Sri Guru Gobind Singh College of Commerce
University of Delhi

SOFTWARE ENGINEERING PROJECT REPORT

SOC HUB

Complete Solution for Society Management & Event Listing

Team Members

Om Gupta - 21078570037
Ashish Durgapal - 21078570007
Varun Sangai - 21078570055
Om Vaish - 21078570038

Guidance

Mrs. Musarrat Ahmed

Get social with SocHUB

Empowering College Societies & Elevating Events with Ease.

Scan QR Code for a Quick App Walkthrough

CERTIFICATE

This is to certify that **Om Gupta, Ashish Durgapal, Varun Sangai, and Om Vaish** students of **B.Sc. (Hons) Computer Science Semester IV** have submitted the project entitled **"SOCHUB: A Society Management Application"** for the partial fulfilment of the requirements of Software Engineering project.

It embodies the work done by them during semester IV of their course under the due supervision of **Mrs. Musarrat Ahmed.**

_____

Signature

ACKNOWLEDGEMENT

We would like to sincerely thank **Mrs. Musarrat Ahmed**, our software engineering instructor, for allowing us to work with her on this project.

We sincerely cherish and are grateful for her esteemed advice and support during this endeavour. She has our sincere gratitude. We appreciate all of **our professors** for giving us a strong foundation for our studies. They have been a wonderful source of inspiration for us, and we sincerely appreciate them.

Last but not least, we would like to express our gratitude to the **Computer Science department** for providing us with all the necessary resources.

INDEX

OVERVIEW

Welcome to **SOCHUB** - a powerful platform that connects college students with their respective college societys and societies. SOCHUB is designed to help you explore, discover and join a variety of college events and societies, and make the most out of your college life.

As a student, you can use SOCHUB to find and join societies that cater to your interests, hobbies or career aspirations. From cultural societys to academic societies, sports societys to social causes - there is something for everyone on SOCHUB.

Once you have joined a society, you can stay up-to-date with its latest news, events, and activities. You can also connect with other members of the society, exchange ideas, and collaborate on projects and events.

If you are a society leader or member, SOCHUB has a powerful dashboard that allows you to manage your society, create and promote events, and raise funds. You can also use SOCHUB to communicate with your society members, assign tasks, and generate reports.

Overall, SOCHUB is a one-stop solution for college students and others who want to explore their interests, create their communities and connect with like-minded peers.

PROBLEM STATEMENT

In India, there are around **1070 Universities** leading to more than **42,000 colleges** as of **2020** and the number is increasing rapidly and each college having around **20+ societies.**

Making it around **10 lacs college societies** ranging from academic societies to cultural societies.

Managing a society is a hectic task and not only involves team management but getting funds, promoting events, and a hell lot of tedious tasks.

Don't worry **SocHub** comes to the rescue. The only partner you need to run your society smoothly.

What does it offer?

☛ **To Societies:**

- ✓ Efficient Team Management Dashboard
- ✓ Robust Event promotion
- ✓ Makes Fund Raising a cup of tea.

☛ **To Students:**

- ✓ One Stop solution for surfing Events and Competitions.
- ✓ Easy Registration.
- ✓ Efficient schedule tracking.

ADVANTAGES

A college-specific society management app like SocHub offers a unique set of features and advantages that are tailored to the needs of college students and their respective societies. Here are some ways in which it can be better than existing players in the market:

- **Targeted audience:** SocHub is specifically designed for college societies, so you can reach out to a targeted audience interested in your society and its events. Instagram and LinkedIn, on the other hand, have a broad user base that may not be interested in your college social events.
- **Society management tools:** SocHub offers a wide range of society management tools that can help you organize your events, manage your members, and track your finances. Instagram and LinkedIn don't have these tools, so you'll need to use separate software to manage your society.
- **Communication channels:** SocHub offers multiple communication channels, including group chats, private messaging, and announcements, which can help keep your members informed and engaged. Instagram and LinkedIn don't offer these features, so it can be harder to keep everyone on the same page.

- ☛ **Event promotion features:** SocHub has features specifically designed for event promotion, such as event pages, RSVPs, and event reminders. Instagram and LinkedIn don't have these features, so it can be harder to get people to attend your events.
- ☛ **Fund Raising:** SocHub offers societies to get sponsors directly. So that societies don't have to pitch sponsors again and again, they can just upload their event description, and their offerings and sponsors will directly approach them.

Overall, **SocHub provides a comprehensive solution for college societies, offering tools and features that are specifically designed to meet their needs.**

*Existing solutions to our problem & how **SocHub** is better:*

- ✓ **UnStop:** The user interface and experience of this app are somewhat not apt, and it may take some time for users to understand how it works and what it does leading to a steep learning curve. Additionally, there are several features that this app is lacking, such as report generation, fundraising, and many others, that our app provides.
- ✓ **Instagram and Linkedin:** These apps are quite distractive as it offers many other features which are not college-specific.
- ✓ **Whatsapp:** Events promotion leads to spamming. Groups are created for society management which is not an efficient way to manage societies. Our app resolves this issue as that is society specific.

UNIQUE SELLING PROPOSTION

- Seamless Collaboration And Management Of Society Team Members.
- One-Click Report Generation.
- Robust Events Promotion.
- Ideal marketplace for societies and sponsors.
- Make Fundraising A Breeze For Your Event With Just One Click!
- Extremely User-Friendly Interface

INITIAL REQUIREMENTS

☛ **Authentication System:**

- ✓ Sign Up Page
- ✓ Log in Page
- ✓ Third-Party Authentication

☛ **Home Screen:**

- ✓ Event Library (Market Place)
- ✓ Home
- ✓ Videos
- ✓ Polls
- ✓ Posts
- ✓ Registration

☛ **Society Management Dashboard:**

- ✓ Team
- ✓ Meet
- ✓ Deadlines
- ✓ Roadmap (Tentative)
- ✓ Report Generation

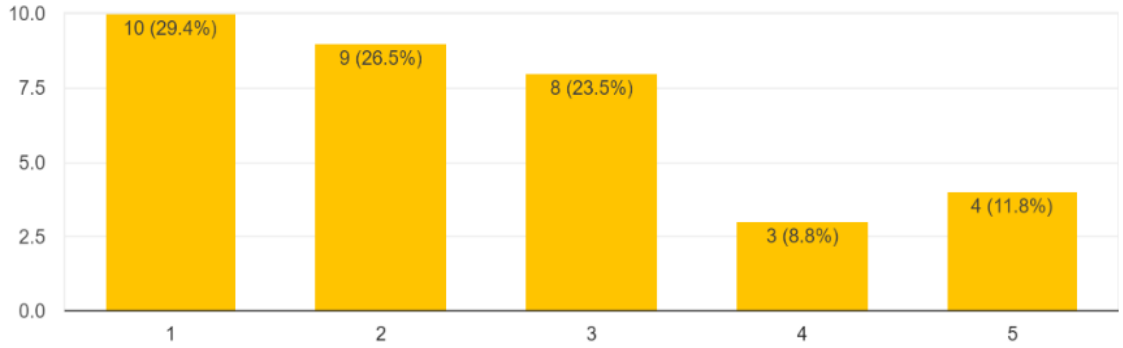
☛ **Society - Sponsor service (Fundraising Model)**

- ✓ Mutual Marketing
- ✓ MoU

REQUIREMENT ANALYSIS

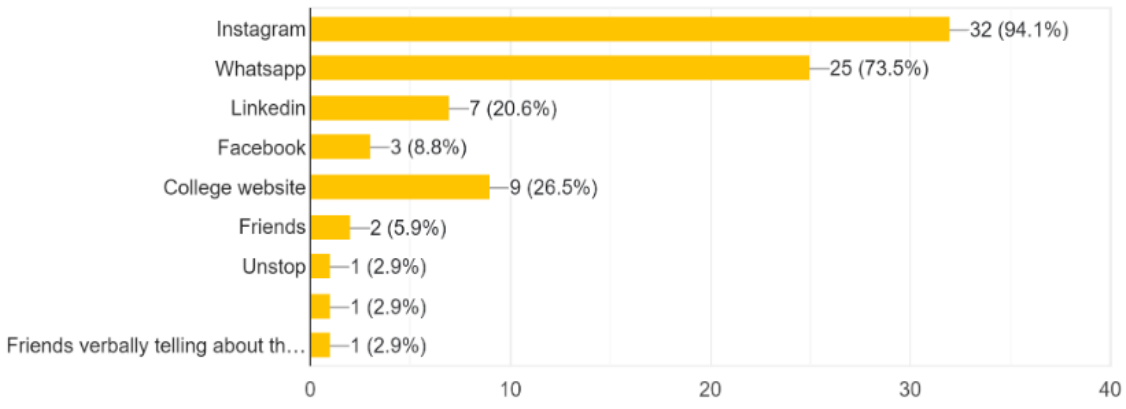
How often were you able to find about events that happen in different colleges across India?

34 responses



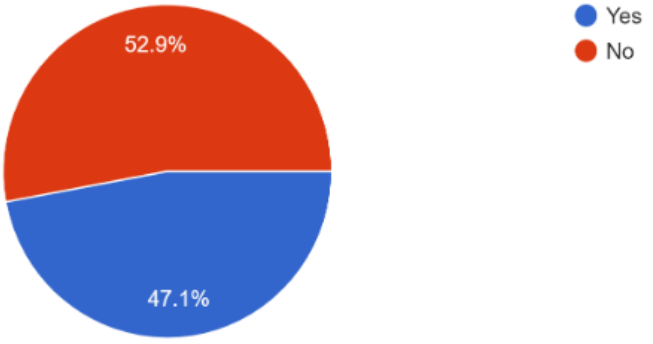
What sources you currently use for finding these events/competitions?

34 responses



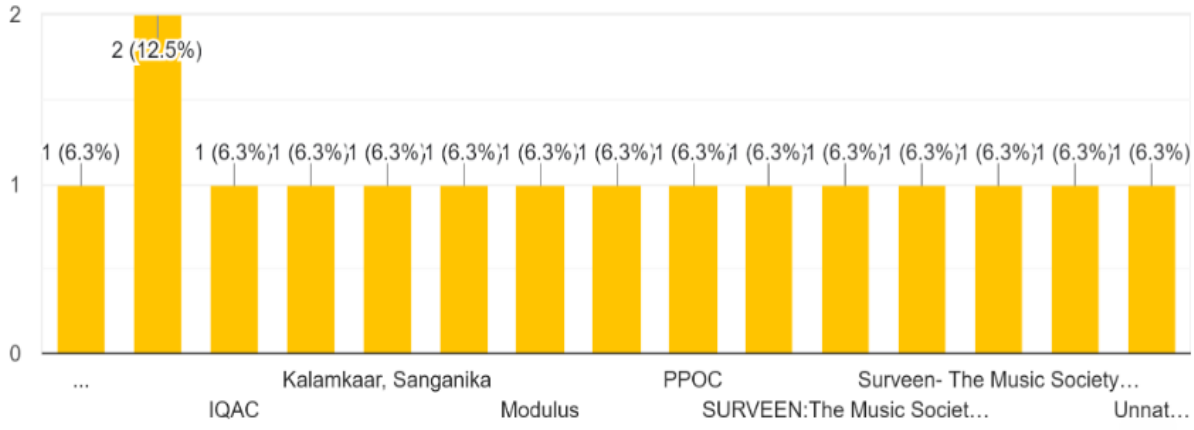
Are you (or was a) part of any college society?

