SEPM LAB REPORT

18CSC206J - Software Engineering and Project Management

Submitted by

KOLLI SESHA REDDY [RA2011003010225]

Under the Guidance of

Dr. C. Santhanakrishnan

Assistant Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

JULY 2022

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this lab report titled **“Last One- Password Manager And Generator”** is the bonafide work done by **kolli** sesha reddy **(RA2011003010225)** who carried out the lab exercises under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE
Dr. C. Santhanakrishnan
SEPM – Course Faculty
Assistant Professor
Department of Computing Technologies

ABSTRACT

The project “Password Manager and Generator” is dedicated to the general users of multiple passwords. The Main objective of the project is to create a software that allows the users to view, manage and modify different passwords of different applications in one place. It has a feature that allows the users to create the passwords of their liking. In the password generation process the user can select the specifications like upper case letters, lower case letters ,word length and special characters

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MVP	-	Minimum Viable Product
DFD	-	Data Flow Diagram
WBS	-	work breakdown structure
NFR	-	Non-Functional Requirement
SWOT	-	Strengths, Weaknesses, Opportunities, and Threats
HTTPS	-	Hypertext Transfer Protocol Secure
ER Diagram	-	Entity Relationship Diagram
UML	-	Unified Modelling Language

CHAPTER 1

PROBLEM STATEMENT

1.1 PROPOSAL

The proposed project “Password manager and generator” is a software where users can view, modify ,add and remove passwords all in one place. The software allows the user to create passwords according to their specifications.

The History

- Users often use common dictionary words combined with predictable series of numbers which makes it easier to guess/crack the password.
- Users with many accounts often forget their password

Limitations

- High encryption of the password is not available
- Autofill to the desired website is not available

Approach

- Database to store and manage all passwords and username

Benefits

- Once you save a password , you'll always have it when you need it, logging in is fast and easy.
- The built-in password generator creates long, randomized passwords
- Get all your passwords and username in one single place.

CHAPTER-2

STAKEHOLDERS & PROCESS MODELS

2.1 SELECTION OF METHODOLOGY

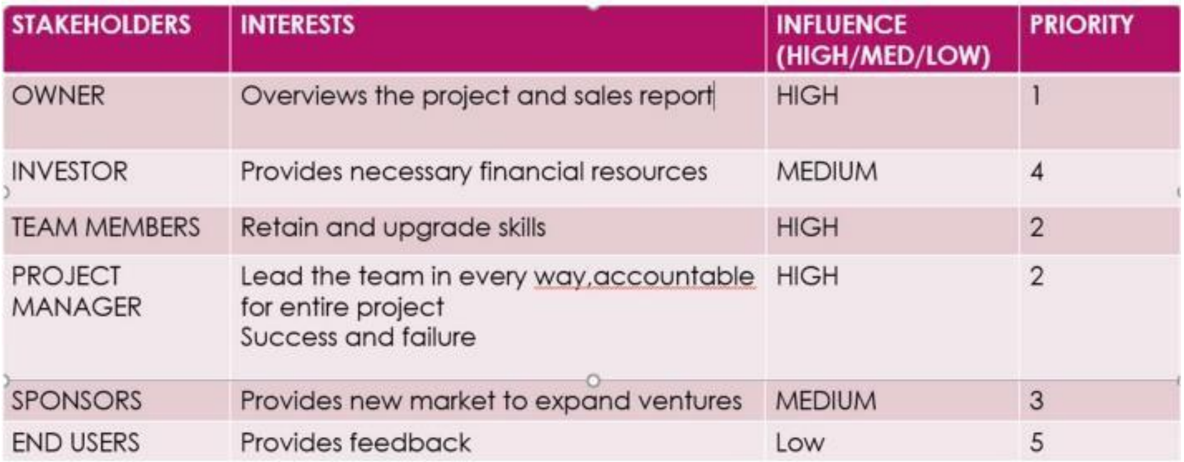
Waterfall methodology is a linear methodology in which client involvement is minimum and adding changes in the project is very difficult.

Agile methodology is much more flexible, client involvement is more and change can be made easily.

We will be using **Agile methodology** as each stage doesn’t depend on the previous stage thus it is flexible and changes can be easily introduced.

2.2. INFORMATION REGARDING STAKEHOLDERS OF THE PROJECT

Table 2.1 - Stakeholder Identification



STAKEHOLDERS	INTERESTS	INFLUENCE (HIGH/MED/LOW)	PRIORITY
OWNER	Overviews the project and sales report	HIGH	1
INVESTOR	Provides necessary financial resources	MEDIUM	4
TEAM MEMBERS	Retain and upgrade skills	HIGH	2
PROJECT MANAGER	Lead the team in every way,accountable for entire project Success and failure	HIGH	2
SPONSORS	Provides new market to expand ventures	MEDIUM	3
END USERS	Provides feedback	Low	5

.

CHAPTER 3
IDENTIFYING REQUIREMENTS

3.1 System Requirements

Hardware:

1)Windows ,Android operating system and dual core processor.

2)64-bit dual core processor

3)2GB RAM,100MB storage

Developer tools:

1) Firebug lite(CSS,HTML,JAVA SCRIPT)

2) Microsoft visual studio(JAVA SCRIPT)

3.2 Functional Requirements

1)The system shall display all the Saved passwords

2)The system shall organize the list of passwords

3) The system shall provide a generator to create password

4) The system shall accept specifications from the user for the password

3.3 Non-Functional Requirements

1)The system shall be available 24hrs per day, 360 days per year.

2)The system shall not loose any data.

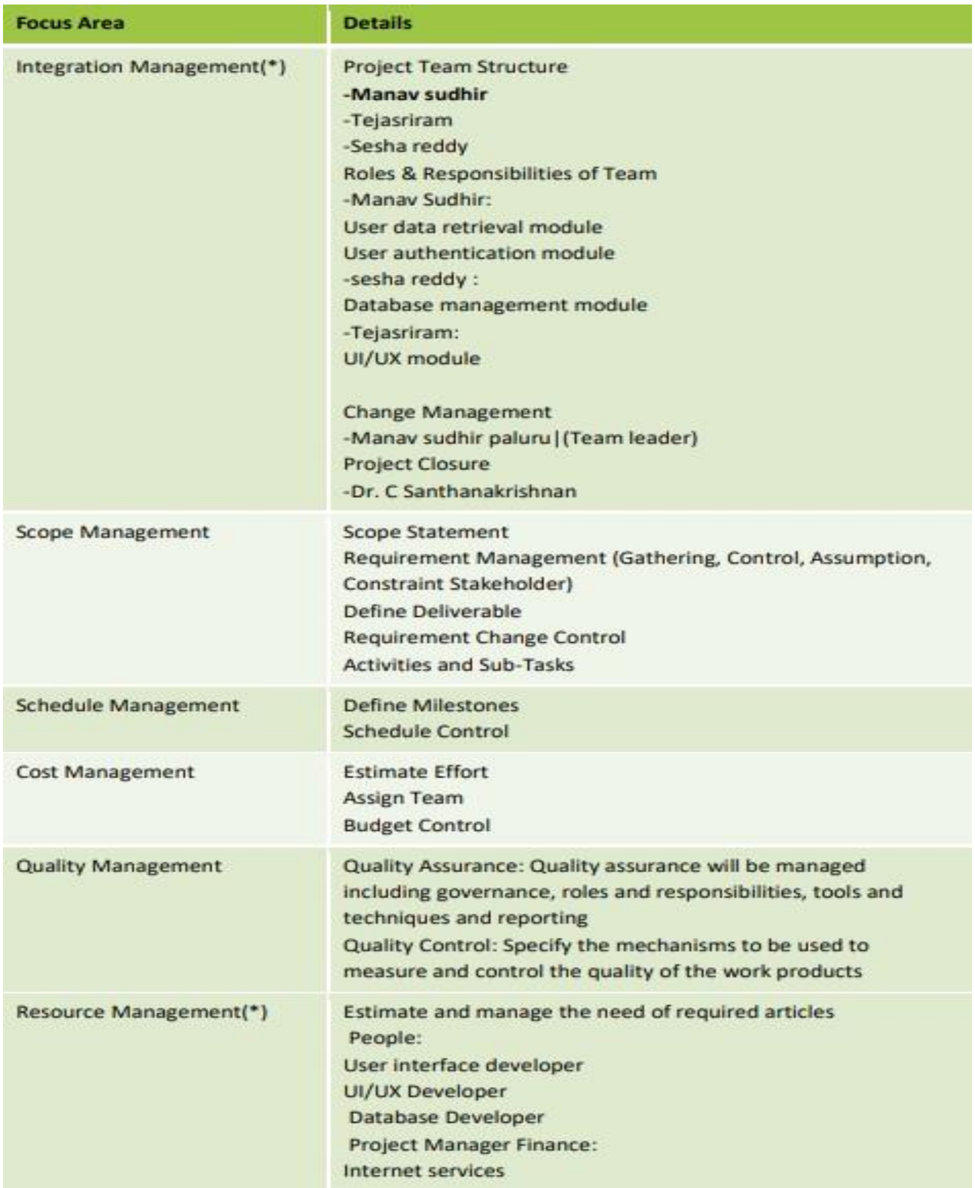
3)The system shall login a customer within 4 seconds.

