as well where he / she can buy IT related products as well. We tried our best to create an IT world.

7.2. Achievements and Improvements

We have achieved a lot throughout the project we completed this project in milestones and we always managed to completed the milestones in given time. Moreover, we have created something very unique according to the market standards. The biggest achievement that we were able to achieve was to boost up our professional skills. Still there are many new things entering in IT world day by day we will try our best to improve our website and keep it up to date according to market standards.

7.3. Critical Review

IT world is growing very fastly and if we talk about our website it's not competitive to other websites already running in market. There is always a scope of improvement and we are very focused to adapt new technologies that will be helpful for this project as well. The app is not that much secure right now but we will make it secure as soon as possible by passing through all the testing phases again and again

7.4. Lessons Learnt

We have learned many things while doing the project, the main thing we learned that there is always something we don't know about. We should have a learning behavior every time, if we will not adapt new things and technologies than we will be considered as outdated people because IT is growing very fastly so every day is a new day in IT world .

7.5. Future Enhancements/Recommendations

We managed to imbed all the necessary functionalities at one platform but still we will keep introducing the new features in our system. Moreover, we will make Dev Connector a cross platform app in future. We will also try to make the user interface more attractive and will Collaborate with renowned IT brands to provide discounts to our users so they can easily buy the products from our IT mart.

Appendices

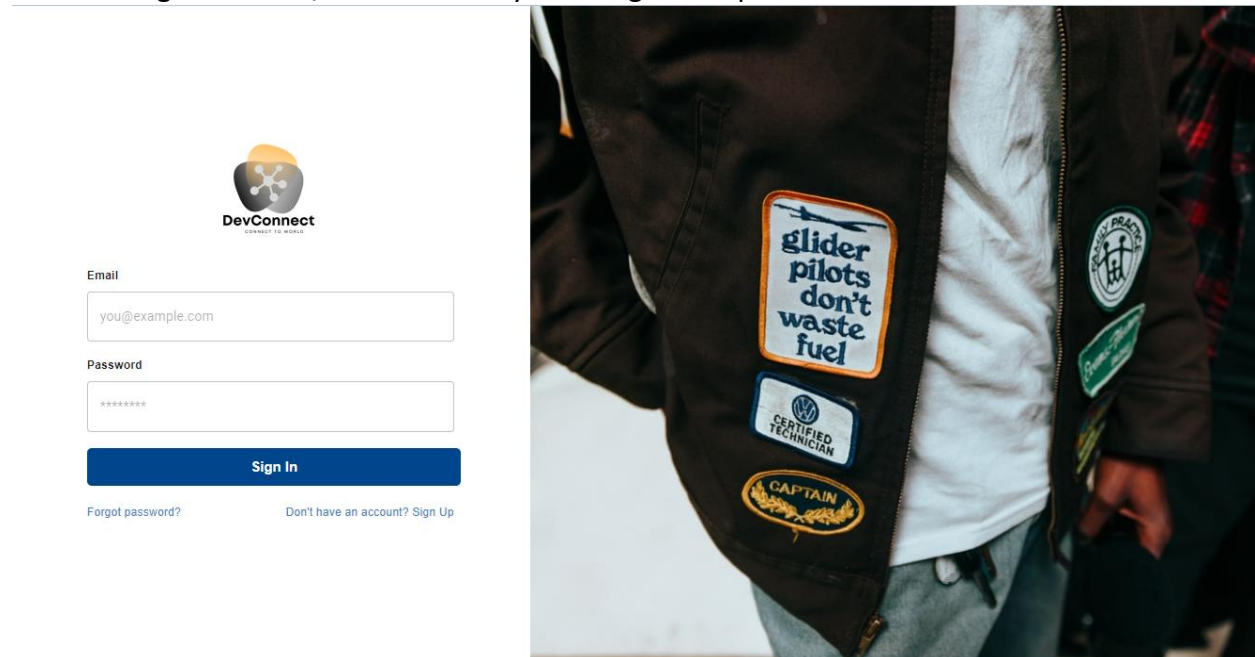
Appendix A: User Manual

This appendix will describe the user related functionalities and user interface as well