34 responses



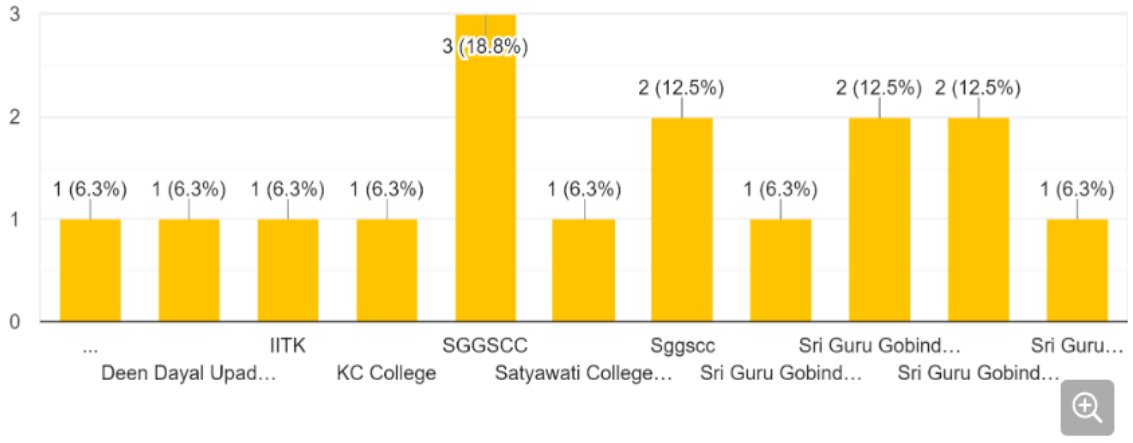
Society Name

16 responses



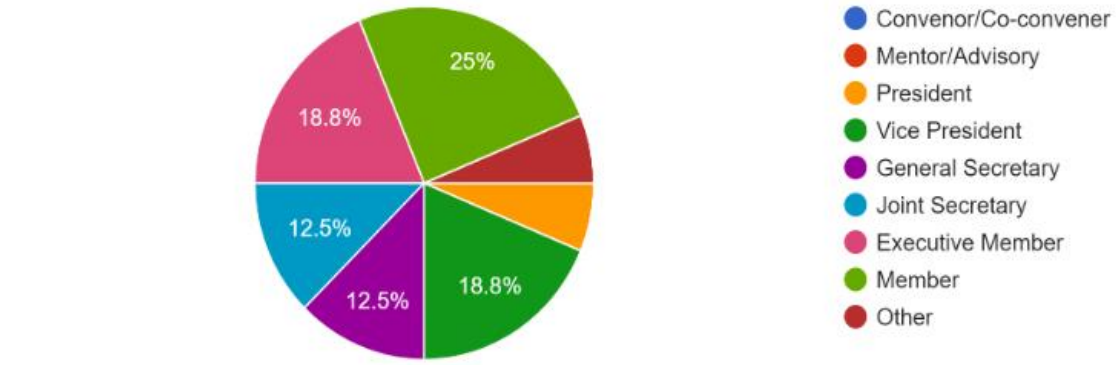
College Name

16 responses



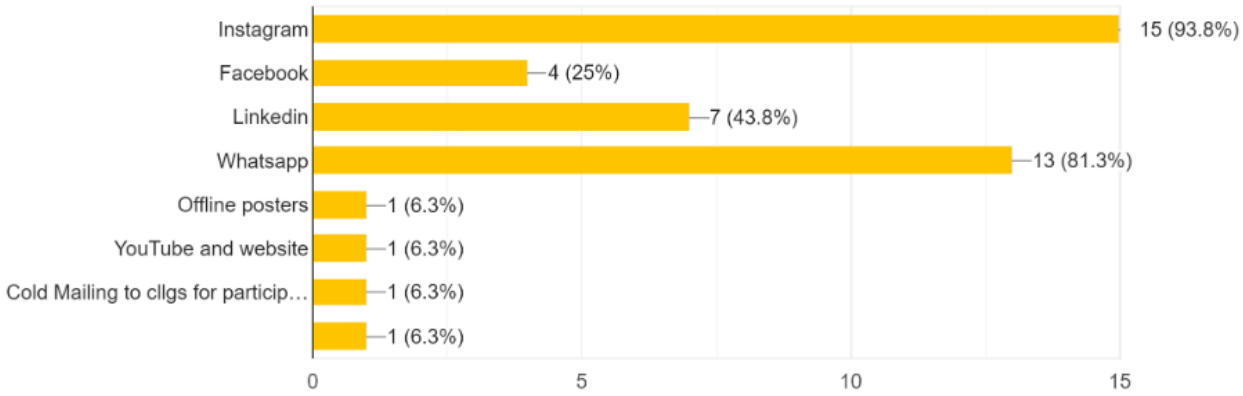
Designation

16 responses



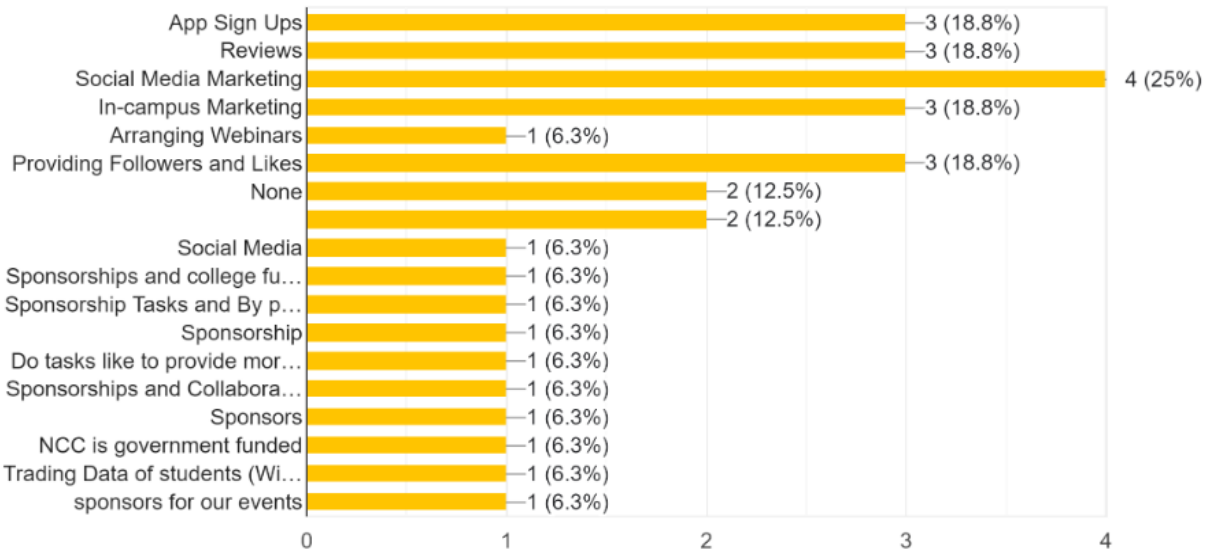
What resources do you currently use for marketing purposes?

16 responses



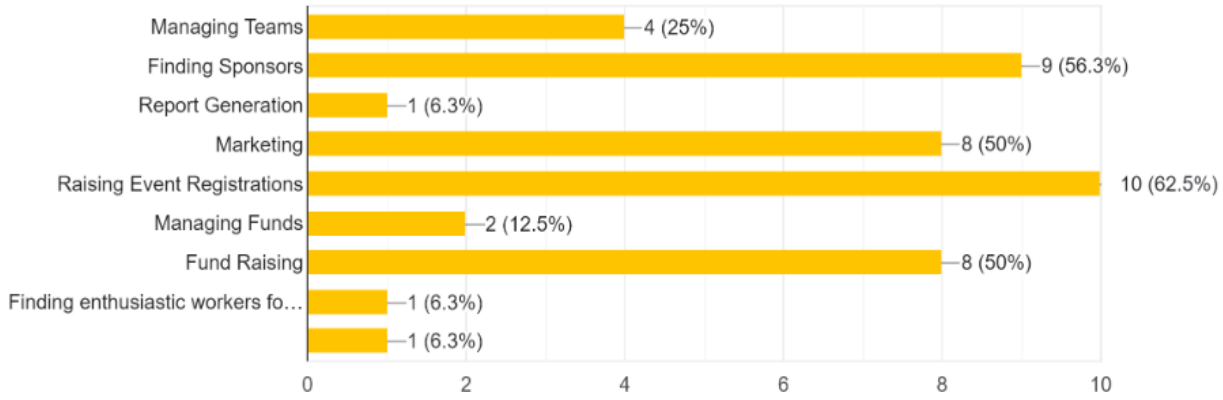
What methods you currently use to raise funds for your society?

16 responses



Challenges faced:

16 responses



Any suggestions you have for us?

7 responses

There are not much sources for marketing . So if it is possible then please do something related to this

Sponsors and sponsorship tasks
Getting more and more people on your website so we are able to do good marketting

Track the status of every member in the society for each and every task or event

Society members contacts which have higher post

-

A active platform where companies wanting to sponser can come while cllgs can register event details to find sponsers.

All is well bro

What steps are you currently taking for solving problems you checked in the last question?

8 responses

Nothing

Getting in touch with other societies regarding sponsorships and performing for them to get deliverables

Hard Work, Dedication and Determination

Joined some groups that bring in sponsorships.

Take advice from seniors how to handle those problems

Working more on it with focused plan

Trying to make a common database for sponsers but not feasible as competition between fests and societies for such sponsers.

We just Chillin

FINAL REQUIREMENTS

☛ **Authentication System:**
- ✓ Sign Up Page
- ✓ Log in Page
- ✓ Google Authentication

☛ **Home Screen:**
- ✓ Event Library (Market Place)
- ✓ Home
- ✓ Videos
- ✓ Polls
- ✓ Posts
- ✓ Registration

☛ **User Profile:**
- ✓ Details
- ✓ Past Participation
- ✓ Badges
- ✓ Achievements

☛ **Explore:**
- ✓ Events
- ✓ Societies

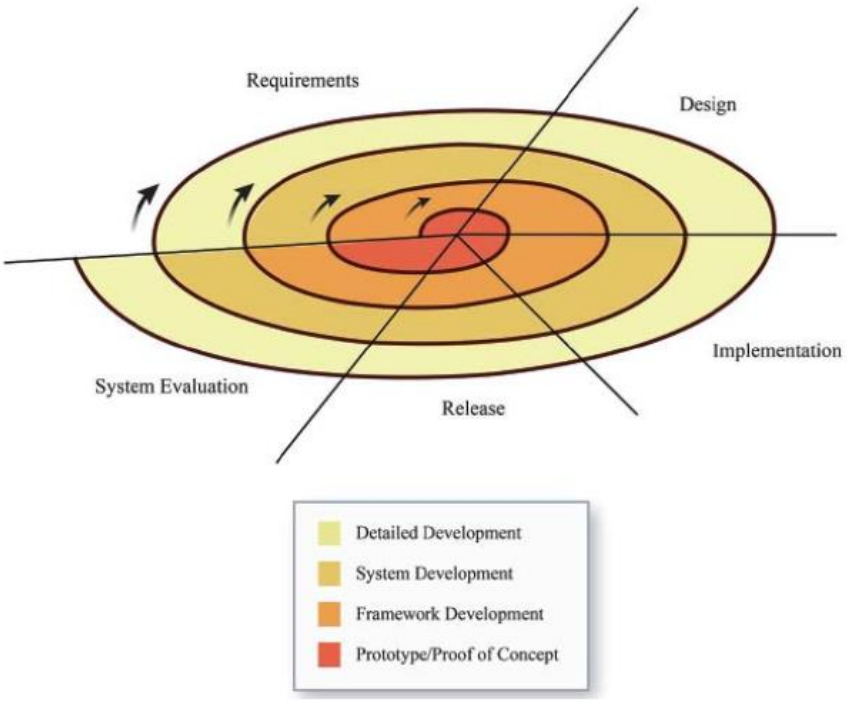
☛ **Society Management Dashboard:**
- ✓ Team
- ✓ Meet
- ✓ Deadlines
- ✓ Roadmap (Tentative)
- ✓ Get Merchandise
- ✓ Report Generation

☛ **Society - Sponsor service (Fundraising Model)**
- ✓ Mutual Marketing
- ✓ MoU

PROCESS MODEL

Spiral Model: It is a software development methodology that combines elements of both the iterative and the waterfall models. It is a risk-driven process model that provides a framework for developing software in a flexible, iterative, and incremental manner.

The spiral model is based on the idea of breaking down the development process into a series of smaller and more manageable stages, each of which involves a review and evaluation of the previous stage. The process proceeds in a spiral fashion, with each iteration building on the previous one, and incorporating feedback from users and stakeholders.



The key phases of the spiral model are:

Planning: This phase involves defining the objectives, requirements, and constraints of the project, as well as identifying potential risks and determining the scope of the project.