4)The system shall support 3000 Concurrent sessions.

CHAPTER 4
PROJECT PLAN AND EFFORT

4.1 PROJECT MANAGEMENT PLAN

Table 4.1:project management plan



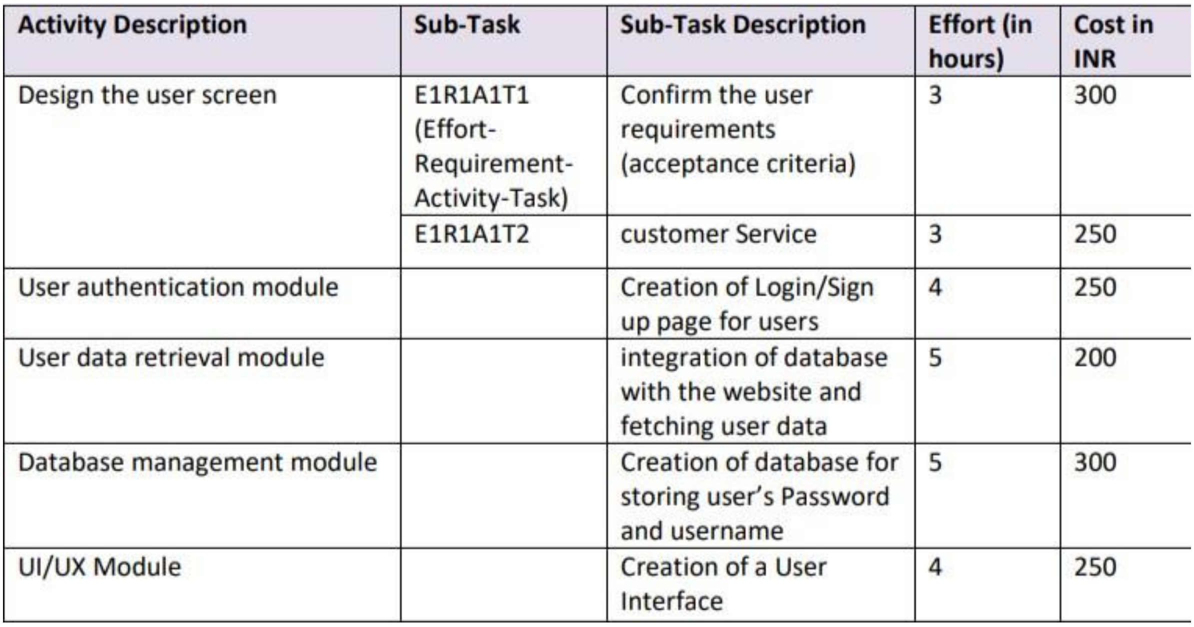
Focus Area	Details
Integration Management(*)	Project Team Structure **-Manav sudhir** -Tejasriram -Sesha reddy Roles & Responsibilities of Team -Manav Sudhir: User data retrieval module User authentication module -sesha reddy : Database management module -Tejasriram: UI/UX module Change Management -Manav sudhir paluru|(Team leader) Project Closure -Dr. C Santhanakrishnan
Scope Management	Scope Statement Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) Define Deliverable Requirement Change Control Activities and Sub-Tasks
Schedule Management	Define Milestones Schedule Control
Cost Management	Estimate Effort Assign Team Budget Control
Quality Management	Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques and reporting Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products
Resource Management(*)	Estimate and manage the need of required articles People: User interface developer UI/UX Developer Database Developer Project Manager Finance: Internet services

4.2 Effort and Cost Estimation

Estimation is done using story points based on the current level of understanding of the requirements. The initial estimates can get revised when the Product Backlog is refined by the Product Owner and the Development Team. The estimate for a story is final only when the story is committed to be delivered in a Sprint.

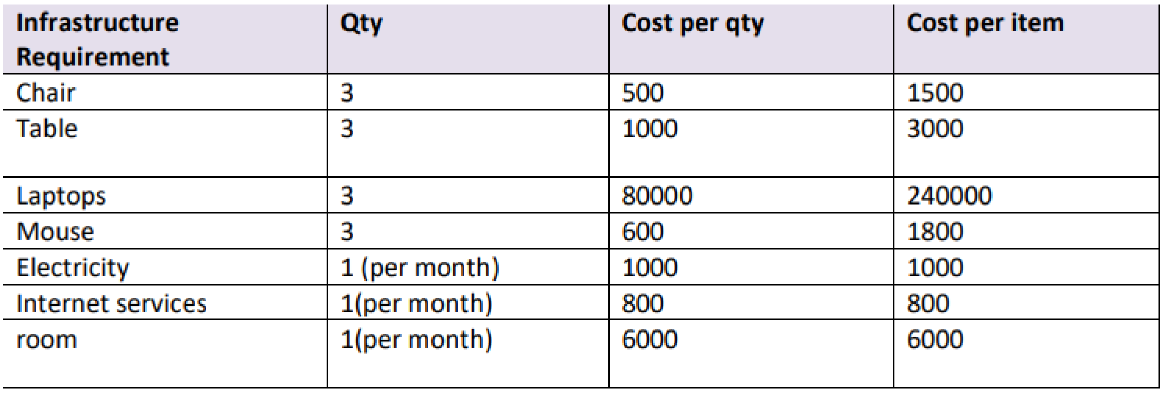
User stories marked are **MVP (Minimum Viable Product)** are the ones that must be done before the first release. All stories that constitute the MVP for the first release are identified. MVP for subsequent releases will be identified by the Product Owner during the development phase of the first release.

Table4.2 effort and cost estimation



Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the user screen	E1R1A1T1 (Effort-Requirement-Activity-Task)	Confirm the user requirements (acceptance criteria)	3	300
	E1R1A1T2	customer Service	3	250
User authentication module		Creation of Login/Sign up page for users	4	250
User data retrieval module		integration of database with the website and fetching user data	5	200
Database management module		Creation of database for storing user's Password and username	5	300
UI/UX Module		Creation of a User Interface	4	250

Table 4.3 Infrastructure/Resource cost



Infrastructure Requirement	Qty	Cost per qty	Cost per item
Chair	3	500	1500
Table	3	1000	3000
Laptops	3	80000	240000
Mouse	3	600	1800
Electricity	1 (per month)	1000	1000
Internet services	1(per month)	800	800
room	1(per month)	6000	6000

Table 4.4 Maintenance and Support Cost [OpEx]

Infrastructure Requirement	Qty	Cost per qty	Cost per item
Hosting Environment (Firebase)	5GB	1,000	5,000
Laptops	3	1,00,000	3,00,000
Wifi-Internet	3	500	1,500

Table 4.5- Identification of Team Members

Name	Role	Responsibilities
Manav	Key Business User (Product Owner)	Provide clear business and user requirements
Manav	Project Manager	Manage the project
Sesha	Business Analyst	Discuss and Document Requirements
Sesha	Technical Lead	Design the end-to-end architecture
Ram	UX Designer	Design the user experience
Ram	Frontend Developer	Develop user interface
Sesha	Backend Developer	Design, Develop and Unit Test Services/API/DB
Manav	Cloud Operations	Provision required Services
Sesha	Tester	Define Test Cases and Perform Testing

Table 4.6 - Responsibility Assignment Matrix

Activity	Scrum Team			Business Stakeholders
	Product Owner	Development Team	Scrum Master	
User Requirements	A	I	I	C
Implementing Agile Practices	R	R	A	
Project Status Reporting	R	R	A	I
Development activities (coding testing, automation)	I	A	C	
Providing feedback on done user stories	R	I	I	A

A	Accountable
R	Responsible
C	Consult
I	Inform

CHAPTER 5

WORK BREAKDOWN STRUCTURE AND RISK ANALYSIS

5.1 Work Breakdown Structure:

The aim of these processes is to ensure that various Project tasks are well coordinated and they.Meet the various project objectives including timely completion of theproject. Project Planning is an aspect of Project Management that focuses a lot on Project Integration. The project plan reflects the current status of all project activities and is used to monitor and control the project.The Project Planning tasks ensure that various elements of the Project are coordinated andtherefore guide the project execution.