Appendix A: Dev Connector

A.1.Login

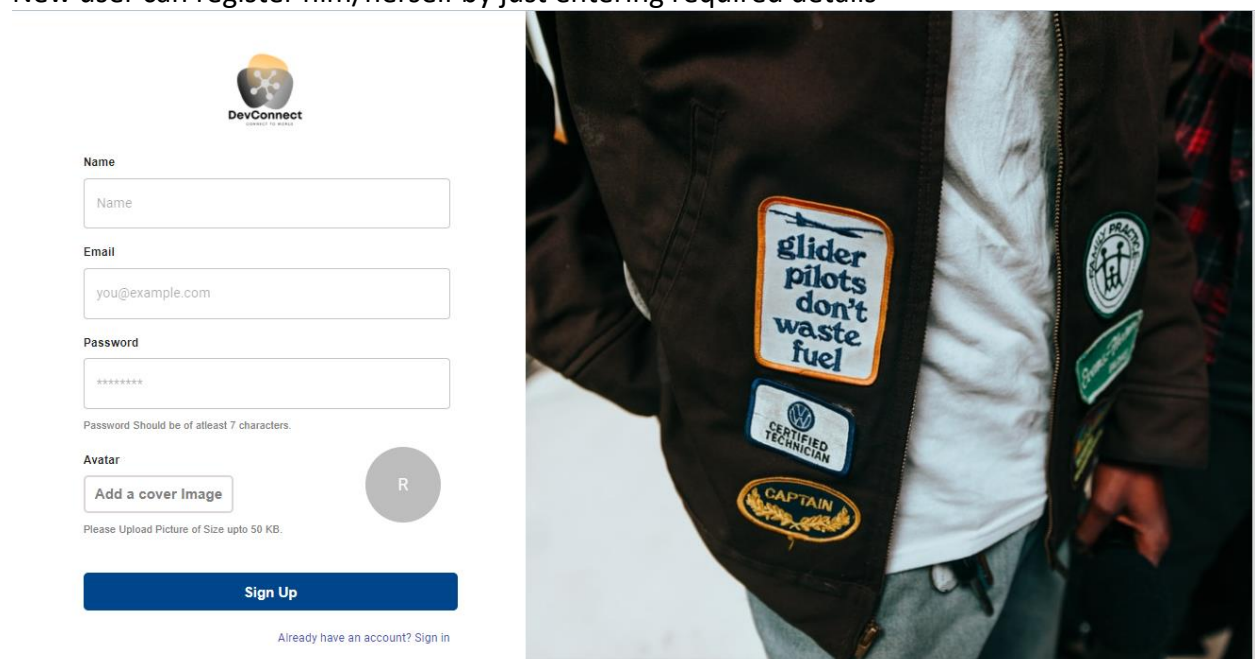
User can login in to his / her account by entering id and password



The image shows two side-by-side elements. On the left is a screenshot of the DevConnect login interface. At the top is the DevConnect logo with the tagline 'CONNECT TO WORLD'. Below it are two input fields: 'Email' containing 'you@example.com' and 'Password' with masked characters. A blue 'Sign In' button is positioned below the fields. At the bottom of the form are two links: 'Forgot password?' and 'Don't have an account? Sign Up'. On the right is a photograph of a dark brown flight jacket. The jacket features several patches: a large orange-bordered patch that reads 'glider pilots don't waste fuel', a blue 'CERTIFIED TECHNICIAN' patch, and a gold 'CAPTAIN' patch. A circular patch on the right chest reads 'REAL PRAGMATIC'.