Risk analysis: In this phase, the potential risks and uncertainties associated with the project are identified and evaluated. This analysis helps to guide the development process and ensure that appropriate measures are taken to mitigate risks.

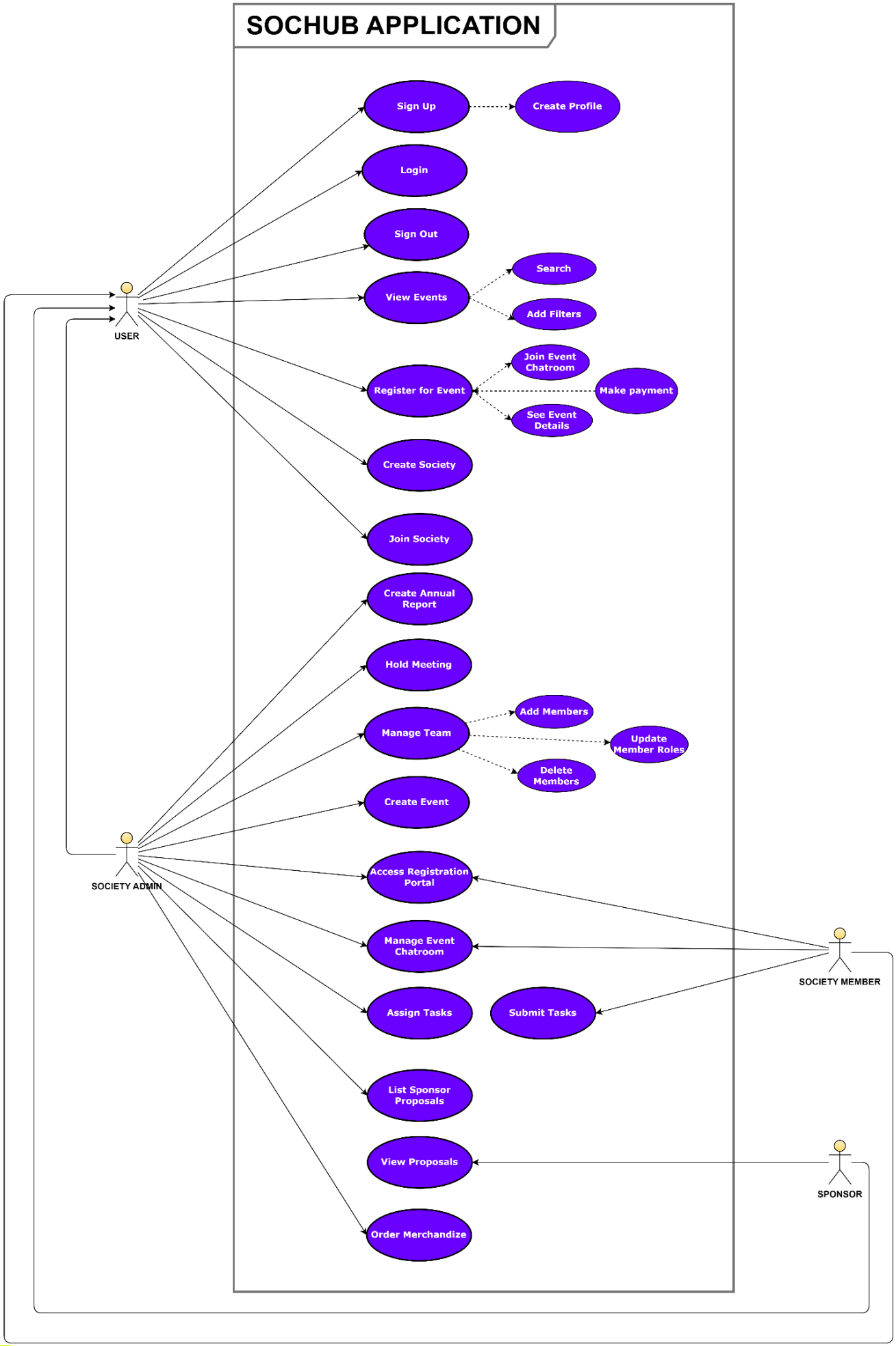
Engineering: This phase involves developing the software, testing it, and refining it based on feedback from users and stakeholders.

Evaluation: In this phase, the software is evaluated to ensure that it meets the requirements and objectives of the project. This evaluation may include testing, user feedback, and other forms of assessment.

The spiral model is particularly useful in complex software development projects where the requirements are uncertain or subject to change, and where risks and uncertainties are high. It provides a flexible and iterative approach to software development that allows developers to respond to changing requirements and feedback from users, while also ensuring that the software meets the objectives of the project.

USE CASE DIAGRAM

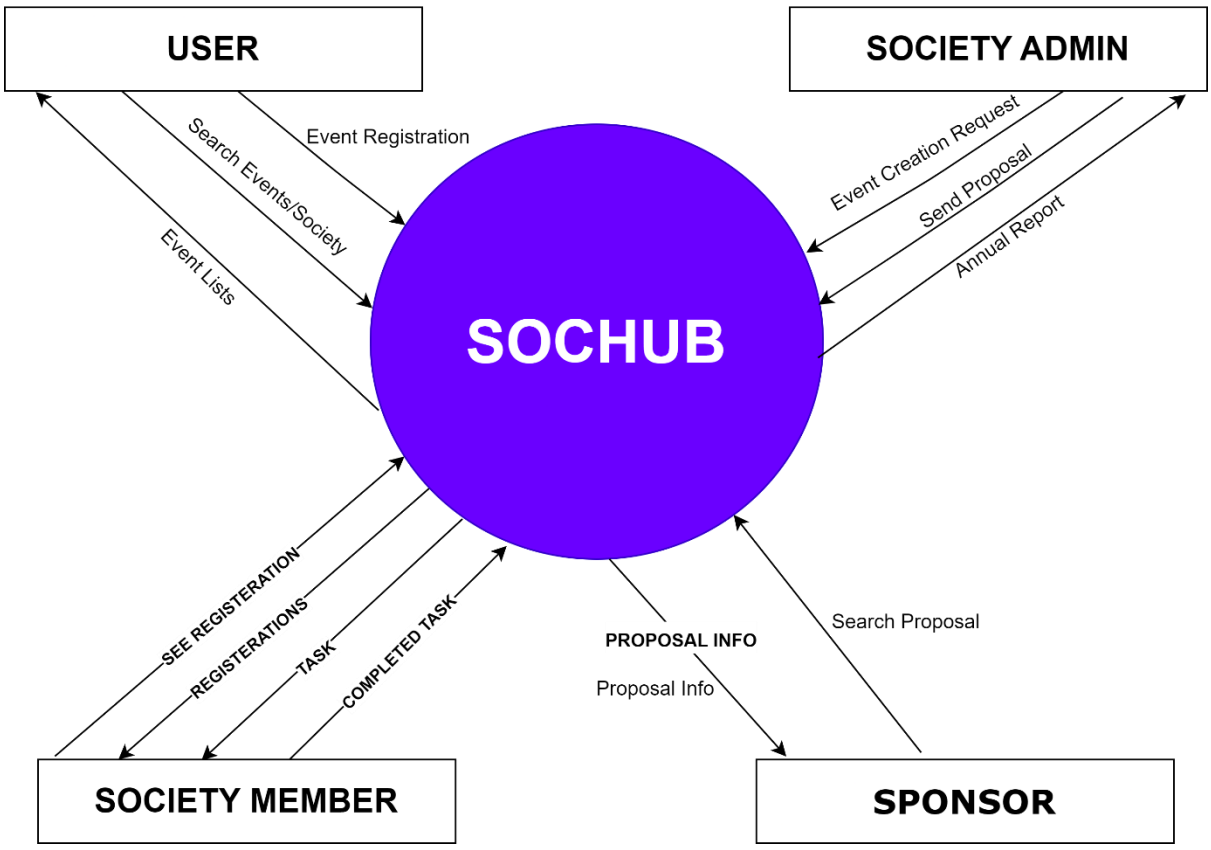
A use case diagram is a visual representation of the different ways in which users can interact with a system or software application, and the system's responses to those interactions.


SOCHUB APPLICATION
Sign Up
Create Profile
Login
Sign Out
View Events
Search
Add Filters
Register for Event
Join Event Chatroom
Make payment
See Event Details
Create Society
Join Society
Create Annual Report
Hold Meeting
Manage Team
Add Members
Update Member Roles
Delete Members
Create Event
Access Registration Portal
Manage Event Chatroom
Assign Tasks
Submit Tasks
List Sponsor Proposals
View Proposals
Order Merchandize
USER
SOCIETY ADMIN
SOCIETY MEMBER
SPONSOR

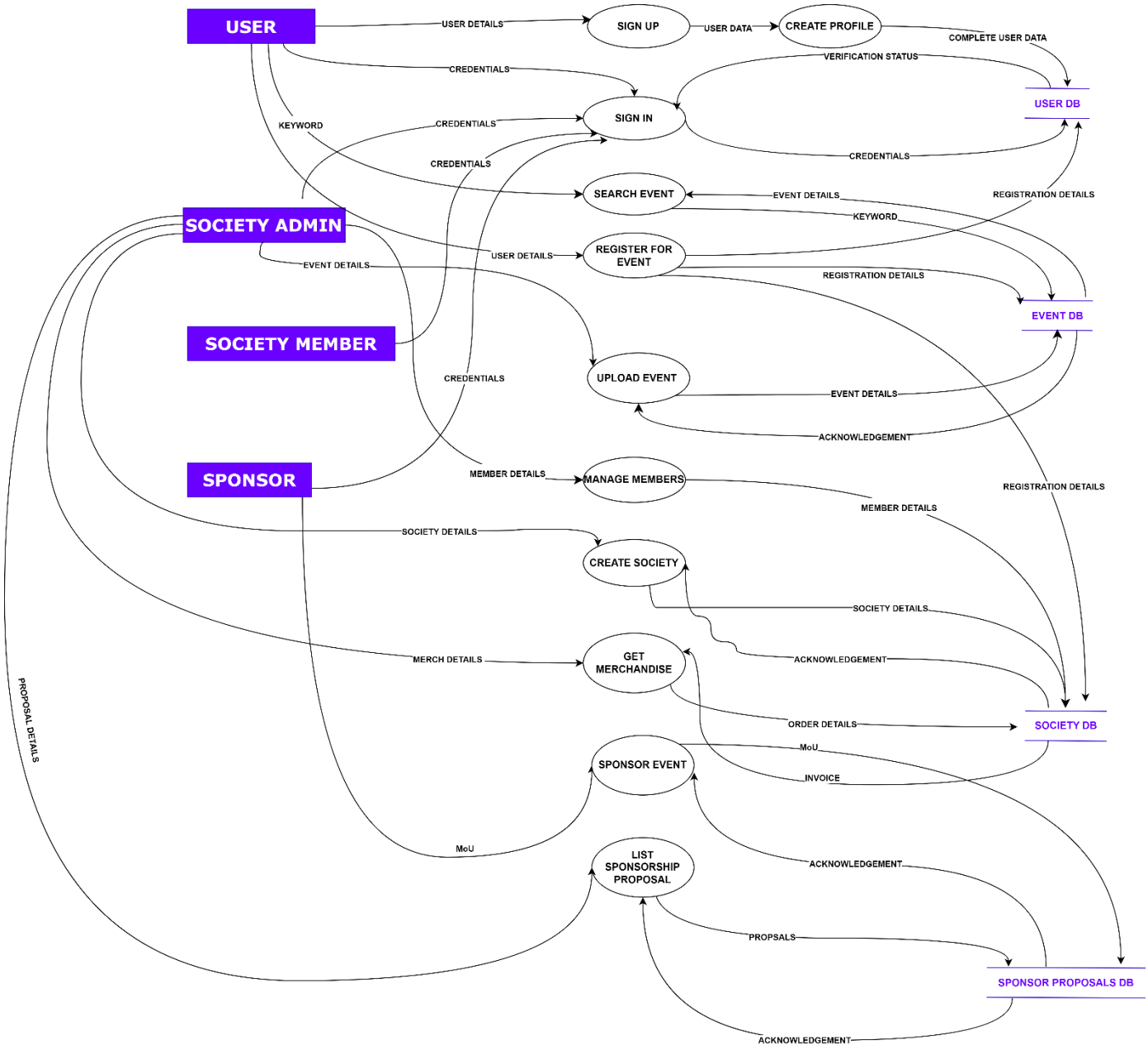
DATA FLOW DIAGRAM

Data flow diagrams (DFDs) are a visual modeling tools used in software engineering to represent data flow through a system. They depict the processes, data stores, and external entities involved in a system and the flow of data between them. DFDs are useful for understanding the data flow and relationships between different system components, and can be used in requirements gathering, system design, and testing. They are often used in conjunction with other modeling tools such as use case diagrams and sequence diagrams.