Breaking work into smaller tasks is a common productivity technique used to make the work more manageable and approachable. For projects, the Work Breakdown Structure (WBS) is thetool that utilizes this technique and is one of the most important project management documents.It singlehandedly integrates scope, cost and schedule baselines ensuring that project plans are inalignment.

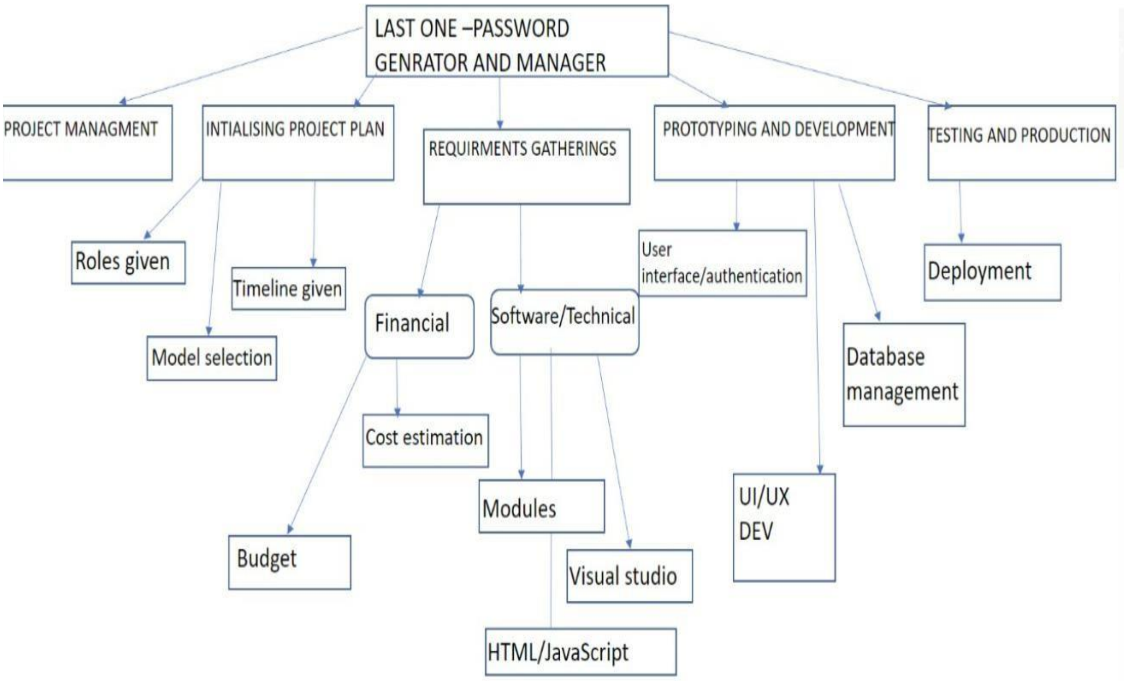
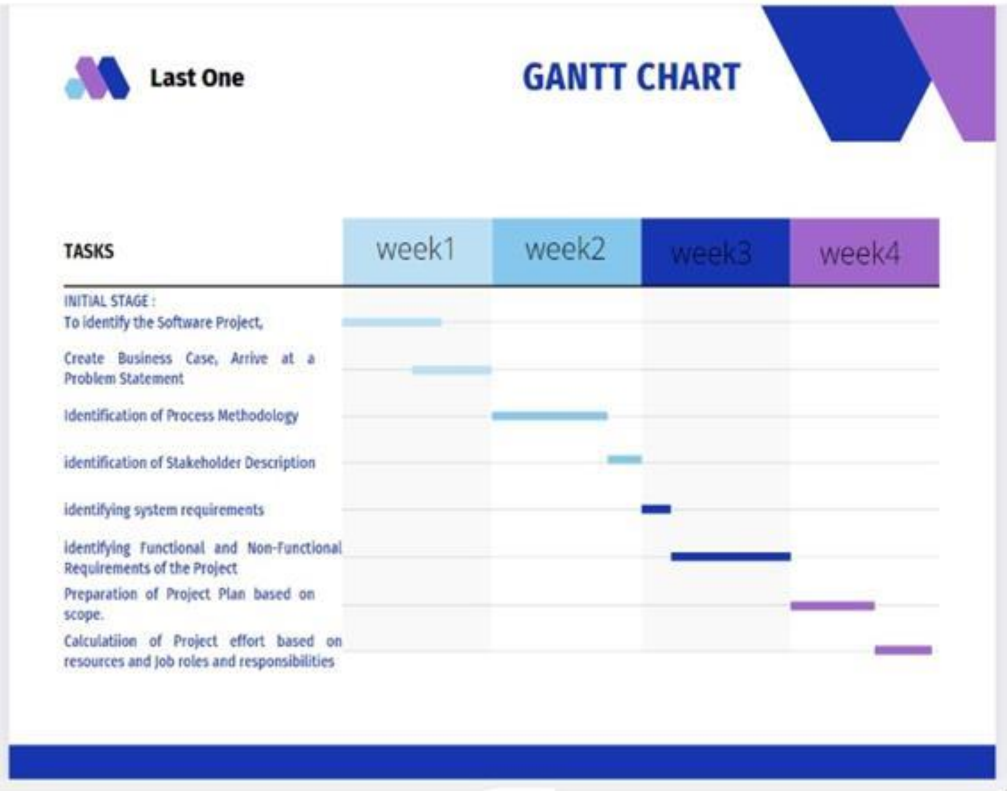


Figure 5.1 - Work Breakdown Structure

5.2 TIMELINE-GANTT CHART:



Last One

GANTT CHART

TASKS

week1

week2

week3

week4

INITIAL STAGE :
To identify the Software Project,

Create Business Case, Arrive at a Problem Statement

Identification of Process Methodology

identification of Stakeholder Description

identifying system requirements

identifying Functional and Non-Functional Requirements of the Project

Preparation of Project Plan based on scope.

Calculatiion of Project effort based on resources and Job roles and responsibilities

Figure 5.2 - Timeline – Gantt Chart

5.3- Risk Analysis – SWOT

5.3.1 STRENGTHS:

1)User-friendly software
2)It can hold 50 passwords
3) unique password Generator
4)Spots weak passwords
5)Quick sign-in

5.3.2 WEAKNESS:

1) Single point of failure
2) Does not auto-fill the passwords
3) It doesn’t work with all the browsers

5.3.3 OPPORTUNITIES:

1)Innovative marketing strategies
2)New data base technology
3)Development to an app (Mobile usage)

5.3.4 THREATS:

1) Keyloggers
2) Brute force/Cracking
3) Password recovery/Reset systems
4)Weak Master password

CHAPTER-6

SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM

To Design a System Architecture, use case and Class Diagram for Passport service system.