A.2.Sign up

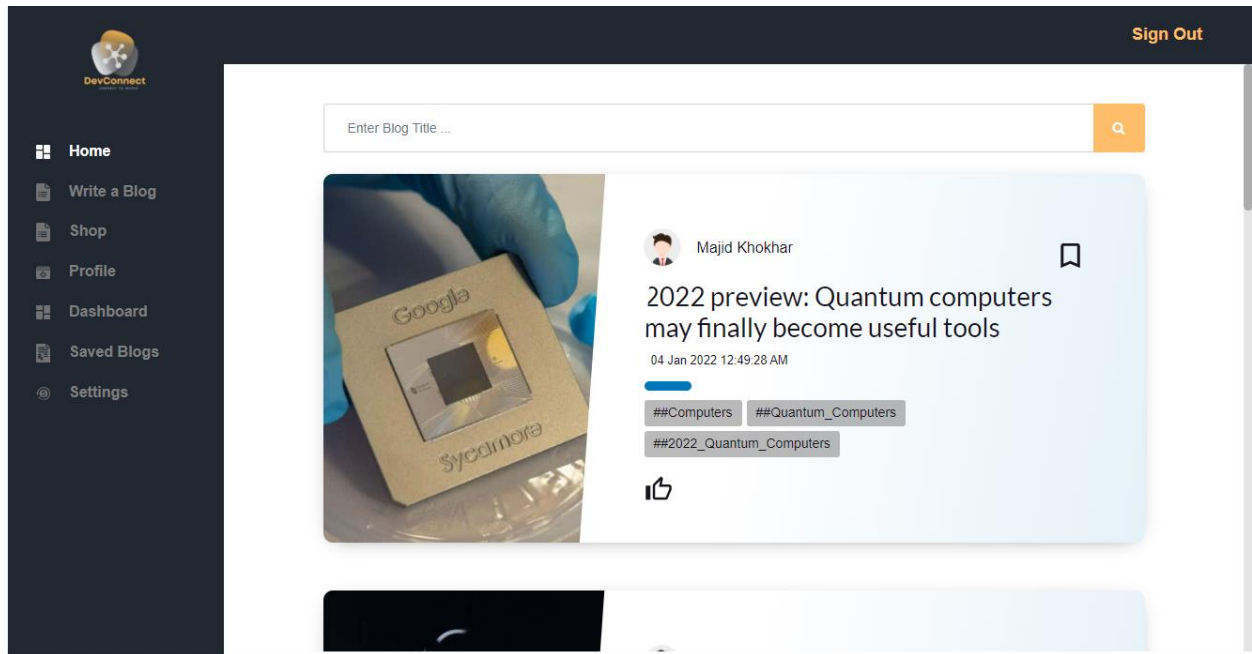
New user can register him/herself by just entering required details



The image shows two side-by-side elements. On the left is a screenshot of the DevConnect sign-up interface. At the top is the DevConnect logo with the tagline 'CONNECT TO WORLD'. Below it are four input fields: 'Name', 'Email' containing 'you@example.com', 'Password' with masked characters, and a note 'Password Should be of atleast 7 characters.'. Below the password field is an 'Avatar' section with a button 'Add a cover Image' and a circular placeholder containing the letter 'R'. A note below the avatar section says 'Please Upload Picture of Size upto 50 KB.'. A blue 'Sign Up' button is at the bottom of the form. At the bottom right of the form is a link: 'Already have an account? Sign in'. On the right is a photograph of a dark brown flight jacket, identical to the one in the previous image, featuring patches for 'glider pilots don't waste fuel', 'CERTIFIED TECHNICIAN', 'CAPTAIN', and 'REAL PRAGMATIC'.

