

Submitted by

Arnav Srivastava [RA2111030010066]

Under the Guidance of

Dr. Gouthaman. P

Assistant Professor, Department of Networking and Communications

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

with specialization in Information Technology



SCHOOL OF COMPUTING

**COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND
TECHNOLOGY KATTANKULATHUR - 603203**

MAY 2023



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

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUE – 603 203
Chengalpattu District

BONAFIDE CERTIFICATE

Register No. RA2111030010066 Certified to be the

bonafide work done by Arnav Srivastava of II Year/IV Sem B.Tech Degree
Course in the **Practical Course – 18CSC206J - Software Engineering and
Project Management** in **SRM INSTITUTE OF SCIENCE AND
TECHNOLOGY**, Kattankulathur during the academic year 2022 – 2023.



SIGNATURE 25/02/2023

Faculty In-Charge
Dr. Gouthaman. P
Assistant Professor
Department of Networking and Communications
SRM Institute of Science and Technology

SIGNATURE

HEAD OF THE DEPARTMENT
Dr. Annapurani Panaiyappan. K
Professor and Head,
Department of Networking and Communications
SRM Institute of Science and Technology

ABSTRACT

This software engineering project is a software application that helps groups of people divide and split expenses, such as a restaurant bill or household bills, in a fair and convenient way. The app allows users to input the total amount of the bill and the number of people sharing it, and then calculates each person's share of the expense, including tax and tip, based on various methods such as splitting equally, by percentage, or by itemizing the bill.

The app may also offer additional features such as the ability to track and split expenses over time, set up group accounts for shared expenses, and send reminders to users about upcoming bills or overdue payments. It may also integrate with popular payment platforms such as Paytm, PayPal, or Gpay to facilitate the transfer of money between users.

In addition, the app may offer a user-friendly interface that is easy to navigate and customizable based on the user's preferences. It may also provide data analytics and reporting tools to help users track their spending habits and identify areas where they can save money.

Overall, a bill splitting app is a useful tool for anyone who wants to simplify the process of dividing expenses among friends, family, or roommates and avoid potential conflicts or misunderstandings.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	III
	LIST OF FIGURES	V
	LIST OF ABBREVIATIONS	VI
1	PROBLEM STATEMENT	1
2	STAKEHOLDERS & PROCESS MODELS	5
3	IDENTIFYING REQUIREMENTS	8
4	PROJECT PLAN & EFFORT	11
5	WORK BREAKDOWN STRUCTURE & RISK ANALYSIS	16
6	SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM	21
7	ENTITY RELATIONSHIP DIAGRAM	25
8	DATA FLOW DIAGRAM	27
9	SEQUENCE & COLLABORATION DIAGRAM	30
10	DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE	33
11	TEST CASE AND REPORTING	37
12	ARCHITECTURE/DESIGN/FRAMEWORK/IMP- LEMENTATION	43
	CONCLUSION	58
	REFERENCES	59

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
5.1	Work Breakdown Structure	17
5.2	Gantt Chart	18
6.1	System Architecture	22
6.2	Use Case Diagram	23
6.3	Class Diagram	23
7.1	Entity Relationship Diagram	26
8.1	Data Flow Diagram Level-0	28
8.2	Data Flow Diagram Level-1	29
9.1	Sequence Diagram	31
9.2	Collaboration Diagram	32
12.1	Initial Screen	55
12.2	Signup/Login Page	56
12.3	Expense Handling and report	56

LIST OF ABBREVIATIONS

ABBREVIATION	WORD
EXP	Experiment
ER	Entity Relation
WBS	Work Breakdown Structure
DFD	Data Flow Diagram
TID	Test ID



School of Computing

SRM IST, Kattankulathur – 603 203

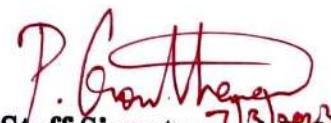
Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	1
Title of Experiment	To identify the Software Project, Create Business Case, Arrive at a Problem Statement
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	23/01/23

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	5
Total		10	9


P. G. Ravichandran
Staff Signature with date

Aim

To Frame a project team, analyze and identify a Software project. To create a business case and Arrive at a Problem Statement for the <FINAC>

Team Members:

S. No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Lead/Rep
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Project Title: FINAC

Project Description

- FINAC is an app designed to split expenses between friends and keep track of who owes whom. It allows users to log expenses they make on behalf of a group and divide the total cost among all participants.
- Debt Management: Helps users keep track of who owes whom and facilitates settling debts by sending reminders and facilitating payment transfers within the app.
- Fairness: Ensures that everyone pays their fair share by accurately tracking expenses and debts.
- Convenience: Provides a simple and convenient platform for managing shared expenses, avoiding confusion, and settling debts.
- Expense Tracking: Allows users to log expenses made on behalf of a group and divide the total cost among all participants.