6.1 SYSTEM ARCHITECTURE

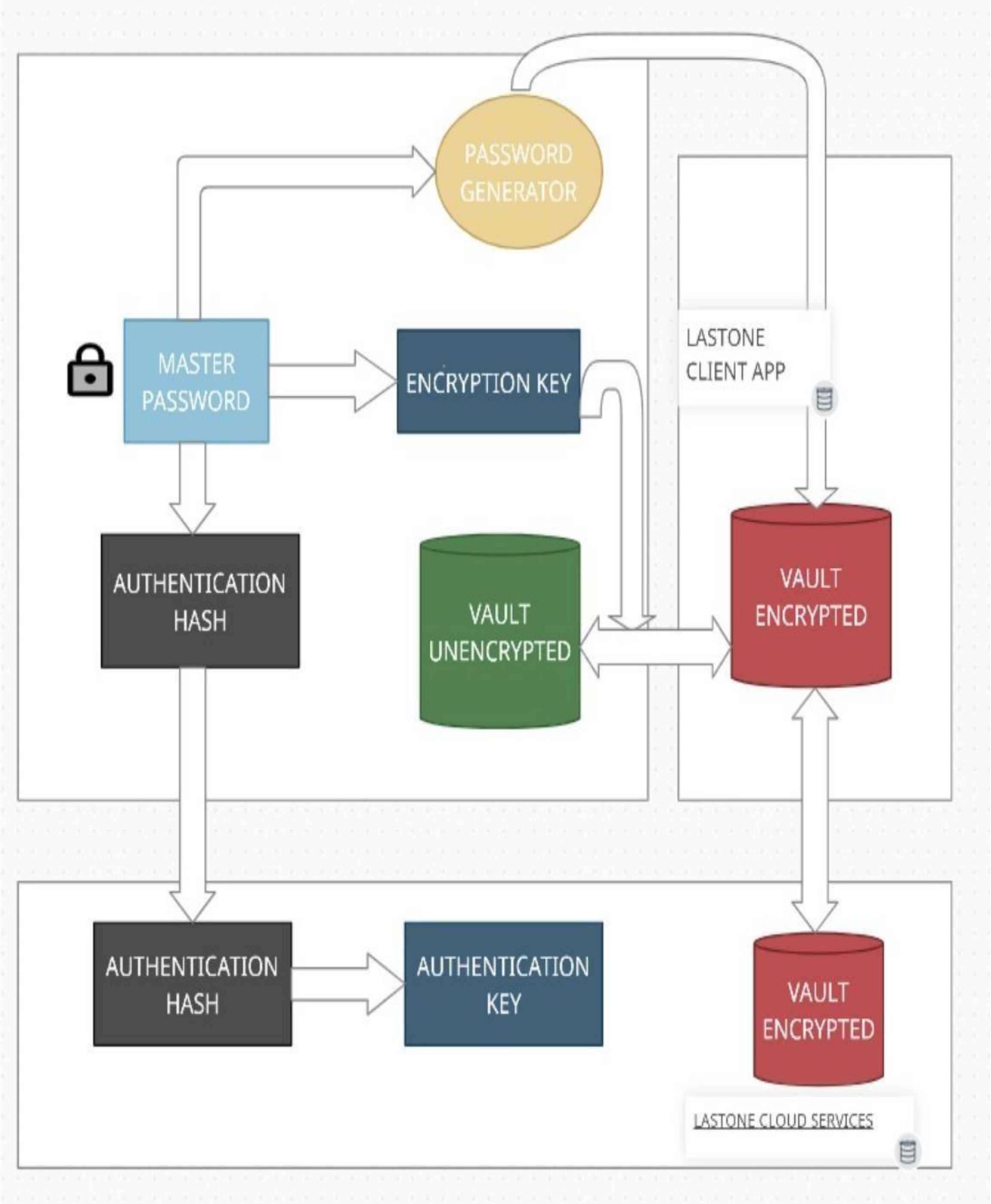


Fig 6.1 system Architecture

6.2 USE CASE DIAGRAM:

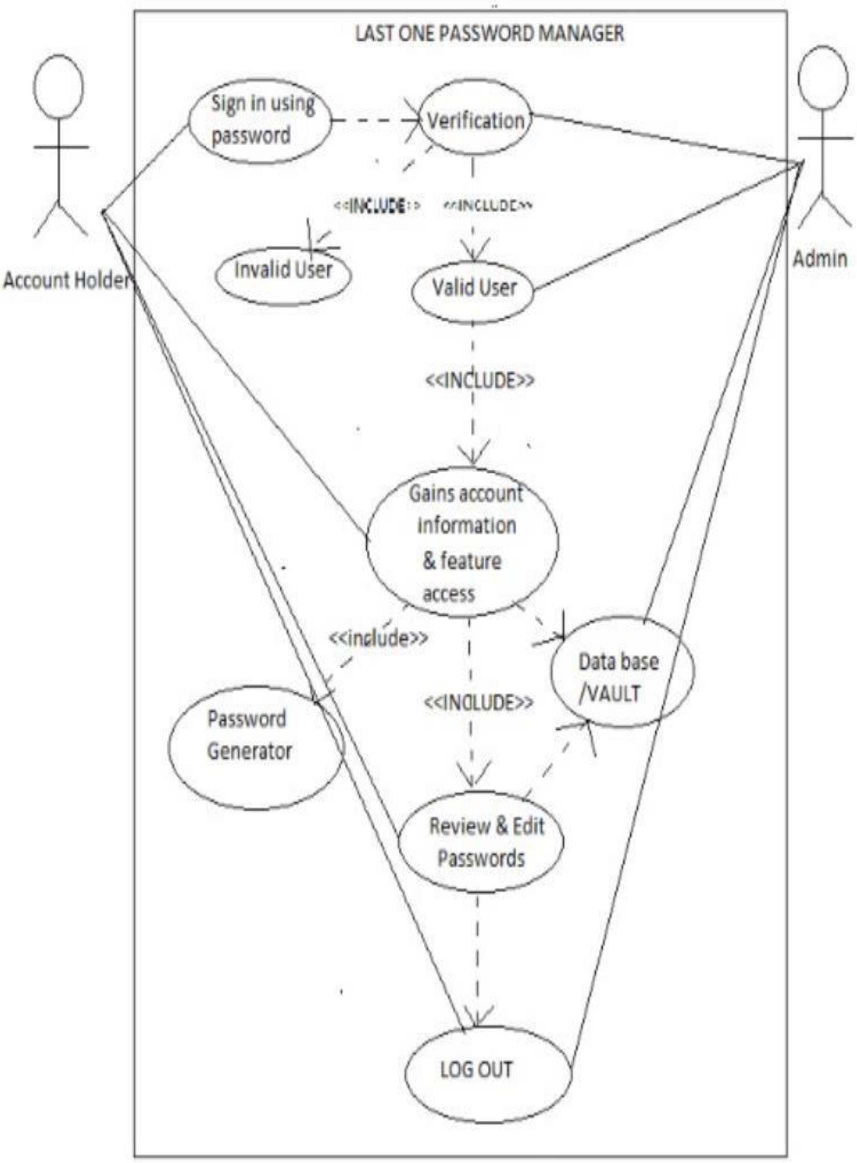


Fig 6.2 Use Case Diagram

6.3 CLASS DIAGRAM:

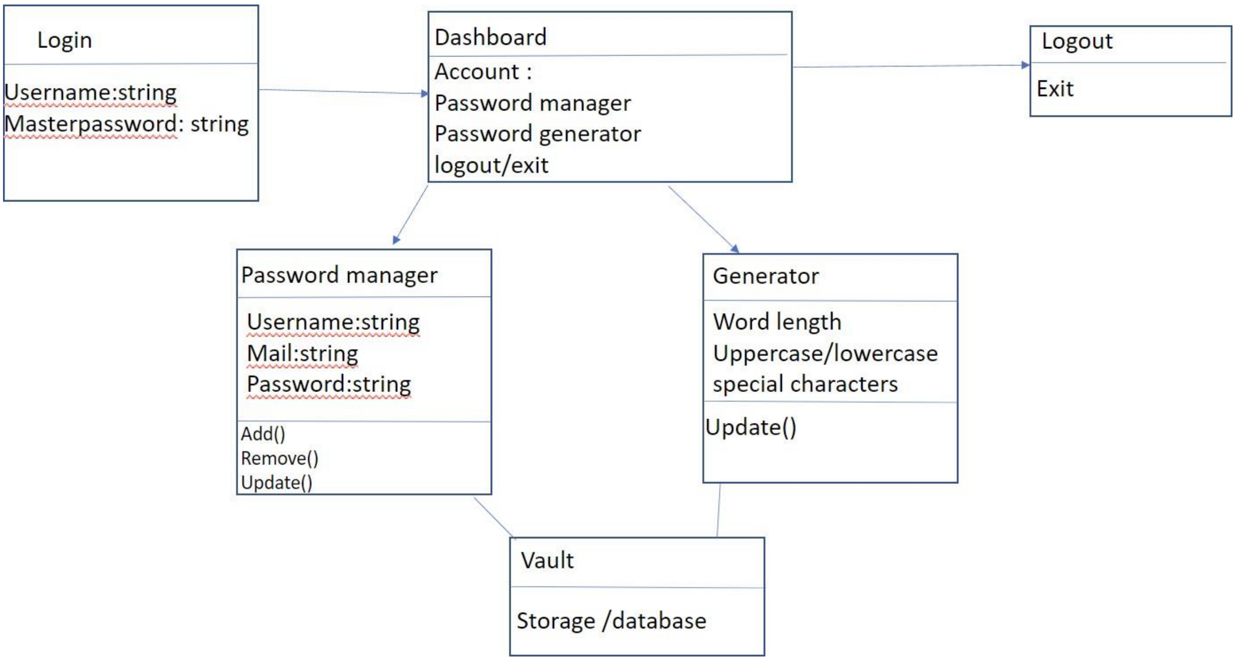


Fig 6.3 Class Diagram

CHAPTER-7

ENTITY RELATIONSHIP DIAGRAM

To create the Entity Relationship Diagram for Secure Digital Authentication System.