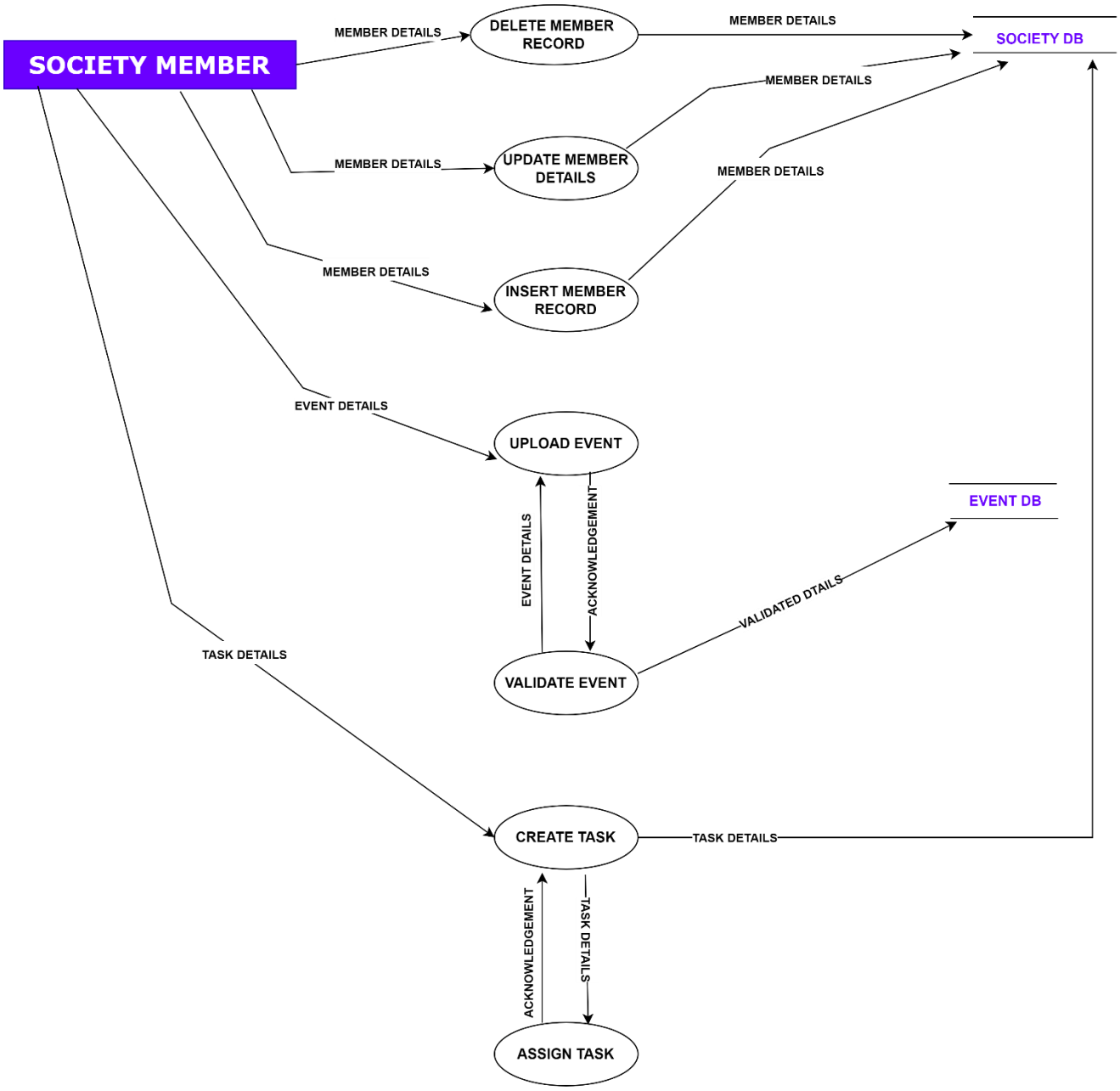
LEVEL ZERO DATA FLOW DIAGRAM



LEVEL ONE DATA FLOW DIAGRAM



LEVEL TWO DATA FLOW DIAGRAM



DATA DICTIONARY

Name of the variable	Data Type	Field Size (Length)	Description	Example
User ID	Integer	10	Unique identifier for each user	1234
Name	String	50	User's full name	“Jane Doe”
Email	String	50	User's email address	“jane.doe@example.com”
Password	String	50	User's password (encrypted)	“sEcReTpaSsWoRd”
College Name	String	50	User's college name	“XYZ College”
College Society Name	String	50	User's college society name (if applicable)	“ABC Society”
Event ID	Integer	10	Unique identifier for each event	5678
Event Name	String	50	Name of the event	“Spring Fling”
Event Date	Date	8	Date of the event	“2023-05-01”

Event Time	Time	5	Time of the event	“16:00”
Location	String	100	Location of the event	“Student Center, Room 201”
Description	Text	500	Description of the event	“Join us for a fun-filled afternoon of games and refreshments! ”
Registration Deadline	Date	8	Deadline for registration	“2023-04-25”
Society ID	Integer	10	Unique identifier for each society	9876
Society Name	String	50	Name of the society	“Chess Society”
Society Leader	String	50	Name of the society leader	“John Smith”
Society Description	Text	500	Description of the society	“We meet weekly to play chess and improve our skills.”
Society Members	Text	500	List of society members (comma-separated)	"Jane Doe, John Smith, Bob Johnson"

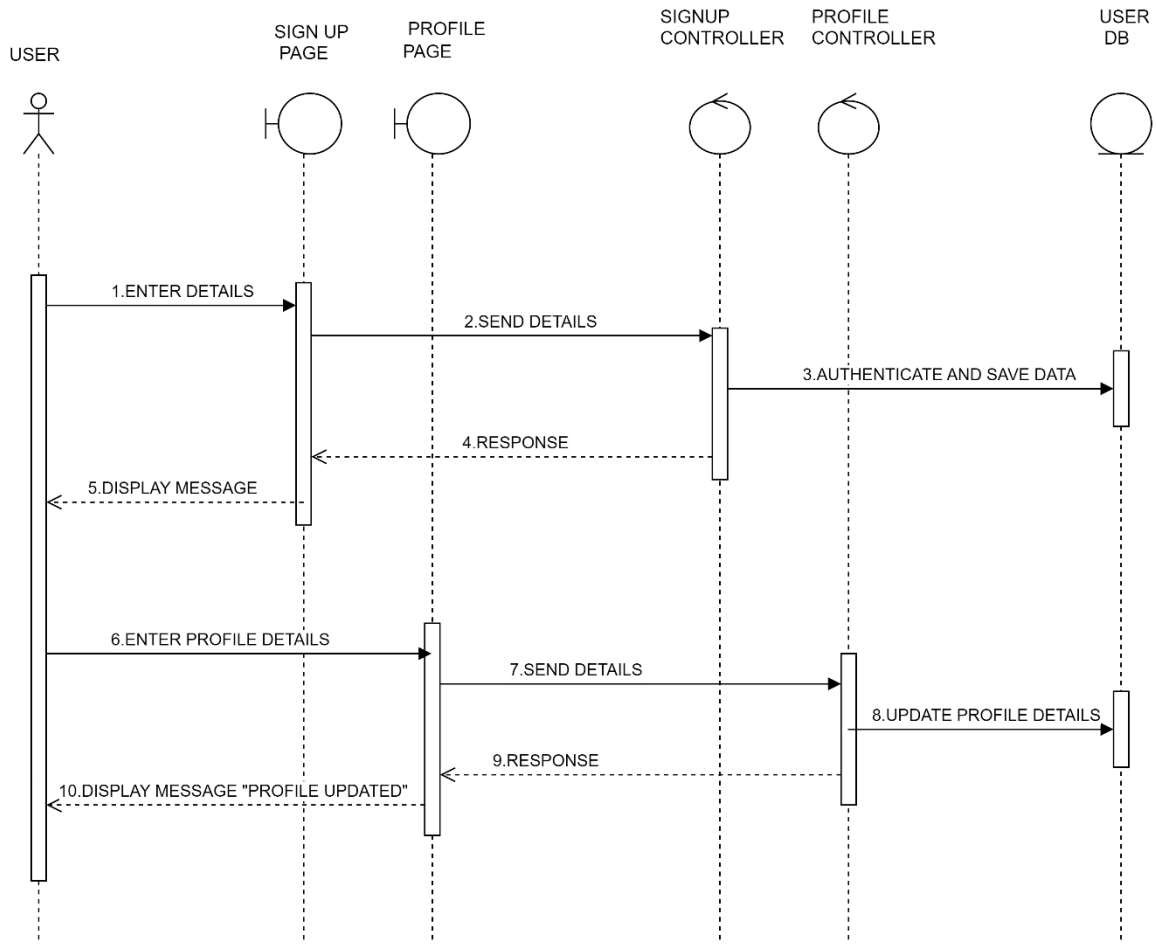
Task Name	String	50	Name of a society task	"Organize tournament"
Task Description	Text	500	Description of a society task	"Coordinate with other societies and reserve venue for tournament."
Task Points	Integer	10	Points earned for completing a society task	10
Society Report	Text	500	Annual report for the society	"During the past year, we hosted two tournaments and held weekly meetings to improve our skills."

SEQUENCE DIAGRAM

Sequence diagrams are a type of behaviour diagram in software engineering that depict the interactions between objects or actors in a particular scenario. They are useful for modelling complex scenarios and can help stakeholders to understand the behaviour of a system. They are commonly used in requirements gathering, system design, and testing, and can serve as a valuable reference for developers as they implement and test the system.