ONE PAGE BUSINESS CASE TEMPLATE

DATE	23-01-2023
SUBMITTED BY	Aswin Sujith Varghese
TITLE / ROLE	Finac – Self-Financial Management System



Finac
Your financial future, made simple

THE PROJECT

In bullet points, describe the problem this project aims to solve or the opportunity it aims to develop.

- This project aims to split bills and manage expenses of the user.
- It is intended to ease splitting in groups.
- The total amount of money on the bill & how many people will split the bill will be used to split the bill.
- A track of categories (entertainment, food, transport etc) where the money was spent will be maintained

THE HISTORY

In bullet points, describe the current situation.

- Many individuals struggle with manual financial tracking methods, leading to inaccuracies and inconsistencies in their financial data.
- It can be difficult to get a clear view of an individual's financial situation and make informed financial decisions without proper tracking and analysis of income and expenses.
- There is a lack of personalized guidance and support for financial decision-making, making it challenging for people to improve their financial health and reach their financial goals.
- Existing financial management solutions may not be user-friendly, accessible, or secure, leading to low adoption and limited effectiveness.
- Financial literacy is often limited, leaving individuals without the knowledge and skills needed to effectively manage their finances.

LIMITATIONS

List what could prevent the success of the project, such as the need for expensive equipment, bad weather, lack of special training, etc.

- Limited to a small group of friends: FINAC is designed for small groups of friends, and may not be suitable for larger groups or organizations.
- Limited payment options: The app currently only supports payment transfers within the app, and may not be suitable for those who prefer alternative payment methods.
- Network Dependent: The app requires an internet connection to work properly and might not be the best option for those without a reliable internet connection.

- Limited to a specific geographic location: GoDutch may not be available in all countries, and the payment transfer feature may be restricted to specific geographic locations.
- Limited Scope: GoDutch is designed specifically for tracking shared expenses among friends and may not be suitable for more complex financial management needs.

APPROACH

List what is needed to complete the project.

To complete a self-financial management system project, the following components are required:

- User interface and design: A user-friendly interface for inputting, organizing, and analyzing financial data.
- Data management and storage: A secure and efficient database for storing and retrieving financial data.
- Financial tracking and analysis: The ability to track income, expenses, and savings and provide insights, tips, and recommendations based on the data.
- Security: Implementations to ensure the security of user financial information.
- Integration with financial institutions: Integration with financial institutions to automatically import financial data.
- Technical expertise: A team with the technical expertise to build and maintain the platform. Front-end and back-end developers
- Financial knowledge: A team with financial knowledge to provide relevant insights, tips, and recommendations to users.
- Testing and quality assurance: A testing and quality assurance process to ensure the platform is functioning correctly and meets users' needs
- Tech-stack: HTML, CSS, JavaScript, React, Node.js, NoSQL database, cloud services like, AWS, GCP or Azure and tools for collaboration like Git, etc.

BENEFITS

In bullet points, list the benefits that this project will bring to the organization.

- The organization will gain user in numerous amount as it can be used and is required by all
- The organization can use its user base to link with other departments and use the data to display appropriate advertisements.
- For example: The organization can track the monthly expense criterias and suggest appropriate credit cards etc

Result

Thus, the project team formed, the project is described, the business case was prepared and the problem statement is arrived.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	2
Title of Experiment	Identification of Process Methodology and Stakeholder Description
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	30/01/23

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	5
2	Viva	5	4
Total		10	9

P. Gautham 13/1/2023
Staff Signature with date

Aim

To identify the appropriate Process Model for the project and prepare Stakeholder and User Description.

Team Members:

Sl No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep/Member
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Project Title:

Selection of Methodology

The Agile model will be applied to the development of the self-financial management system, it would involve an iterative and flexible approach to the development process. The following is an overview of the Agile model:

- **Planning and Requirements Gathering:** The development team will work with stakeholders and end-users to prioritize and gather the requirements for the self-financial management system.
- **Iterations and Sprint Planning:** The development team will break down the requirements into small, manageable chunks, and plan short sprints, typically lasting 1–4 weeks.
- **Development and Testing:** The development team will work on implementing the requirements for each sprint, and perform ongoing testing and quality assurance.
- **Feedback and Refinement:** The development team will receive feedback from stakeholders and end-users on each sprint, and incorporate this feedback into the next iteration.
- **Deployment and Maintenance:** The self-financial management system will be deployed incrementally, with each sprint building on the previous one. The development team will provide ongoing support and maintenance for the self-financial management system.