T

ER Diagram stands for Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases. ER diagrams are created based on three basic concepts: entities, attributes and relationships. ER Diagrams contain different symbols that use rectangles to represent entities, ovals todefine attributes and diamond shapes to represent relationships. ER Diagram includes many specialized symbols, and its meanings make this model unique. The purpose of ER Diagram is to represent the entity framework infrastructure

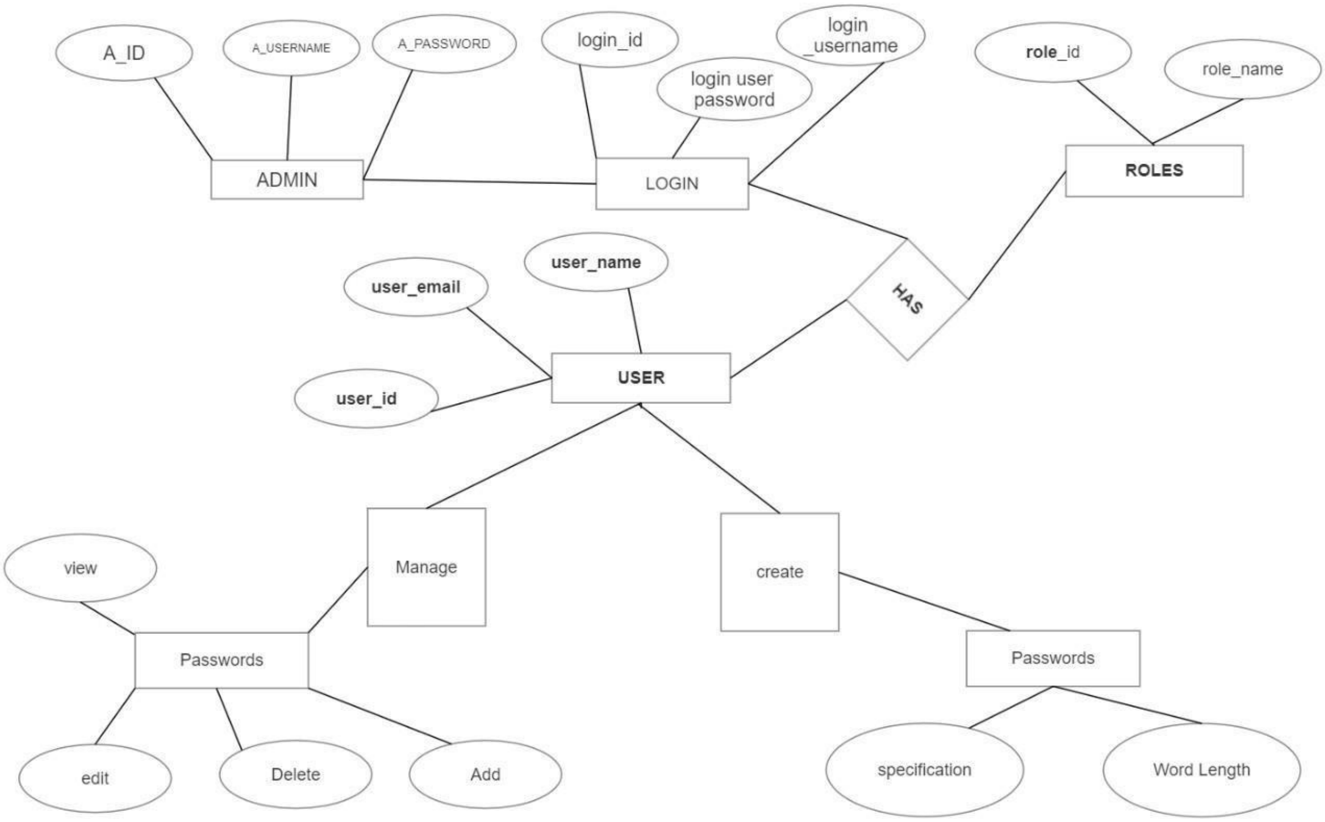


Figure 7.1 Entity Relationship Diagram

CHAPTER-8
DATA FLOW DIAGRAM

To develop the data flow diagram up to level 1 for the Passport service system.

8.1 Data Flow Diagram:

The DFD takes an input-process-output view of a system. That is, data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software. Data objects are represented by labeled arrows, and transformations are represented by circles (also called bubbles). The DFD is presented in a hierarchical fashion. That is, the first data flow model (sometimes called a level 0 DFD or context diagram) represents the system as a whole. Subsequent data flow diagrams refine the context diagram, providing increasing detail with each subsequent level.

The data flow diagram enables you to develop models of the information domain and functional domain. As the DFD is refined into greater levels of detail, you perform an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of data as it moves through the processes that embody the application.

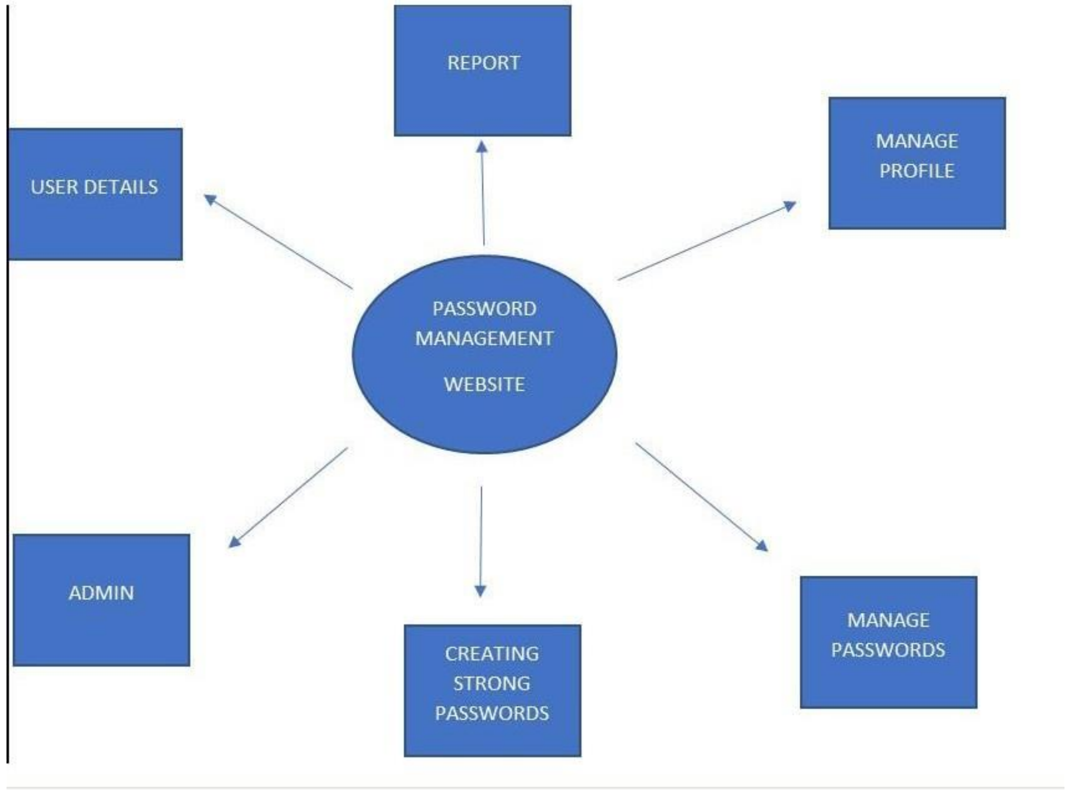


Figure 8.1 Main Page (Level 0)