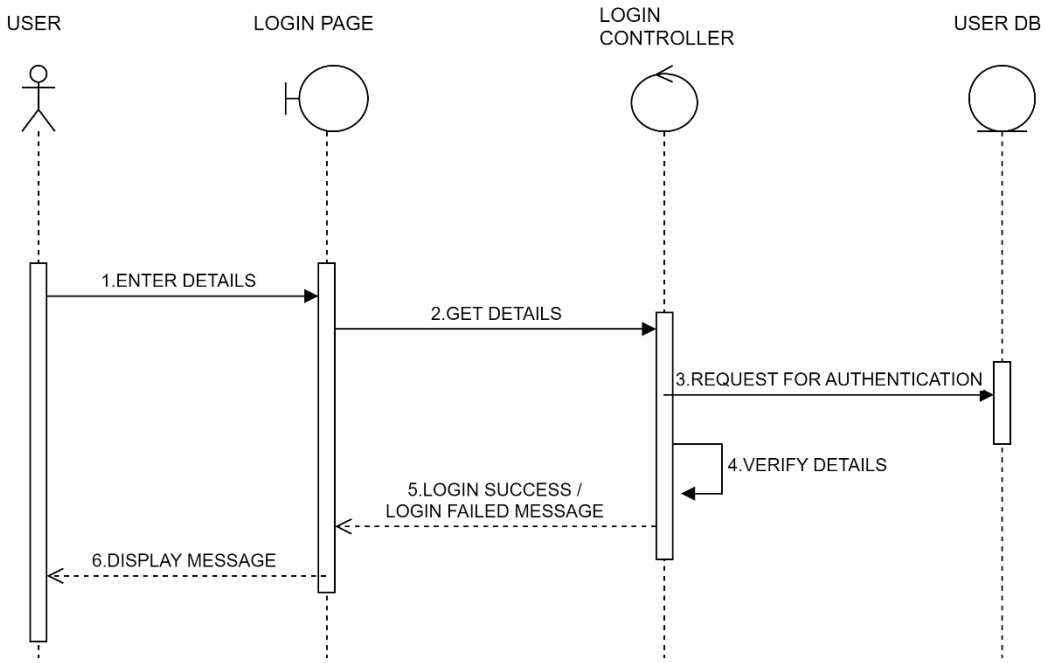
SEQUENCE DIAGRAM

FOR SIGNUP -> CREATE PROFILE



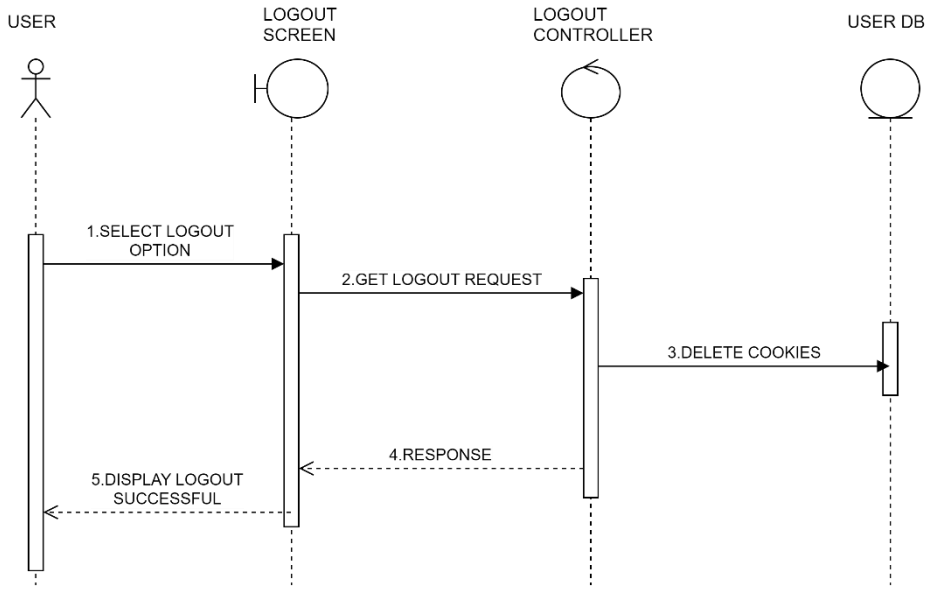
SEQUENCE DIAGRAM

FOR LOGIN



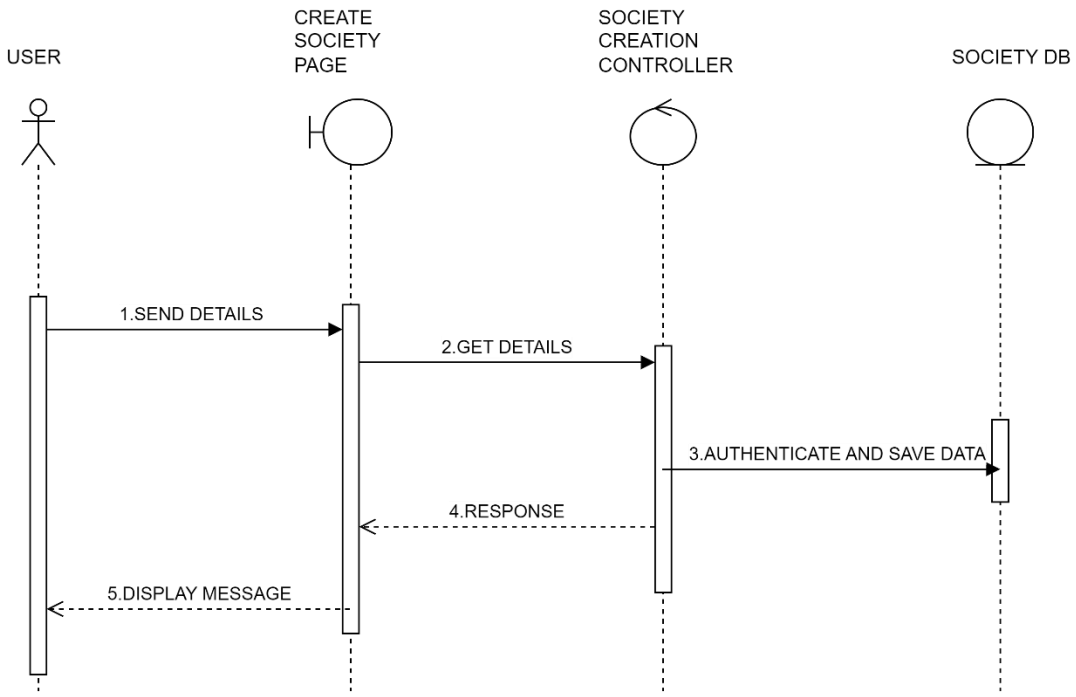
SEQUENCE DIAGRAM

FOR LOGOUT



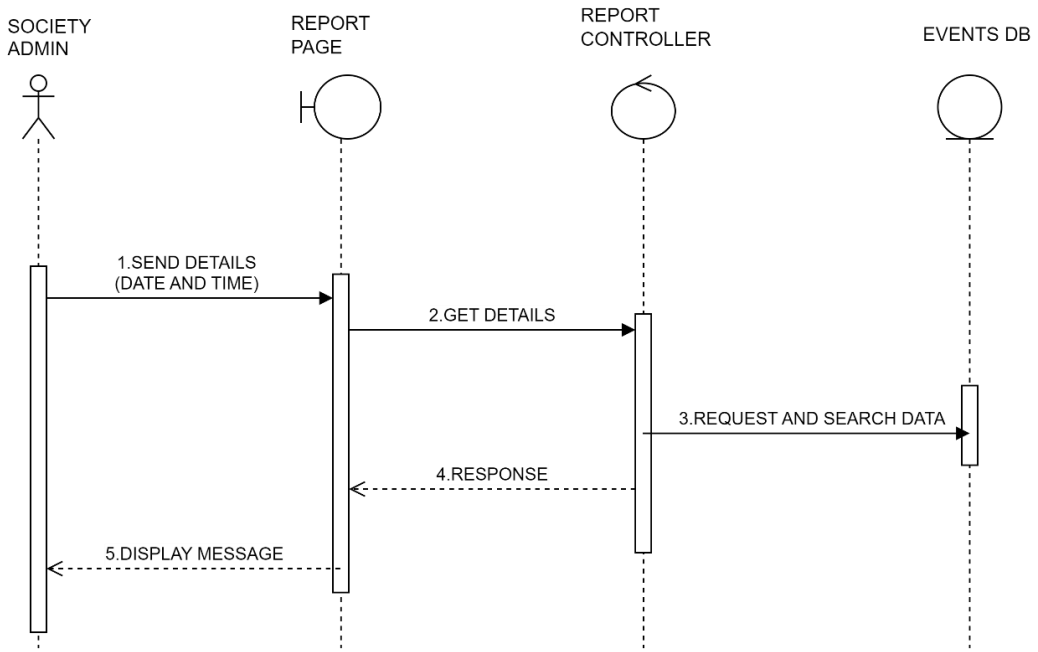
SEQUENCE DIAGRAM

CREATE SOCIETY



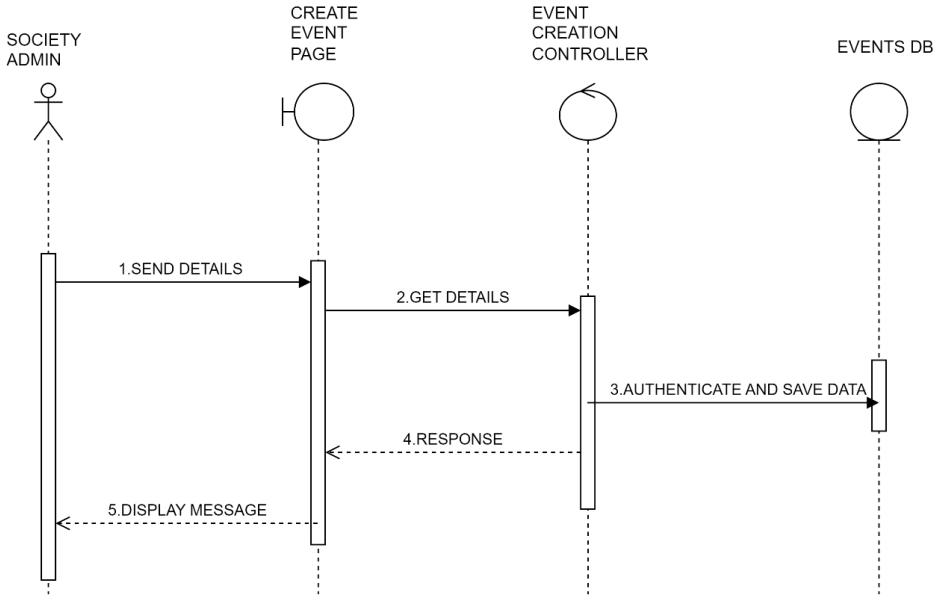
SEQUENCE DIAGRAM

ANNUAL REPORT



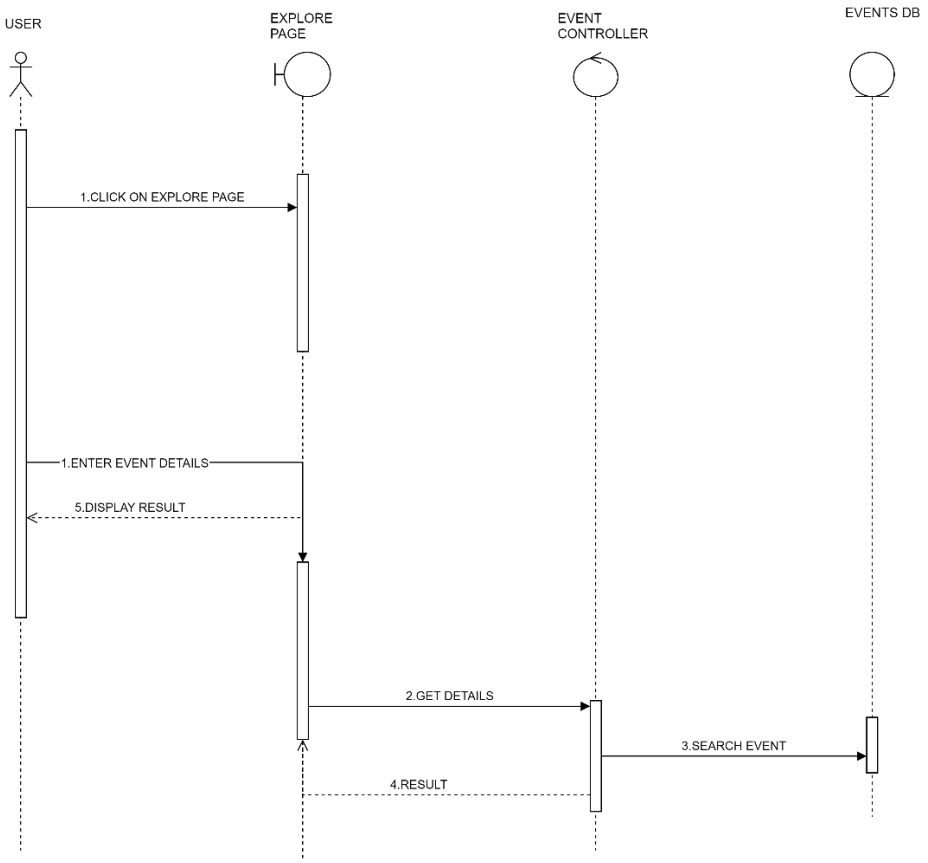
SEQUENCE DIAGRAM

CREATE EVENT



SEQUENCE DIAGRAM

FOR VIEW EVENT --> SEARCH EVENT & FILTER EVENTS



TIMELINE CHART

Timeline charts show the duration of project tasks or activities, their beginning and ending dates, the interval between them, whether any tasks overlap, and, if any, their interdependencies. Additionally, it's a fantastic tool for summarising and emphasizing your work in reports for stakeholders.