The Agile model is well-suited for projects with complex requirements, rapidly changing requirements, or where the end goal may not be clear at the start of the project. The flexible and iterative nature of the Agile model allows the development team to respond quickly to changes in requirements and end-user feedback, and ensures that the self-financial management system meets the needs of the stakeholders and end-users.

Incorporate information to below table regarding stakeholders of the project

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
Owner	Business ideators and founders.	High	High	High
Partners	Enhancing of the existing product	Low	Medium	High
Investors	Funding of the project.	High	High	High
Managers	Overlooking the workflow and smooth working of the strategy.	Medium	High	Medium
Employees	Creation of the application and website necessary.	Medium	Medium	Low
Customers	Buying products and providing feedback.	Medium	Medium	Low

Result

Thus the Project Methodology was identified and the stakeholders were described.



INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	3
Title of Experiment	System, Functional and Non-Functional Requirements of the Project
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	14-02-2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	7
2	Viva	5	4
	Total	10	8

P. Ganesh 7/3/2023
Staff Signature with date

Aim

To identify the system, functional and non-functional requirements for the project.

Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep/Member
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Project Title: <FINAC>

• 35.

System Requirements

- Web Development:** The self-financial management system is likely to be developed as a web application, requiring expertise in web development technologies such as HTML, CSS, JavaScript, and server-side programming languages such as PHP, Ruby, or Python.
- Database Management:** A self-financial management system will need to have robust data management capabilities, with features such as secure storage, data backup, and data recovery. The database management system such as MySQL, Oracle, or PostgreSQL may be required.
- User Interface:** The self-financial management system will require a user-friendly interface for users to input, track and analyze their financial data. The development team will need to have experience in UI/UX design and development to create an effective interface.
- Security:** Financial data is sensitive information and must be protected from unauthorized access. The development team will need to include security features such as encryption, authentication, and access control in the application to ensure the security of users' financial data.
- Analytics:** A self-financial management system may require data analysis tools to provide users with valuable insights and recommendations on their financial management practices. The development team will need to have expertise in data analysis and visualization such as line graphs, pie charts, etc.
- Performance:** The self-financial management system must be designed to handle a large number of users and data volume. The development team will need to consider performance optimization techniques such as caching, load balancing, and database optimization to ensure the system is performant and responsive.

Functional Requirements

FINAC is an app designed to split expenses between friends, group members, and colleagues. Here are some of the functional requirements of a typical FINAC app:

1. **User Management:** The app should allow users to create and manage their profiles, including profile information, payment methods, and transaction history.
2. **Group Management:** The app should allow users to create and manage groups, invite friends to join, and view the transaction history of the group.
3. **Expense Tracking:** The app should allow users to track expenses and see who owes what to whom, including the ability to add expenses, categorize them, and assign them to specific group members.
4. **Payment Processing:** The app should allow users to make payments directly from the app, either via credit card or another payment method, and track who has paid and who hasn't.
5. **Notifications:** The app should send notifications to users about upcoming expenses and remind them to pay their debts.
6. **Currency Conversion:** The app should support multiple currencies and automatically convert the amount to each user's local currency.
7. **Data Backup and Recovery:** The app should have robust data backup and recovery mechanisms to ensure that users' data is secure and can be easily recovered in case of any data loss.
8. **User-Friendly Interface:** The app should have a user-friendly interface that is easy to navigate and understand, making it accessible to users of all ages and technical proficiency.

Non-Functional Requirements

Non-functional requirements describe how a system should perform, and specify the qualities or attributes that the system should have. Some of the possible non-functional requirements for a self-financial management system might include:

1. **Security:** The system should be secure and protect user data from unauthorized access.
2. **Scalability:** The system should be scalable, with the ability to handle increasing amounts of data and users.
3. **Reliability:** The system should be reliable, with data backup and recovery features to ensure that user data is not lost.
4. **Usability:** The system should be easy to use, with a user-friendly interface and intuitive navigation.
5. **Accessibility:** The system should be accessible to all users, including those with disabilities or using assistive technologies.

6. **Availability:** The system should be available to users 24/7, with minimal downtime for maintenance or upgrades.
7. **Compatibility:** The system should be compatible with different browsers, devices, and operating systems.
8. **Maintainability:** The system should be easy to maintain and upgrade, with modular code and clear documentation.
9. **Performance under stress:** The system should be designed to perform well even under heavy usage, such as during peak usage periods or high traffic events.