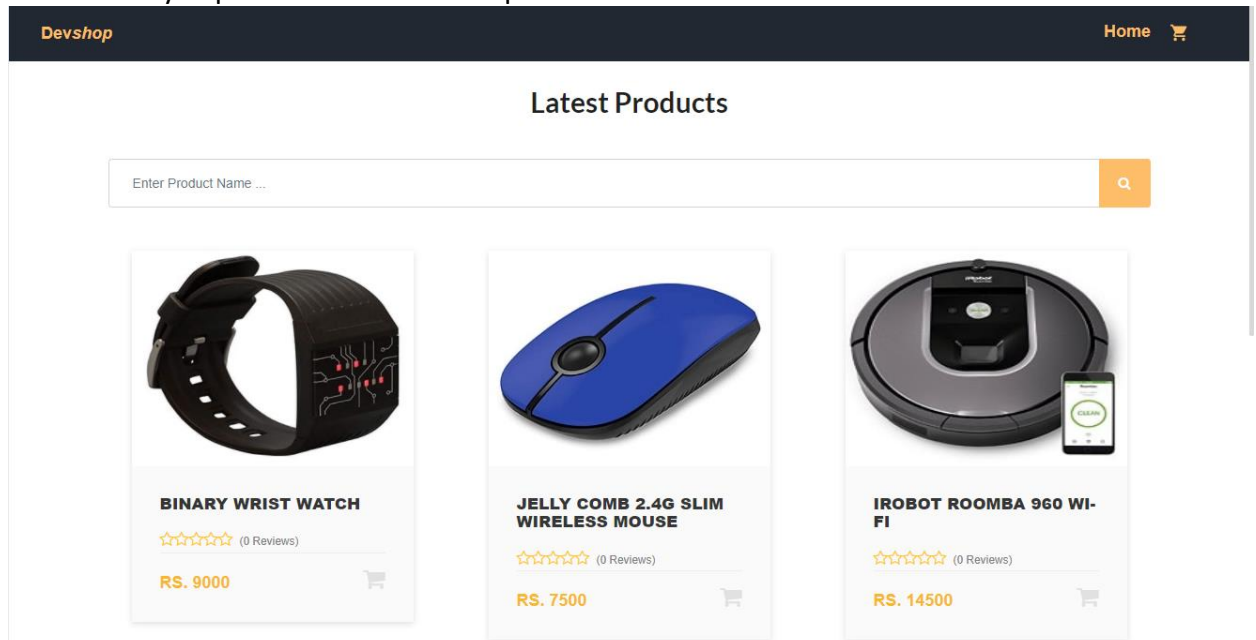
A.3.Home Screen

User can enter the home screen after login



A.4.DEVSHOP

User can buy IT products from Devshop



A.5.Client Dashboard

Dashboard

Sign Out

Total Blogs 3

Total Order 2

Total Likes 5

Total Saved Blogs 2

BLOGS

Name	Created At	Likes	
this is the title of my Post!!!!	10 Oct 2021	1	
aaaa	25 Nov 2021	2	
aaa	25 Nov 2021	2	

A.4.Order Details

Sign Out

Order # 6169613984256e40d44877f4

[Download Invoice](#)

Shipping Info

Name: Majid Hussain
 Email: majidkhokhar0099@gmail.com
 Phone: 03034070680
 Address: sultan town lahore, lahore, undefined, Pakistan
 Shipping Price: RS. 250
 Total Amount (without shipping price) : RS. 500
 Total Amount (with shipping price) : RS. 750