S. No.	Process/ Phase	Start Date	End Date
1	Problem Statement	06.01.23	14.01.23
2	Initial Requirement	15.01.23	21.01.23
3	Requirement Analysis	22.01.23	29.01.23
4	Final Requirement	30.01.23	06.02.23
5	Process Model	07.02.23	13.02.23
6	Use Case Diagram	14.02.23	20.02.23
7	Data Flow Diagram	21.02.23	27.02.23
8	Sequence Diagram	28.02.23	06.03.23
9	Function Point Analysis	07.03.23	15.03.23
10	UI Design	16.03.23	22.03.23
11	Data Design	23.03.23	31.03.23
12	Pseudo code	01.04.23	14.04.23
13	White Box Testing	15.04.23	21.04.23
14	Implementation	22.04.23	29.04.23

FUNCTION POINT ANALYSIS

FPA is a standard metric for the relative size and complexity of a software system, originally developed by Alan Albrecht of IBM in the late 1970s. Function Points (FPs) can be used to estimate the relative size and complexity of software in the early stages of development- analysis and design.

It assesses the functionality delivered to its users, based on the user's external view of the functional requirements. It measures the logical view of an application not the physically implemented view or the internal technical view.

The FPA technique is used to analyze the functionality delivered by software and Unadjusted Function Point (UFP) is the unit of measurement.

OBJECTIVES OF FPA

- The objective of FPA is to measure the functionality that the user requests and receives.
- The objective of FPA is to measure software development and maintenance independently of the technology used for implementation.

☛ It should be simple enough to minimize the overhead of the measurement process.

☛ It should be a consistent measure among various projects and organizations.

TYPES OF FPA

☛ Transactional Functional Type –

- ✓ **External Input (EI):** EI processes data or control information that comes from outside the application's boundary. The EI is an elementary process.
- ✓ **External Output (EO):** EO is an elementary process that generates data or control information sent outside the application's boundary.
- ✓ **External Inquiries (EQ):** EQ is an elementary process made up of an input-output combination that results in data retrieval.

☛ Data Functional Type –

- ✓ **Internal Logical File (ILF):** A user-identifiable group of logically related data or control information maintained within the boundary of the application.
- ✓ **External Interface File (EIF):** A group of user-recognizable logically related data allusion to the software but maintained within the boundary of another software.

BENEFITS OF FPA

- FPA is a tool to determine the size of a purchased application package by counting all the functions included in the package.
- It is a tool to help users discover the benefit of an application package to their organization by counting functions that specifically match their requirements.
- It is a tool to measure the units of a software product to support quality and productivity analysis.

STEPS FOR CALCULATING FPA

STEP – 1 : Calculating Adjustment Factor

S No.	Parameters	Adjustment Factor
1.	Does the system require reliable backup and recovery?	f1=3
2.	Are specialized data communications required to transfer information to or from application?	f2=4
3.	Are there distributed processing functions?	f3=3
4.	Is performance critical?	f4=4
5.	"Will the system run in an existing, heavily utilized operational Environment?"	f5=4
6.	Does the on-line data entry require the input transaction to be built over multiple screens or operations?	f6=3

7.	Does the system require online data entry?	f7=4
8.	Is the code designed to be reusable?	f8=3
9.	Is the system designed for multiple installations in different organizations?	f9=3
10.	"Are the inputs, outputs, inquiries complex?"	"f10=1
11.	Are the Internal Logical Files updated online?	f11=2
12.	Is the internal processing complex?	f12=3
13.	Are conversion and installation included in the design?	f13=1
14.	Is the application designed to facilitate change and ease of use by the user?	f13=1

Scale varies from 0 to 5 according to character of Complexity Adjustment Factor (CAF). Below table shows scale:

0 = No influence

1 = Incidental

2 = Moderate

3 = Average

4 = Significant

5 = Essential

Formula for calculating Complexity Adjustment Factor (CAF):

$$CAF=0.65 + 0.01 * \Sigma fi$$

Calculating ∑fi

∑fi = f1 + f2 + f3 +f4 + f5 + f6 + f7 + f8 + f9 + f10 + f11 + f12 + f13 + f14

= 3 + 4 + 3 + 4 + 4 + 3 + 4 + 3 + 3 + 1 + 2 + 3 + 1 + 4

= 42

Calculating CAF

CAF = 0.65 + 0.01 * ∑fi

= 0.65 + 0.01*42

= 0.65 + 0.42

= 1.07

☛ STEP 2: UNADJUSTED FUNCTION POINT (UFP)

EXTERNAL INPUTS:

Input Name	Fields
Sign In	2
Sign Up	3
Search event	1
Location	1

EXTERNAL OUTPUTS:

Output Name	Fields
Event List	1
Event Details	1
Notifications	1

INTERNAL LOGICAL FILES:

ILF	Fields
View Events	1
Search Events	1
Register Events	1
Join Society	1
View Roadmap	2

EXTERNAL LOGICAL FILES:

ELF	Fields
Order merchandise	1
Hold meeting	2
Sign in using google	4

☛ TABLE FOR UNADJUSTED FUNCTION POINT (UFP):

Calculation:

The weight factor is assumed to be average.

Function Type	Estimated Factor	Weight factor			Function type total
		Low	Average	High	
EI	4	3	4	6	16
EO	3	4	5	7	15
EQ	0	3	4	6	0
ILF	5	7	10	15	50
ELF	3	5	7	10	21

Total Unadjusted Function Point Count =16+15+0+50+21 = 102

Calculating Function Point Count

AFP = UFP × CAF

AFP = 102 × 1.07

AFP = 109.14

Where, AFP = Adjusted Function Point , UFP = Unadjusted Function Point, CAF = Complexity Adjustment Factor

RISK MANAGEMENT

Many different risks may be present in a software project. It is important to categorise risks into various groups in order to be proficient at systematically identifying the key risks that may have an impact on a software project. Following that, the project manager can determine which risks from each class apply to the project.

There are three major categories of hazards that can have an impact on a software project:

- Project Risks
- Business Risks
- Technical Risks

- **Project Risks-** Project risks include several types of financial, logistical, human resource, and customer-related issues. Slippage in the schedule is a significant project risk. It is extremely difficult to monitor and supervise a software project because the software is intangible.
- **Technical risks-** Technical risks relate to conceivable implementation, interface, testing, and maintenance problems. In addition, it has vague, imperfect, and changing specifications as well

as technical uncertainty and obsolescence. The majority of technical hazards arise from the development team's incomplete understanding of the project.

☛ **Business risks:** These risks frequently endanger the project or the final product and pose a threat to the viability of the software that needs to be developed. the following list of the top five business risks:

- ✓ Building a fantastic system or product that no one truly wants is a **market risk**.
- ✓ Building a product that no longer works with the company's broader business strategy is a **strategic risk**.
- ✓ Building a product that the sales force is unable to market is a **sales risk.**
- ✓ **Management risk** is losing the support of senior management due to a change in focus or a change in people.
- ✓ **Budget risk** is losing budgetary or personnel commitment.

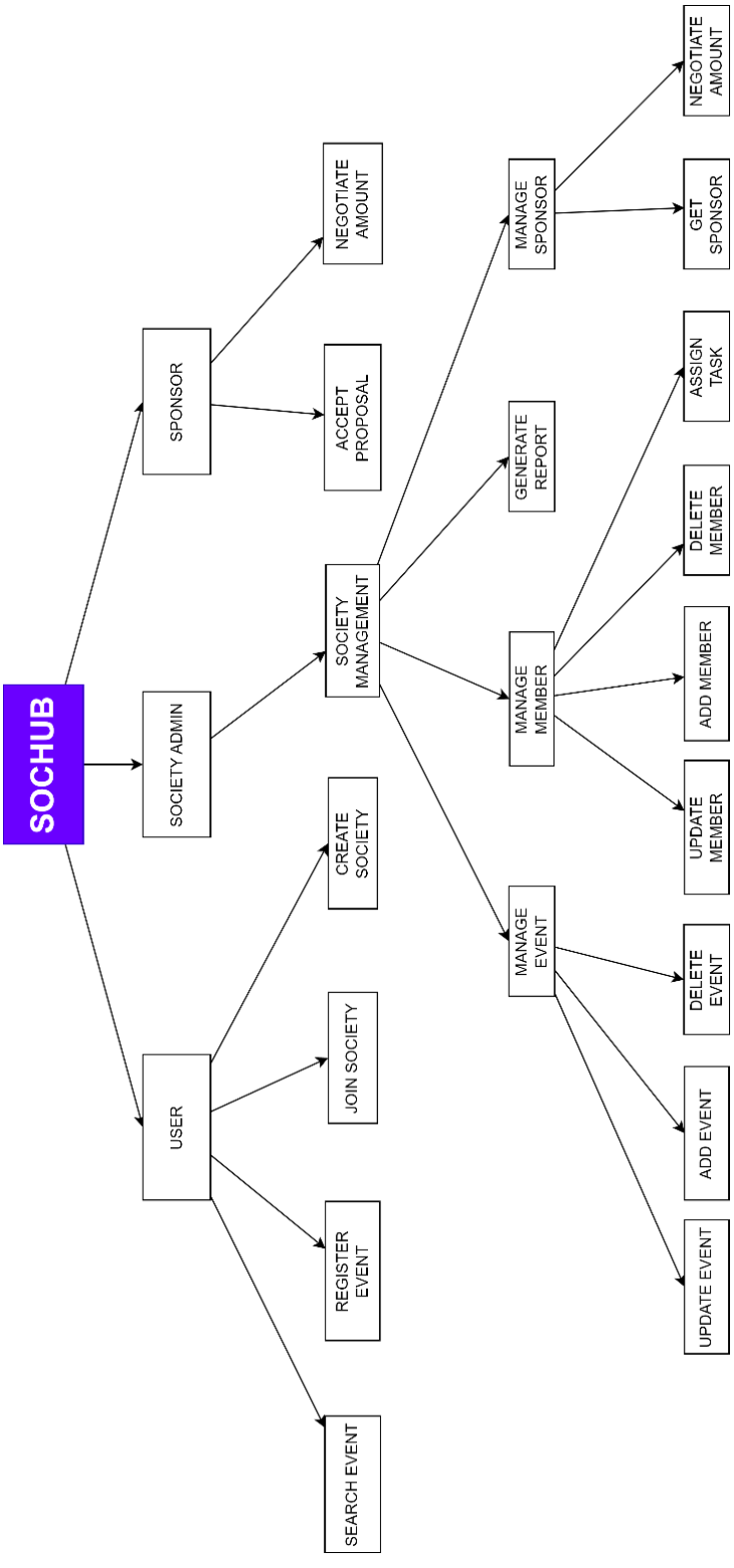
Risk Category	Description of Risk	Probability	Impact	RMMM
Technical	System downtime	Moderate	High	"Regular system maintenance and backup, fault-tolerant design, load balancing, and disaster recovery planning."
	Data breaches	Moderate	High	"Regular security audits and patches, strong access controls,

				encryption, and employee training on security protocols."
	Integration issues	Low	Moderate	"Thorough testing and validation during integration, using industry-standard protocols and APIs, and maintaining close collaboration with third-party providers."
Business	Market competition	High	Moderate	"Continuous market research, product innovation, targeted marketing, and user feedback analysis."
	Revenue fluctuations	Moderate	High	"Diversifying revenue streams, maintaining efficient cost structures, and seeking new partnerships and investment opportunities."
	Regulatory compliance	High	High	"Regular monitoring of

				legal and regulatory changes, compliance training for employees, maintaining detailed records, and seeking legal counsel as needed."
Operational	Human errors	Moderate	Moderate	"Robust training and support programs for employees, strong standard operating procedures, and regular performance reviews."
	Communication breakdowns	Low	Moderate	"Regular team meetings, open channels of communication, and collaboration tools such as project management software."
	System scalability	Low	High	"Continuous system monitoring and capacity planning, efficient resource allocation, and

				upgrading hardware and software as needed."
Environmental	Natural disasters	Low	High	"Disaster recovery planning, regular testing of backup systems, and geographically redundant infrastructure."
	Political instability	Low	Moderate	"Monitoring of geopolitical developments, maintaining multiple hosting locations, and having contingency plans in place."