Result

Thus the requirements were identified and accordingly described.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	4
Title of Experiment	Prepare Project Plan based on scope, Calculate Project effort based on resources and Job roles and responsibilities
Name of the candidate	Arnav Srivastava
Team Members	Aswin Sujith Varghese, Anish Bharat
Register Number	RA2111030010066
Date of Experiment	21-02-2023

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	5
2	Viva	5	3
Total		10	8

A handwritten signature in black ink, appearing to read 'P. Gautham Palaniswami'.
Staff Signature with date

Aim

To Prepare Project Plan based on scope, Calculate Project effort based on resources, Find Job roles and responsibilities

Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep/Member
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Project Title: <FINAC>

1. Project Management Plan

<u>Focus Area</u>	<u>Details</u>
Cost Management:	Cost estimation Cost budgeting Cost control
Quality Management :	Defining quality objectives Creating quality standards Quality planning Quality assurance Quality control
Stakeholder:	Identifying Analyzing Engaging Stakeholders

2. Estimation

2.1. Effort and Cost Estimation

Activity Description	Sub-Task Description	Effort (in hours)	Hourly Rate (INR)	Cost in INR
Project Planning and Management	organizing, coordinating, and controlling resources (people, time, and money)	60	750 INR	INR 45,000
Requirements Gathering and Analysis	identifying, understanding, and documenting	80	750 INR	INR 60,000
System Design and Architecture	overall structure and components of a software or technology system.	100	1125 INR	INR 1,12,500
Front-end Development	Develop user interface	200	1125 INR	INR 2,25,000
Back-end Development	Design, Develop and Unit Test Services/API/DB	300	1500 INR	INR 4,50,000
Database Design and Development	Requirements analysis conceptual and logical design	80	1500 INR	INR 1,20,000
Integration and Testing	Implementation, testing and validation	100	1500 INR	INR 1,50,000
User Acceptance Testing and Deployment	Test planning, executing ,defect tracking and resolution	60	1125 INR	INR 67,500
Documentation and User Training	Identifying user needs, creating documentation and conducting user testing	40	750 INR	INR 30,000
Contingency	Risk identification, analysis, risk monitoring and control	-	-	INR 3,75,000
Total	-	1020	-	INR 16,35,000

2.2. Infrastructure/Resource Cost [CapEx]

Infrastructure Requirement	Qty	Cost per qty	Cost per item
Server hardware	2	50,000	1,00,000
Networking equipment	1	20,000	20,000
Storage devices	2	15,000	30,000
Workstations for development team	5	30,000	1,50,000
Software licenses	-	-	50,000
Cloud hosting fees (1 year)	-	-	1,20,000
Total infrastructure and resource cost (CapEx)	-	-	4,70,000

2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum
Infrastructure Maintenance	Server maintenance (includes hardware and software maintenance)	2	INR 75,000
Infrastructure Maintenance	Storage device maintenance (includes hardware and software maintenance)	2	INR 37,500
Infrastructure Maintenance	Networking equipment maintenance (includes hardware and software maintenance)	2	INR 37,500
Application Maintenance	Application server maintenance (includes hardware and software maintenance)	1	INR 75,000
Application Maintenance	Database maintenance (includes hardware and software maintenance)	1	INR 75,000
Application Maintenance	Software updates and bug fixes	-	INR 3,75,000
User Support	Help desk support (phone, email, chat)	-	INR 3,75,000
Contingency	Unforeseen maintenance and support costs	-	-
Total	-	8	INR 11,25,000

3. Project Team Formation

3.1. Identification Team members

Name	Role	Responsibilities
Arnav Srivastava	Key Business User (Product Owner)	Provide clear business and user requirements
Arnav Srivastava	Project Manager	Manage the project
Anish Bharat	Business Analyst	Discuss and Document Requirements
Aswin Sujith Varghese	Technical Lead	Design the end-to-end architecture
Anish Bharat	UX Designer	Design the user experience
Aswin Sujith Varghese	Frontend Developer	Develop user interface
Arnav Srivastava	Backend Developer	Design, Develop and Unit Test Services/API/DB
Anish Bharat	Cloud Architect	Design the cost effective, highly available and scalable architecture
Aswin Sujith Varghese	Cloud Operations	Provision required Services
Arnav Srivastava	Tester	Define Test Cases and Perform Testing