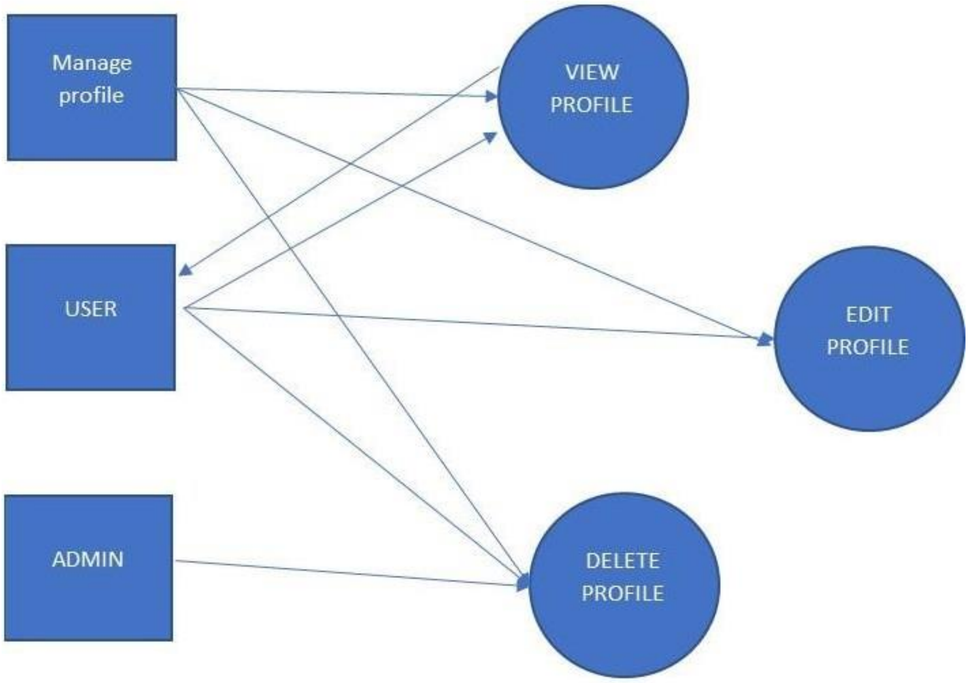


Figure 8.2 Manage Profile (Level 1)

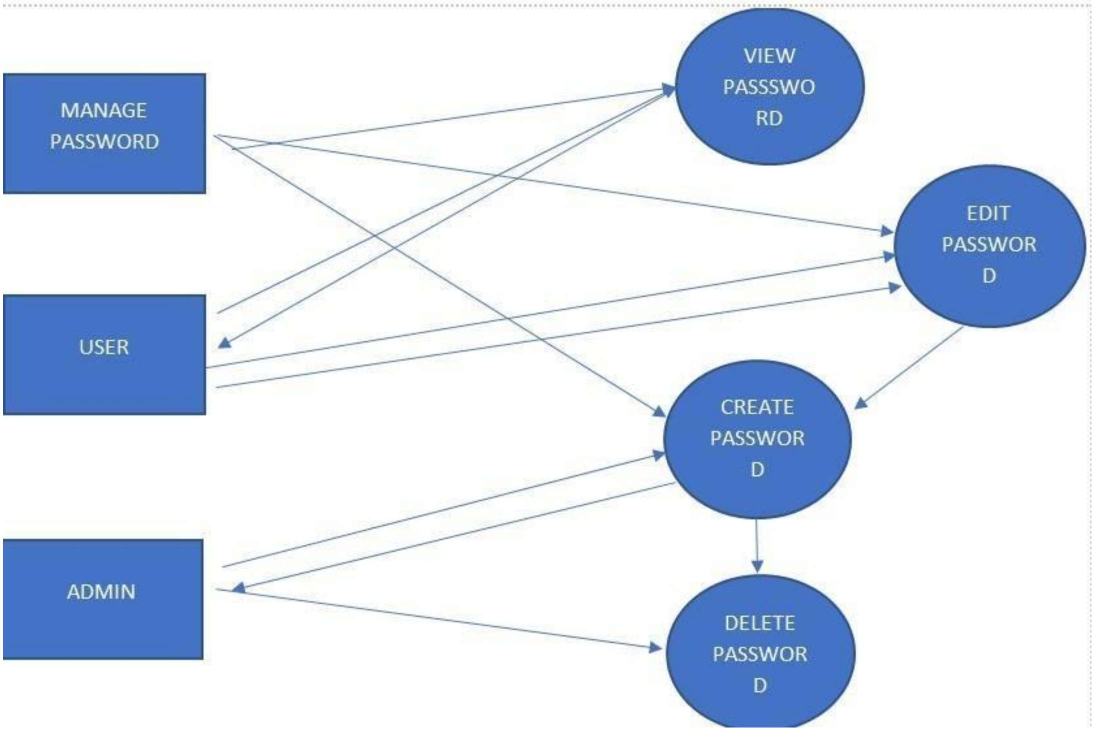


Figure 8.3 Manage Password (Level 1)

CHAPTER-9
SEQUENCE AND COLLABORATION DIAGRAM

To create the sequence and collaboration diagram for the Passport service system.

9.1 Sequence Diagram:

A sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering. Here it depicts the processes involved and the sequence of messages exchanged between the processes for passport application needed to carry out the functionality.

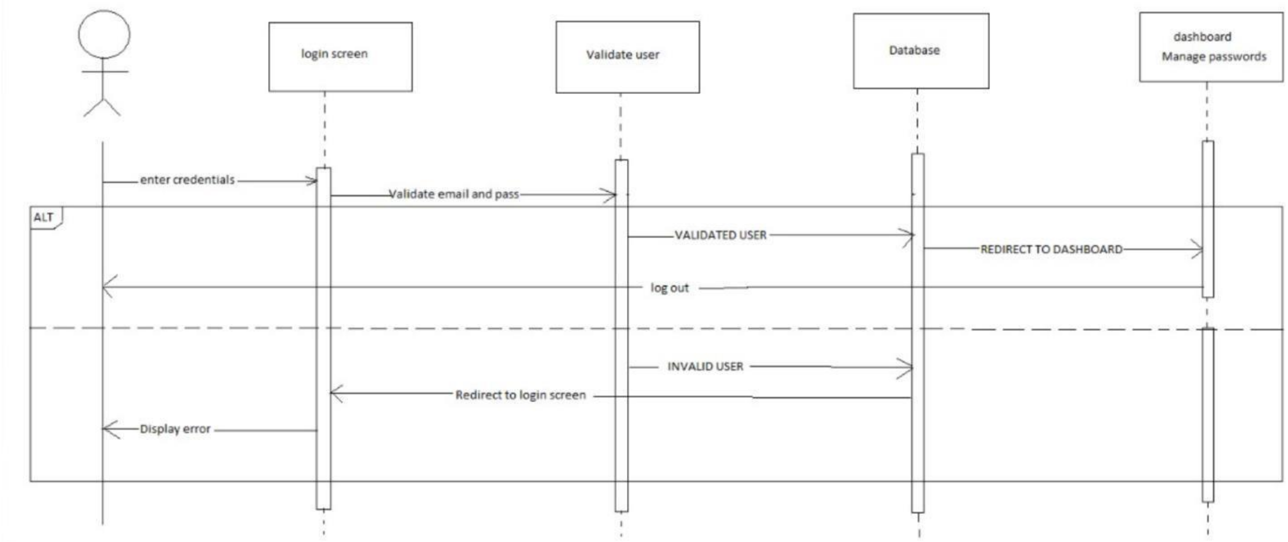


Figure 9.1 Sequence Diagram

9.2 Collaboration Diagram:

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use-case and define the role of each object.