ARCHITECTURAL DESIGN



DATA DESIGN

EVENTS DATABASE

1	"BUG BYTE LEAGUE"	"ACE"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	15000	49
2	"THINK TANK"	"ACE"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	5000	29
3	"TANGELD TAGS"	"ACE"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	6000	49
4	"E LAFDA"	"WEBSTERS"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	7000	69
5	"WEB HIVE"	"WEBSTERS"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	15000	49
6	"DARK CODING"	"WEBSTERS"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	20000	29
7	"KREATIV"	"CYBERNAUTS"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	5000	49
8	"ALGORHTHM"	"CYBERNAUTS"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	5000	49
10	"DYNAMIX"	"CYBERNAUTS"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	5000	49
11	"MINISTRY OF MAGIC"	"BLITZ"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	7000	49
12	"AI TANK"	"BLITZ"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	5000	49
13	"STREAM DRAW"	"BLITZ"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	5000	49
14	"CREEPY CRYPT HUNT"	"BLITZ"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	6000	29
15	"E TAMBOLA"	"PARIKALAN"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	5000	49
16	"TREASURE HUNT"	"PARIKALAN"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	7000	49
17	"QUIZ KHALIFA"	"PARIKALAN"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	3000	29
19	"CODE ESCAPE"	"CURIEUX"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	7000	49
20	"CAN YOU BEAT THE AI"	"CURIEUX"	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	10000	49

SOCIETY DATABASE

	_id ObjectId	Societylogo String	SocietyName String
1	ObjectId('64555a28473371cdae4…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"ACE"
2	ObjectId('64555a28473371cdae4…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"WEBSTERS"
3	ObjectId('64555a28473371cdae4…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"CYBERNAUTS"
4	ObjectId('64555a28473371cdae4…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"BLITZ"
5	ObjectId('64555a28473371cdae4…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"PARIKALAN"
6	ObjectId('64555a28473371cdae4…	"/9j/4AAQSkZJRgABAQAAAQABAAD/…	"DEVCOM"

IMPLEMENTATION

MAIN

```
import 'package:flutter/material.dart';
import 'package:sochub/pages/home.dart';
import 'package:sochub/pages/loading.dart';

void main() => runApp(MaterialApp(
  theme: ThemeData(
    useMaterial3: true,
  ),
  routes: {
    '/': (context) => const Loading(),
    '/home': (context) => const SocHub(),
  },
));
```

PAGES

✓ home.dart

```
import 'package:flutter/material.dart';
import 'package:sochub/widgets/feed_video.dart';
import 'package:sochub/widgets/semi_bold_text.dart';
import '../widgets/profile_update.dart';

class SocHub extends StatefulWidget {
  const SocHub({Key? key}) : super(key: key);

  @override
  State<SocHub> createState() => _SocHubState();
}

class _SocHubState extends State<SocHub> {
```

```
  List<List<Map<String, dynamic>>> _data = [];

  @override
  Widget build(BuildContext context) {
    _data = ModalRoute.of(context)?.settings.arguments as
List<List<Map<String, dynamic>>>;

    return Scaffold(
      backgroundColor: const Color(0xFF2F2F2F),

      appBar: AppBar(
        title: SemiBoldText(text: "SocHub",color: Colors.white,
size: 24,),
        backgroundColor: const Color(0xFF222222),
        scrolledUnderElevation: 0,
        leading: Builder(
          builder: (BuildContext context) {
            return IconButton(
              icon: const Icon(
                Icons.menu_rounded,
                color: Colors.grey,
                size: 48,
              ),
              onPressed: () { Scaffold.of(context).openDrawer();
},
              tooltip:
MaterialLocalizations.of(context).openAppDrawerTooltip,
            );
          },
        ),
      ),

      body: ListView(
          // physics: const BouncingScrollPhysics(),

          children: [

            Container(
              height: 146,
              clipBehavior: Clip.antiAlias,
              margin: const EdgeInsets.only(bottom: 8,),
              decoration: const BoxDecoration(
                color: Colors.pink,
                borderRadius: BorderRadius.only(
                    bottomLeft: Radius.circular(50.0),
                    bottomRight: Radius.circular(50.0)),
                boxShadow: [
                  BoxShadow(
```

```
                    color: Color(0x2affffff),
                    blurRadius: 100,
                  ),
                ],
                gradient: LinearGradient(
                    begin: Alignment.center,
                    end: Alignment.bottomCenter,
                    colors: [
                      Color(0xFF222222),
                      Color(0xFF212121),
                    ],
                ),
              ),
              child: ListView.separated(
                padding: const EdgeInsets.all(21.0),
                separatorBuilder: (context, index) => const
SizedBox(
                  width: 21,
                ),
                scrollDirection: Axis.horizontal,
                // physics: const BouncingScrollPhysics(),
                itemCount: _data[0].length,
                itemBuilder: (_,index){
                  return Row(
                    children: [
                      ProfileUpdate(
                          text: _data[0][index]['SocietyName'],
                          image: _data[0][index]['Societylogo']
                      ),
                    ],
                  );
                  },
              ),
            ),
            Column(
              children: List.generate(_data[1].length, (index) =>
FeedVideo(
                    eventName: _data[1][index]['Eventname'],
                    organizerName: _data[1][index]['OrganizerName'],
                    organizerLogo: _data[1][index]['Organizerlogo'],
                    image: _data[1][index]['image'],
                    prizePool: _data[1][index]['Prize'],
                    registrationPrice:
_data[1][index]['RegistrationPrice'],
              )
              ),
            )
        ]
```

```
      ),
      bottomNavigationBar: BottomNavigationBar(
        backgroundColor: Colors.black,
        selectedItemColor: const Color(0xffCCFF00),
        unselectedItemColor: const Color(0xffCCFF00),
        enableFeedback: true,
        showSelectedLabels: false,
        showUnselectedLabels: false,
        items:  const <BottomNavigationBarItem>[
          BottomNavigationBarItem(
            icon: Icon(Icons.home),
            label: "Home",
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.explore_outlined),
            label: "Explore",
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.account_circle_outlined),
            label: "Profile",
          ),
        ],
      ),
    );
  }
}
```

✓ loading.dart

```
import 'package:flutter/material.dart';
import 'package:flutter_spinkit/flutter_spinkit.dart';
import 'package:sochub/widgets/app_text.dart';
import '../services/data_services.dart';

class Loading extends StatefulWidget {
  const Loading({super.key});

  @override
  State<Loading> createState() => _LoadingState();
}

class _LoadingState extends State<Loading> {
  @override
  void initState() {
    super.initState();
    // Fetch data from MongoDB when the app starts
    _fetchData();
```

```
  }

  Future<void> _fetchData() async {
    if (!mounted) return;
    Navigator.pushReplacementNamed(context, '/home', arguments:
await DataService.fetchData());
  }


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        color: Colors.black,
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: const [
            SpinKitRipple(
              color: Color(0xffCCFF00),
              size: 200.0,
              borderWidth: 30,
              // duration: Duration(milliseconds: 500),
            )
          ],
        ),
      ),
    );
  }
}
```

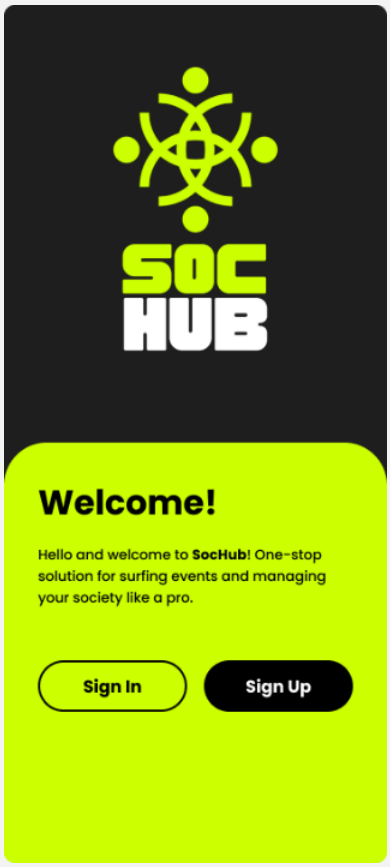
SERVICES

✓ data_services.dart

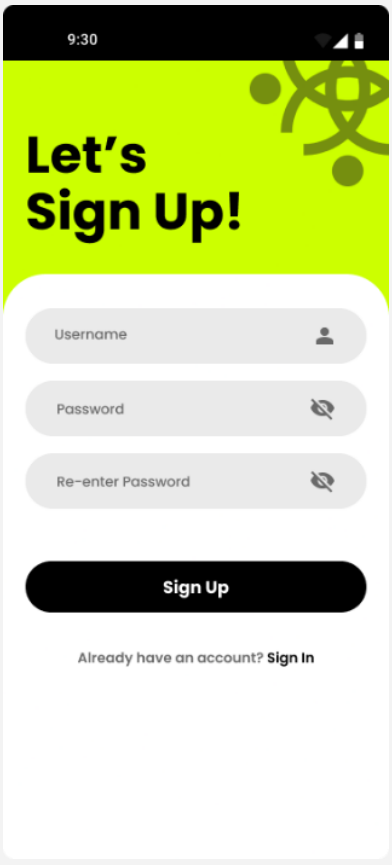
```
import 'package:mongo_dart/mongo_dart.dart';

class DataService {
  static Future<List<List<Map<String, dynamic>>>> fetchData()
async {
    try {
      // Connect to the MongoDB database
```

```
      var db = await
Db.create('mongodb+srv://varunsangai11:varun%40123@cluster0.kbsesu
f.mongodb.net/sochub?retryWrites=true&w=majority');
      await db.open();

      // Get a reference to the collection you want to query
      var collection1 = db.collection('society');
      var collection2 = db.collection('events');

      // Query the collection for documents that match your
criteria
      // var query = where.eq('field', 'value');
      var cursor1 = await collection1.find().toList();
      var cursor2 = await collection2.find().toList();

      // Convert the cursor results to a List<Map<String,
dynamic>> and return it
      var data1 = cursor1.map((document) => document).toList();
      var data2 = cursor2.map((document) => document).toList();

      // Close the database connection
      await db.close();

      return [data1,data2];
    } catch (e) {
      // Handle the ConnectionException error
      if (e is ConnectionException) {
        print('Unable to connect to the MongoDB database');
      } else {
        print('An error occurred: $e');
      }

      return [];
    }
  }
}
```