3.2. Responsibility Assignment Matrix

RACI Matrix Activity	Team Members					
	Business Analyst	Developer	Designer	Project Manager	Quality Assurance	Key Business User
User Requirement Documentation	A	C/I	C	I	I	R
Gather requirements and define scope	R	C	C	A	I	R
Design user interface and user experience	C	A	R	I	C	C
Develop front-end code	C	R	C	I	I	C
Develop back-end code	C	R	C	I	I	C
Integrate front-end and back-end code	C	R	C	I	I	C
Conduct testing and quality assurance	C	C	C	I	R	C
Deploy application to production environment	C	R	C	A	I	C/I
Provide ongoing maintenance and support	C	R	C	A	I	C/I

A	Accountable
R	Responsible
C	Consult
I	Inform

Result:

Thus, the Project Plan was documented successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	5
Title of Experiment	Prepare Work breakdown structure, Timeline chart, Risk identification table
Name of the candidate	Arnav Srivastava
Team Members	Aswin Sujith Varghese, Anish Bharat
Register Number	RA2111030010066
Date of Experiment	27/02/23

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	3
Total		10	7

P. Goutham 25/3/2023
Staff Signature with date

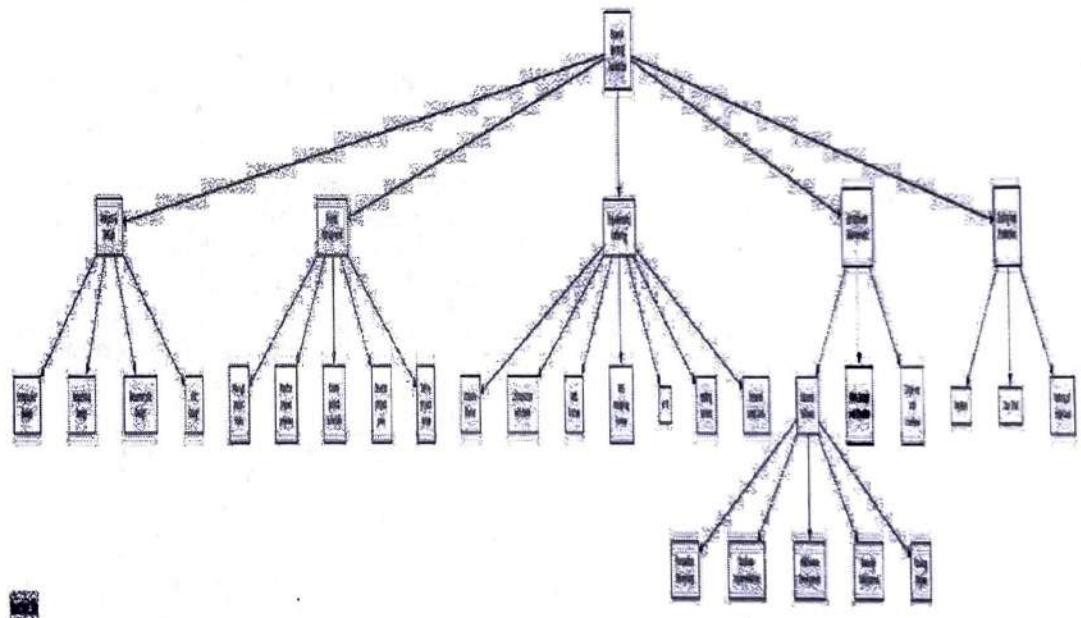
Aim

To Prepare Work breakdown structure, Timeline chart and Risk identification table

Team Members:

Sl No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

RISK ANALYSIS – SWOT & RMMM



0.0 Expense Splitting Application

1.0 Project Management

- 1.1 Define project scope
- 1.2 Develop project plan
- 1.3 Create project schedule
- 1.4 Monitor project progress
- 1.5 Manage project risks

2.0 Requirements Gathering

- 2.1 Personal computers
- 2.2 Hosting servers
- 2.3 Wifi
- 2.4 Web designing license
- 2.5 AWS license
- 2.6 Collaboration with bank
- 2.7 Domain license

3.0 Analysis & Design

- 3.1 Flat Design
- 3.2 Skeuomorphic Design
- 3.3 Responsive Design
- 3.4 Infographic Design

4.0 Site Software Development

- 4.1 HTML Design and Creation
- 4.2 Backend Software
 - 4.2.1 Database Implementation
 - 4.2.2 Middleware Development
 - 4.2.3 Security Subsystems
 - 4.2.4 Catalog Engine
 - 4.2.5 Transaction Processing
- 4.3 Graphics and Interface