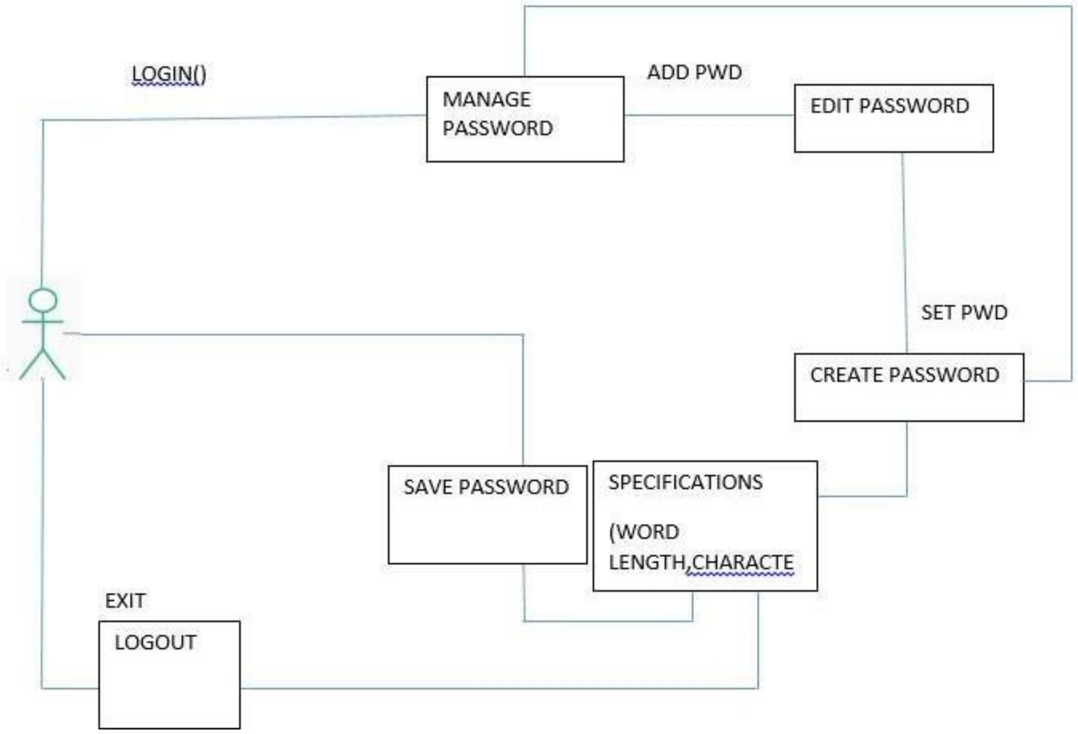


Figure 9.2 Collaboration diagram

CHAPTER-10

DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE

To develop the testing framework and/or user interface framework

10.1 Test Plan

10.1.1 Functional Testing Scope

The following user stories are covered in the functional testing for release . Any change to this has to be managed by the Product Owner in collaboration with the Development Team.

Table 10.1 functional testing

No.	User Story No.	Title
1	USTR1	as a user I want to provide information about me so that i a can register myself in the platform.
2	USTR2	as a user I want to upload documents so that I can prove my eligibility for using the platform
3	USTR3	AS an administrator I want to view the documents uploaded by user so that I can approve or reject them.
4	USTR5	AS a user i want to create a password using key words (word length, characters)
5	USTR6	As a user I want to manage all passwords in my account .
6	USTR7	As a user I want to add or remove passwords

10.1.2Non-Functional Testing Scope

The following NFRs are covered by the testing scope for release . Any change to this has to be managed by the Product Owner in collaboration with the Development Team.

Table 10.2 Non Functional testing

NFR #	Category of NFR	Requirement Specification
NFR1	Performance	The system should have a response time of less than 3 seconds with a load of 100 transactions per second on a 4G mobile network.
NFR4	Security	Authentication and authorization mechanism for user access.
NFR5	Scalability	The system should be easily scalable with additional infrastructure.

NFRs not included in the testing scope will be verified using a combination of manual and automated process as outlined in the following sections.

Table 10.3 Types of Testing, Methodology, Tools

Category	Methodology	Tools Required
Unit Testing	Automated	Unit testing framework (PyTest).
Functional Testing	Manual	Word Template.
Performance & Scalability Testing (NFR1 & NFR5)	Automated	Load testing tool (JMeter).
Security (NFR4)	Manual	Word Template.

10.3 NFR Verification

NFRs not included in the testing scope will be verified using a combination of manual and automated process as outlined in the following sections.

Table 10.4 NFR Verification

NFR #	Type	Methodology	Tools Required
NFR6	Maintainability	Automated	Automated static code analysis tool integrated to the automated build (SonarQube).
NFR2	Usability	Manual	Review based on checklist. Feedback from Product Owner.
NFR3	Availability	Manual	Expert review of deployment architecture.
NFR7	Compliance	Manual	Expert review of design.

CHAPTER-11

TEST CASES

11.1 Test Case

Table 11.1 Functional Test Cases

Test ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Verify user credentials	1.Accept only Valid email id and correct password 2.Captcha should not be left empty and mismatch is not accepted	1. Enter registered password and email 2. Verify captcha and click enter	User should be navigated to the dashboard	The user grants access to the website dashboard	Pass / Failure	success
2	View the user's saved passwords	Passwords should load successfully in the data base	1.User clicks on different accounts for the password2	Users will be able to check and modify their password	The user could see and modify his account	Pass / Failure	success
3	Generate a password	A unique password should be created and displayed.	1.user selects the specification of the password	User can generate a unique password	The password is created successfully	Pass / Failure	success

11.1.2 Non-Functional Test Cases

Table 11.2 Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Quit	Quit the website	User clicks on the log out button.	Website should close	Website closes	Pass/ Failure	success

CHAPTER 12

Test Case Reporting

12.1 Test Report

Testing can be done either manually or automation testing. Automation testing finally leads to regression testing. Manual testing leads to unit testing. To check for business requirements, we have used unit testing that will be performed on the entire function system, i.e. for each and every function of the calculator. This testing tests each and every function with regards to its output as expected by the user.

Table 12.1 Test Report

Category	Progress Against Plan	Status
Functional Testing	Amber	In-Progress
Non-Functional Testing	Amber	In-Progress

Table 12.2 Current status of testing

Functional	Test Case Coverage (%)	Status
LOGIN MODULE	75%	In-Progress
User dashboard Module	40%	In-Progress
Password creator module	100%	Completed
Data base module	40%	In-Progress

CHAPTER 13

CONCLUSION

It is concluded that the application works well and satisfy the users. The application is tested very well and errors are properly debugged. The site is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested. The site works according to the restrictions provided in their respective browsers. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one. The speed of the transactions become more enough now. User friendliness is provided in the application with various controls provided by system rich user interface. This system makes the overall project management much easier and flexible. Variousclasses have been used for maintain the details of all the users and catalogue.

Authentication is provided for this application only registered users can access. Report generation features is provided using to generate different kind of reports. In this system the individual fills all the information or details required for passport application through online. The individual has tofill all the details and can’t leave any field as blank.

REFERENCES

[1] https://www.pmi.org/

[2] https://www.projectmanagement.com/

[3] https://www.tpsgc-pwgsc.gc.ca/biens-property/sngp-npms/ti-it/ervcpgpm-dsfvpmpt-eng.html

[4] https://scrumguides.org/

[5] https://azure.microsoft.com/en-in/pricing/details/virtual-machines/windows/

APPENDIX – I (SAMPLE CODE)

```
html: <!DOCTYPE html>
<html lang="en">
  <head>
    <title>PASSWORD GENERATOR APP</title>
    <link rel="stylesheet" href="layout.css" />
    <script src="script.js" defer></script>
  </head>
  <body>
    <form id="passwordGeneratorForm">
      <div class="container">
        <h2>Password Generator Application</h2>
        <div class="result__container">
        <span id="result"></span>
        <button id="copy">Copy</button>
      </div>
      <div class="options">
        <div class="option">
          <label>Length</label>
          <input type="number" id="length" min="4" max="20" value="10">
        </div>
        <div class="option">
          <label>Include Uppercase</label>
          <input type="checkbox" id="uppercase" checked>
        </div>
        <div class="option">
          <label>Include Numbers</label>
          <input type="checkbox" id="numbers" checked>
        </div>
        <div class="option">
          <label>Include Symbols</label>
          <input type="checkbox" id="symbols" checked>
        </div>
      </div>
      <button class="btn" id="generate" type="submit">Generate Password</button>
      </div>
    </form>
  </body>
</html>
```

CSS

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  height: 100vh;
  width: 100vw;
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  font-family: 'Oswald', sans-serif;
  background-color: #39378f;
}