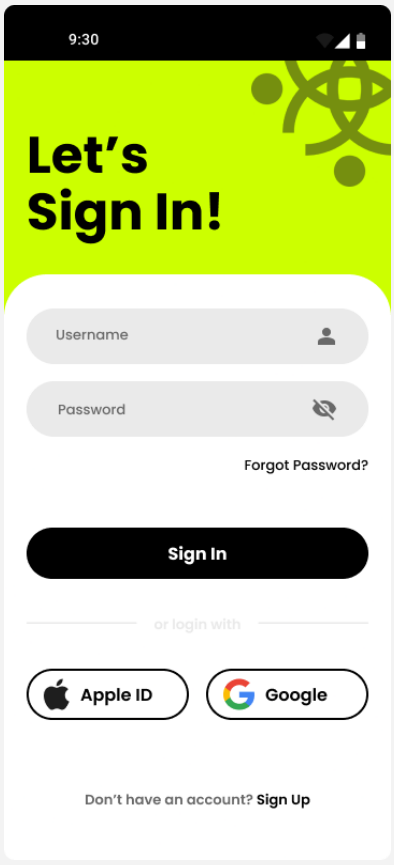
INTERFACE DESIGN



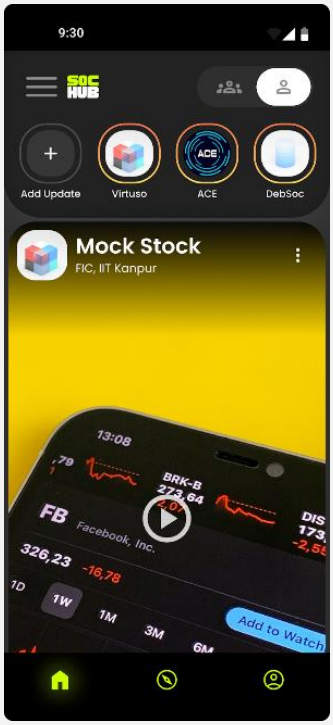
WELCOME SCREEN



SIGN UP SCREEN



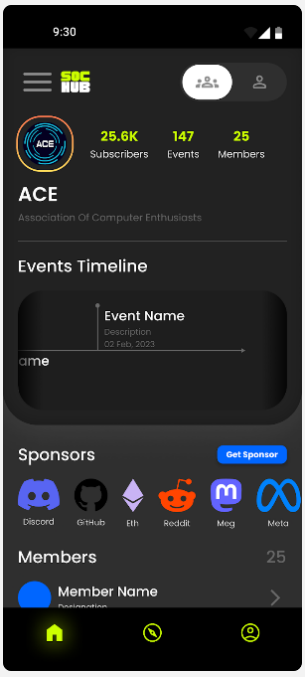
SIGN IN SCREEN



HOME - USER

EXPLORE PAGE

USER PROFILE



HOME - SOCIETY

PSUEDOCODE

```
FUNCTION register_event(event_id, uid):
    IF event_exists(event_id) THEN
        IF NOT is_user_registered(event_id, uid) THEN
            user_details ← get_user_details(uid)
            add_registration(event_id, user_details)
            PRINT "You have successfully registered for the
event!"
        ELSE
            PRINT "You are already registered for this event."
        END IF
    ELSE
        PRINT "The event does not exist."
    END IF
END FUNCTION

FUNCTION event_exists(event_id):
    RETURN event_id IN events_db
END FUNCTION

FUNCTION is_user_registered(event_id, uid):
    FOR each registration in registrations_database[event_id] DO
        IF registration['uid'] == uid THEN
            RETURN True
        END IF
    END FOR
    RETURN False
END FUNCTION

FUNCTION get_user_details(uid):
    user_details ← {'uid':uid}
    WHILE True DO
        em = INPUT "Enter your email: "
        IF valid_email(em) THEN
            user_details['email'] ← em
            EXIT WHILE
        ELSE
```

```
            PRINT "Please Enter Valid Email"
        END IF
    END WHILE

    user_details['name'] ← INPUT "Enter your name: "

    WHILE True DO
        TRY
            age = INT(INPUT "Enter your age: ")
            IF age == 0 THEN
                RAISE
            END IF

            user_details['age'] ← age

            EXIT WHILE
        EXCEPT
            PRINT 'Please enter valid age'
            PRINT "Please Enter Valid Age"
        END TRY
    END WHILE

    user_details['institution'] ← INPUT "Enter your institution
name (if any): "

    user_details['phone'] ← INPUT "Enter your phone number: "

    user_details['course'] ← INPUT "Enter your course (if any): "

    user_details['team_size'] ← INPUT "Enter your team size (if
any): "

    user_details['country'] ← INPUT "Enter your country of
origin: "
    RETURN user_details
END FUNCTION

FUNCTION add_registration(event_id, user_details):
    APPEND user_details TO registrations_database[event_id]
END FUNCTION

FUNCTION valid_email(email):
    RETURN ("@" IN email) AND ("." IN email)
END FUNCTION
```

TESTING

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code.

Testing Approaches:

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

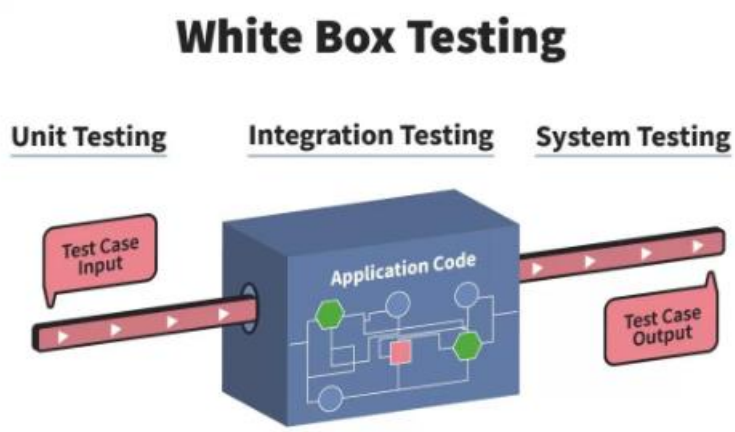
When functionality is being tested without taking the actual implementation in concern it is known as black-box testing. The other side is known as white-box testing where not only functionality is tested but the way it is implemented is also analysed.

We have implemented white-box testing in our project.

WHITE-BOX TESTING

White box testing is a software testing technique that involves examining the internal workings of an application or system being tested. It is also known as clear box testing or structural testing.

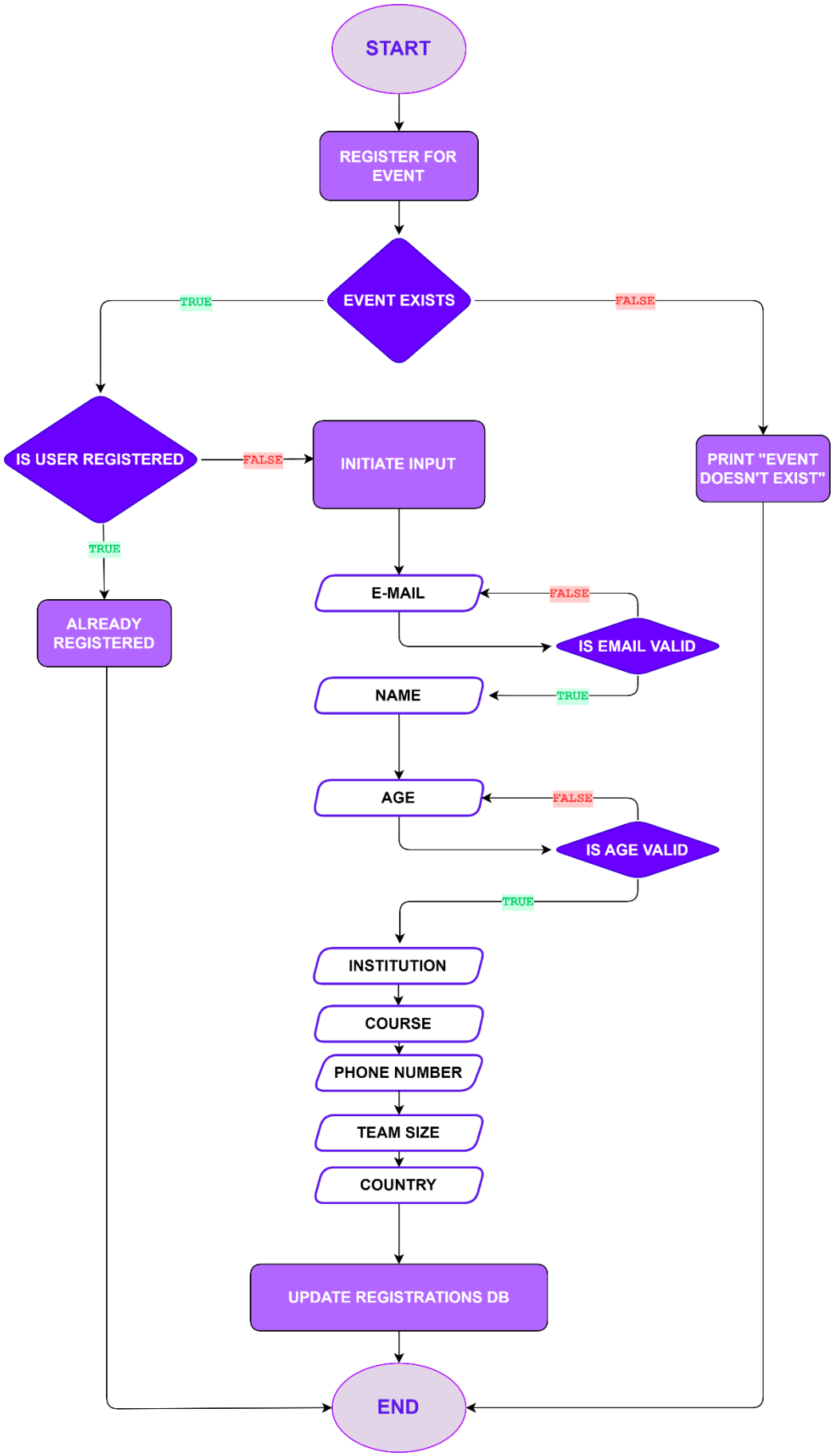
During white box testing, the tester has access to the source code of the software being tested and uses this information to design and execute test cases. This approach allows for more in-depth testing of the software's functionality and ensures that all possible paths through the code are tested.



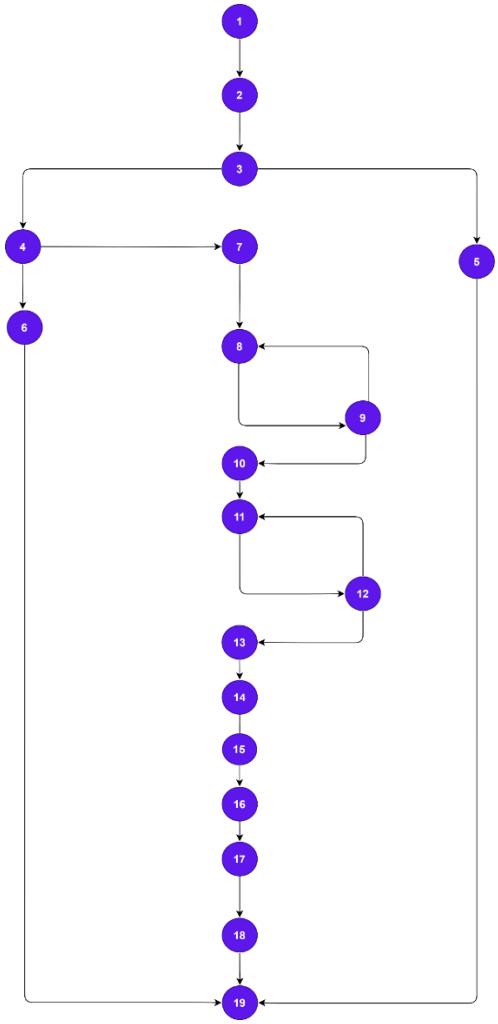
White box testing is typically performed by developers, or by specialized testers who have a strong understanding of programming languages and code structures. Its objective is to identify defects and errors in the code, as well as to verify that the software is performing as expected and meeting the design and functional requirements.

Examples of white box testing techniques include unit testing, integration testing, and code coverage analysis

FLOWCHART


START
REGISTER FOR EVENT
EVENT EXISTS
TRUE
FALSE
IS USER REGISTERED
FALSE
INITIATE INPUT
PRINT "EVENT DOESN'T EXIST"
TRUE
ALREADY REGISTERED
E-MAIL
FALSE
IS EMAIL VALID
NAME
TRUE
AGE
FALSE
IS AGE VALID
TRUE
INSTITUTION
COURSE
PHONE NUMBER
TEAM SIZE
COUNTRY
UPDATE REGISTRATIONS DB
END

FLOWGRAPH



CYCLOMATIC COMPLEXITY

INDEPENDENT PATHS:

i. 1→2→3→4→6→19

ii. 1→2→3→5→19

iii. 1→2→3→4→7→8→9→10→11→12→13→14→15→16→17→18→19

iv. 1→2→3→4→7→8→9→8→9→10→11→12→13→14→15→16→17→18→19

v. 1→2→3→4→7→8→9→10→11→12→11→12→13→14→15→16→17→18→19

The cyclomatic complexity is a measure of the number of **linearly independent paths** through a program's source code. It can be calculated using the following formula:

$$M = E - N + 2P$$

Where:

- **M** = cyclomatic complexity

- **E** = number of edges in the flow graph

- **N** = number of nodes in the flow graph

- **P** = number of connected components

From the flowchart and the flowgraph, we can count the number of nodes (N), edges (E), and connected components (P) to calculate the cyclomatic complexity of the given pseudocode.

From the flowchart, we can count:

- N = 15

- E = 18

- P = 1

Therefore, the cyclomatic complexity is:

$$M = E - N + 2P$$

$$M = 18 - 15 + 2(1)$$

$$M = 5$$

So the cyclomatic complexity of the given pseudocode is **5.**