5.0 Testing and Production

- 5.1 Dry Run
- 5.2 User Trial
- 5.3 testing of Expenses

TIMELINE – GANTT CHART

Task	Duration	Start Date	End Date	Dependencies
1. Define Requirements	2 weeks	03/21/2023	04/04/2023	None
2. Design UI/UX	3 weeks	04/05/2023	04/25/2023	1
3. Develop Backend	4 weeks	04/26/2023	05/23/2023	2
4. Develop Frontend	4 weeks	05/24/2023	06/20/2023	2
5. Integration and Testing	2 weeks	06/21/2023	07/04/2023	3,4
6. Bug Fixing	1 week	07/05/2023	07/11/2023	5
7. User Acceptance Testing	2 weeks	07/12/2023	07/25/2023	5

8. Deployment	1 week	07/26/2023	08/01/2023	7
9. Post-deployment Support	Ongoing	08/02/2023	-	8

SWOT ANALYSIS

Strengths	Weaknesses
<ul style="list-style-type: none"> - Streamlines and automates expense tracking and reporting process - Enables real-time visibility into company spending and budget management - Reduces the risk of errors and fraud in expense reporting - Provides valuable data insights and analytics to inform business decisions 	<ul style="list-style-type: none"> - May not be able to track all expenses accurately - Users may be hesitant to share financial information with others - May not be useful for individuals who live alone - May require a significant amount of development and marketing to gain traction in the market
Opportunities	Threats
<ul style="list-style-type: none"> - Could be expanded to include other financial management features such as bill payments and budget tracking - Could partner with banks or other financial institutions to offer additional services - Could target specific demographics, such as college students or young professionals - Could expand globally to other markets where expense-splitting apps are not yet popular 	<ul style="list-style-type: none"> - Competition from other expense-splitting apps or financial management tools - Changes in consumer behavior or preferences - Changes in regulations or laws that affect financial apps - Economic downturns or other external factors that affect consumer spending habits.

Risk Management Framework

Response	Strategy	Examples
Avoid	Risk avoidance is a strategy where the project team takes action to remove the threat of the risk or protect from the impact	<ul style="list-style-type: none"> • Lack of accountability: • Poor technology integration • Inadequate communication
Transfer	Risk transference involves shifting or transferring the risk threat and impact to a third party. Rather transfer the responsibility and ownership.	<ul style="list-style-type: none"> • Warranties • Insurance coverage • Payment processors • Fraud detection software
Mitigate	Risk mitigation is a strategy where the project team takes action to reduce the probability of the risk occurring. This does not risk or potential impact, but rather reduces the likelihood of it becoming real.	<ul style="list-style-type: none"> • Conducting regular quality checks • Regular communication and monitoring • Implementing safety measures
Accept	Risk acceptance means the team acknowledges the risk and its potential impact, but decides not to take any preemptive action to prevent it. It is dealt with only when it occurs.	<ul style="list-style-type: none"> • Policy violations • Fraudulent expense claims • Late expense submission

Result:

Thus, the work breakdown structure with timeline chart and risk table were formulated successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	6
Title of Experiment	Design a System Architecture, Use Case and Class Diagram
Name of the candidate	Anish Bharat
Team Members	Aswin Sujith Varghese , Arnav Srivastava
Register Number	RA2111030010092
Date of Experiment	03/03/23

Mark Split Up

S.No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	3
2	Viva	5	2
	Total	10	5

P. Gautham 23/3/2023
Staff Signature with date

Aim

To Design a System Architecture, Use case and Class Diagram

Team Members:

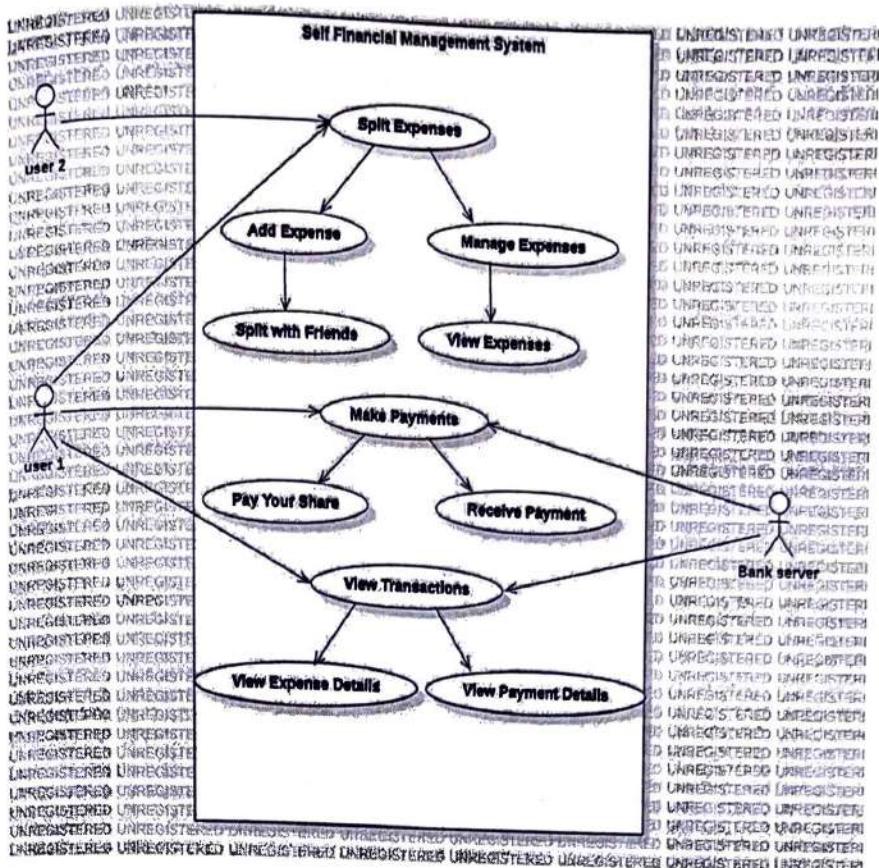
Sl No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Requirements