Payment

NOT PAID

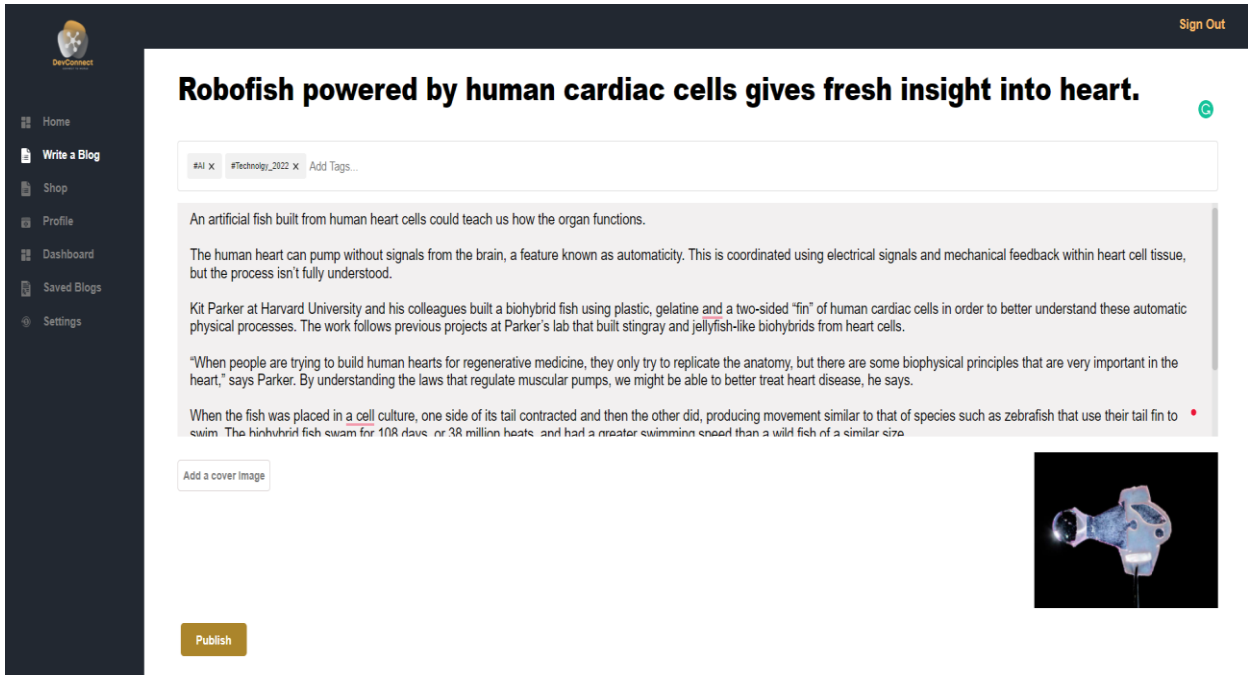
Order Status:

Processing

Order Items:

abc	abc	RS. 500	1 Piece(s)
-----	-----	---------	------------

A.4. Blog/Post



The screenshot shows a dark-themed web interface for a blog post editor. On the left is a sidebar with navigation links: Home, Write a Blog (active), Shop, Profile, Dashboard, Saved Blogs, and Settings. The top right corner has a 'Sign Out' link. The main content area features a title 'Robofish powered by human cardiac cells gives fresh insight into heart.' with a green 'G' icon. Below the title is a text input field containing '#AI X #Technology_2022 X Add Tags...'. The main text area contains the following content:

An artificial fish built from human heart cells could teach us how the organ functions.

The human heart can pump without signals from the brain, a feature known as automaticity. This is coordinated using electrical signals and mechanical feedback within heart cell tissue, but the process isn't fully understood.

Kit Parker at Harvard University and his colleagues built a biohybrid fish using plastic, gelatine and a two-sided "fin" of human cardiac cells in order to better understand these automatic physical processes. The work follows previous projects at Parker's lab that built stingray and jellyfish-like biohybrids from heart cells.

"When people are trying to build human hearts for regenerative medicine, they only try to replicate the anatomy, but there are some biophysical principles that are very important in the heart," says Parker. By understanding the laws that regulate muscular pumps, we might be able to better treat heart disease, he says.

When the fish was placed in a cell culture, one side of its tail contracted and then the other did, producing movement similar to that of species such as zebrafish that use their tail fin to swim. The biohybrid fish swam for 108 days, or 38 million beats, and had a greater swimming speed than a wild fish of a similar size.

Below the text is a button labeled 'Add a cover image' and a 'Publish' button. On the right side, there is a small image of a blue, translucent biohybrid fish with a white fin.

Standeer:





PROBLEM & SOLUTION

It is a very difficult task to find out the platform where you can in touch with the people according to your field of interest, Like Soft. house developers, freelancers and others could not find the proper platform to communicate with their related people.

Developer connector is a platform that will allow different communities to get connected as above mentioned like Soft. houses, Universities, Freelancers, Researchers and Students easily on onespecific platform.

TOOLS/TECHNOQUES/LIBRARIES

-  VS Code
-  Postman
-
- Agile Method
-
-  JS
-  React JS
-  Express JS
-  Node JS
-  Mongo Db
-  Bootstrap
-  JWT












MAJID HUSSAIN
BCSM-S18-014



USAMA GULZAR
BCSM-S18-007



QAIUM SALEEM
BCSM-S18-039

Supervisor
Hafiz Naeem