.container {
  padding: 1rem 1.5rem;
  border: 1px solid black;
  width: 350px;
  background-color: #4abd15;
}

h2 {
  text-align: center;
  padding: 15px 0;
}


.option {
  display: flex;
  justify-content: space-between;
  padding: 4px;
}
.result_container {
  height:  50px;
  width: 100%;
  display: flex;
  justify-content: space-between;
  align-items: center;
  border: 1px solid black;
  padding: 0 5px;
}

.result-container #result {
  word-wrap: break-word;
  max-width: calc(100% - 40px);
}

.result_container #copy {
  height: 40px;
  width: 40px;
  background-color: #eb1606;
  color: #ffffff;
  border: none;
  cursor: pointer;
  outline: none;
}

.result_container #copy:hover {
  background-color: #1c2541;
  color: #ffffff;
}

#generate {
  height: 40px;
  width: 100%;
  border-radius: 10px;
  border: none;
  background-color: #0b132b;
  color: #ffffff;
  font-size: 15px;
  font-weight: bold;
  cursor: pointer;
  outline: none;
}
```

```
#generate:hover {
  background-color: #5bc0be;
  color: #ffffff;
}
```

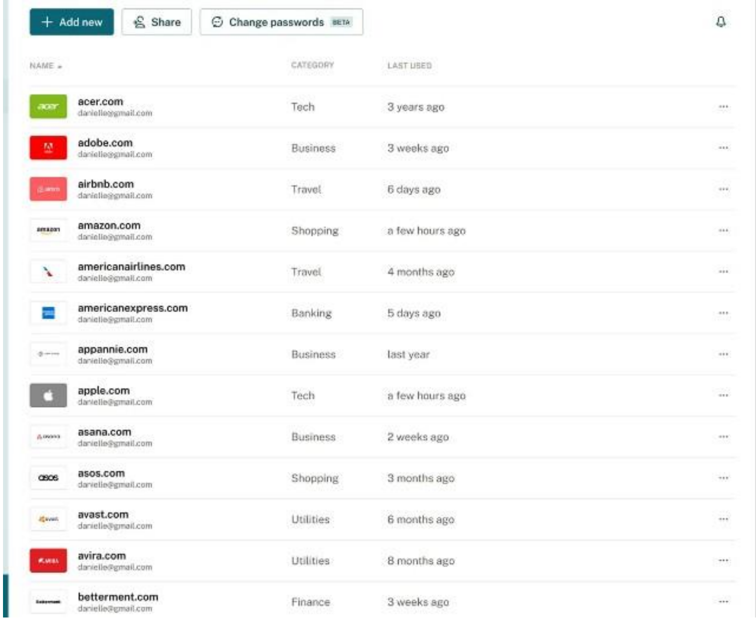
JS

```
: // Getting the DOM Elements
const resultDOM = document.getElementById("result");
const copybtnDOM = document.getElementById("copy");
const lengthDOM = document.getElementById("length");
const uppercaseDOM = document.getElementById("uppercase");
const numbersDOM = document.getElementById("numbers");
const symbolsDOM = document.getElementById("symbols");
const generatebtn = document.getElementById("generate");
const form = document.getElementById("passwordGeneratorForm");
Generating Character Codes For The Application
const UPPERCASE_CODES = arrayFromLowToHigh(65, 90);
const LOWERCASE_CODES = arrayFromLowToHigh(97, 122);
const NUMBER_CODES = arrayFromLowToHigh(48, 57);
const SYMBOL_CODES = arrayFromLowToHigh(33, 47)
  .concat(arrayFromLowToHigh(58, 64))
  .concat(arrayFromLowToHigh(91, 96))
  .concat(arrayFromLowToHigh(123, 126));
// Character Code Generating Function
function arrayFromLowToHigh(low, high) {
const array = [];
  for (let i = low; i <= high; i++) {
    array.push(i);
  }
  return array;
}
// The Password Generating Function
let generatePassword = (
characterAmount,
  includeUppercase,
  includeNumbers,
  includeSymbols
) => {
  let charCodes = LOWERCASE_CODES;
  if (includeUppercase) charCodes = charCodes.concat(UPPERCASE_CODES);
  if (includeSymbols) charCodes = charCodes.concat(SYMBOL_CODES);
  if (includeNumbers) charCodes = charCodes.concat(NUMBER_CODES);
  const passwordCharacters = [];
  for (let i = 0; i < characterAmount; i++) {
    const characterCode =
      charCodes[Math.floor(Math.random() * charCodes.length)];
    passwordCharacters.push(String.fromCharCode(characterCode));
  }
  return passwordCharacters.join("");
};
// Copy password button

copybtnDOM.addEventListener("click", () => {
  const textarea = document.createElement("textarea");
  const passwordToCopy = resultDOM.innerText;
  // A Case when Password is Empty
  if (!passwordToCopy) return;
  // Copy Functionality
  textarea.value = passwordToCopy;
  document.body.appendChild(textarea);
  textarea.select();
  document.execCommand("copy");
  textarea.remove();
  alert("Password Copied to Clipboard");
});
// Checking the options that are selected and setting the password
form.addEventListener("submit", (e) => {
  e.preventDefault();
  const characterAmount = lengthDOM.value;
  const includeUppercase = uppercaseDOM.checked;
  const includeNumbers = numbersDOM.checked;
  const includeSymbols = symbolsDOM.checked;
  const password = generatePassword(
  characterAmount,
    includeUppercase,
    includeNumbers,
    includeSymbols
  );
  resultDOM.innerText = password;
});
```

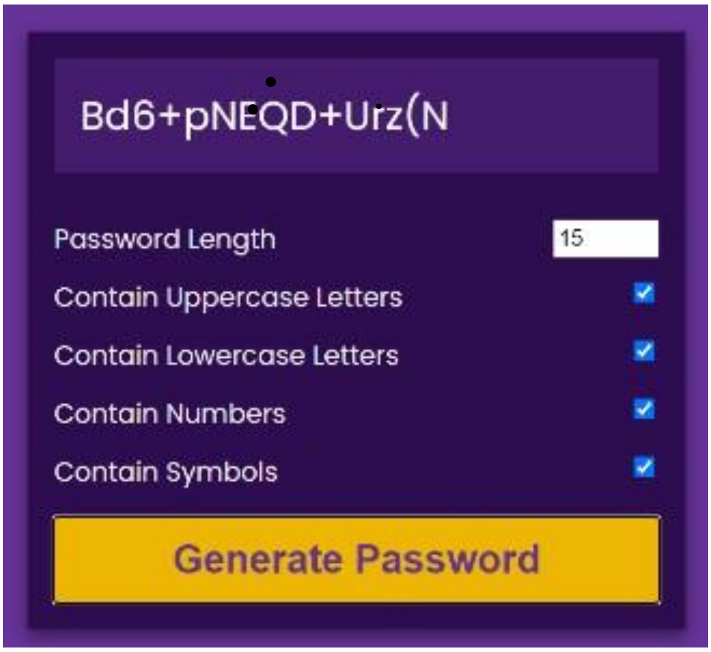
APPENDIX – II (SCREENSHOTS)

Dashboard



NAME	CATEGORY	LAST USED	
acer.com danielle@gmail.com	Tech	3 years ago	···
adobe.com danielle@gmail.com	Business	3 weeks ago	···
airbnb.com danielle@gmail.com	Travel	6 days ago	···
amazon.com danielle@gmail.com	Shopping	a few hours ago	···
americanairlines.com danielle@gmail.com	Travel	4 months ago	···
americanexpress.com danielle@gmail.com	Banking	5 days ago	···
appannie.com danielle@gmail.com	Business	last year	···
apple.com danielle@gmail.com	Tech	a few hours ago	···
asana.com danielle@gmail.com	Business	2 weeks ago	···
asos.com danielle@gmail.com	Shopping	3 months ago	···
avast.com danielle@gmail.com	Utilities	6 months ago	···
avira.com danielle@gmail.com	Utilities	8 months ago	···
betterment.com danielle@gmail.com	Finance	3 weeks ago	···

Password Generator



Bd6+pNEQD+Urz(N

Password Length: 15

Contain Uppercase Letters ☑

Contain Lowercase Letters ☑

Contain Numbers ☑

Contain Symbols ☑