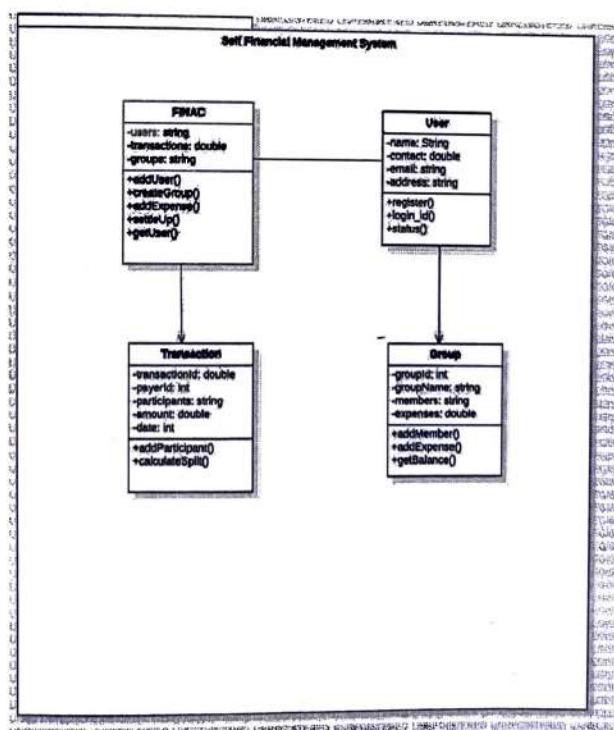
System Architecture:

1. Client-side application: The FINAC app is a mobile application that runs on both iOS and Android platforms. The client-side application handles user interactions and sends requests to the backend server.
2. Authentication and authorization: The company should have a strong authentication and authorization system that controls access to its resources. This can include two-factor authentication, access controls, and role-based access control.
3. Payment gateway: The payment gateway is a third-party service that handles payment processing for the FINAC app. It securely handles payment transactions and provides the backend server with transaction details.
4. Cloud infrastructure: The FINAC app is hosted on a cloud infrastructure, such as Amazon Web Services or Google Cloud Platform. This infrastructure provides scalability, security, and availability for the app.
5. Third-party integrations: The FINAC app may integrate with third-party services, such as messaging platforms or social media networks, to provide additional functionality for users.
6. Notification system: The FINAC app includes a notification system that alerts users of important events, such as payment confirmations, expense updates, or reminders. It uses push notifications or email notifications to keep users informed and engaged.
7. Analytics and monitoring: The FINAC app collects and analyzes user data to improve the user experience and track app performance. It may use tools like Google Analytics, Firebase, or New Relic to monitor app usage, detect errors, and measure user engagement.

Use Case Diagram:



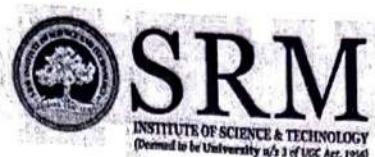
Class Diagram:



Caption

Result:

Thus, the system architecture, use case and class diagram created successfully.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	7
Title of Experiment	Design a Entity relationship diagram
Name of the candidate	Arnav Srivastava
Team Members	Aswin Sujith Varghese , Anish Bharat
Register Number	RA2111030010066
Date of Experiment	13/03/23

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	2
Total		10	6

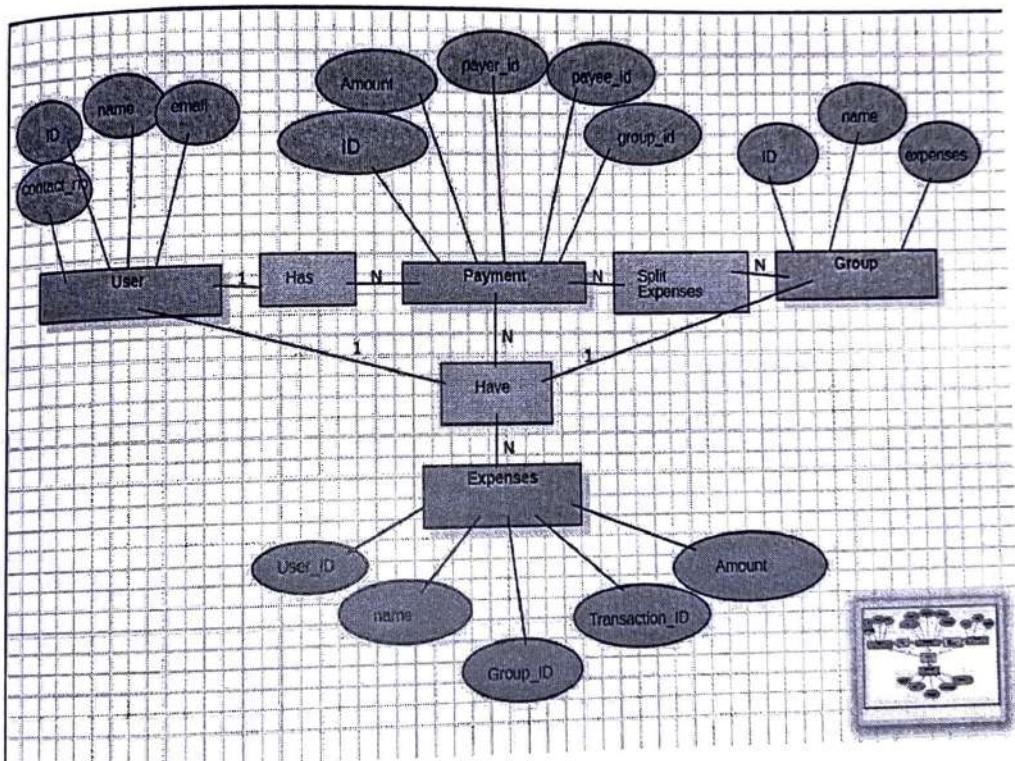
P. Senthilnath
Staff Signature with date

Aim

To create the Entity Relationship Diagram

Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep/Member
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Entity-Relationship Diagram**Result:**

Thus, the entity relationship diagram was created successfully.



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

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	8
Title of Experiment	Develop a Data Flow Diagram (Process-Up to Level 1)
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	23/03/2023

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	3
2	Viva	5	4
Total		10	7

P.G. Senthil 30/3/2023
Staff Signature with date

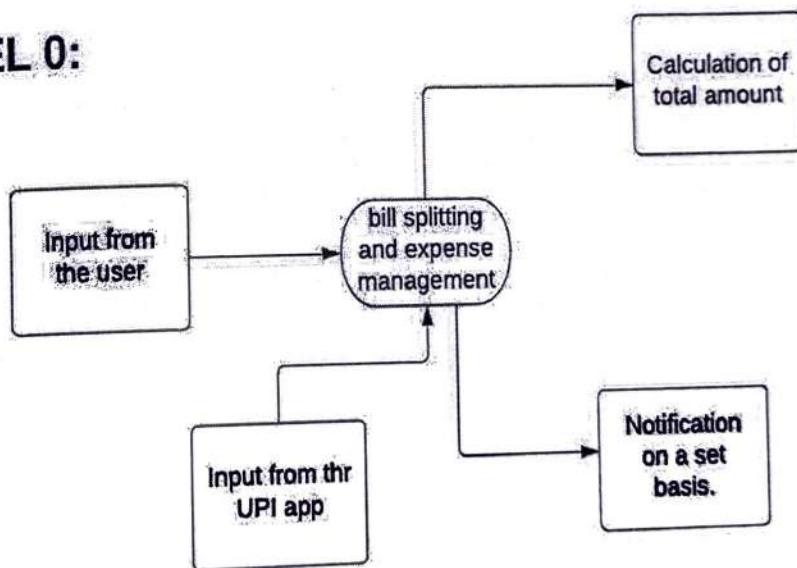
Aim
To develop the data flow diagram up to level 1 for the <project name>

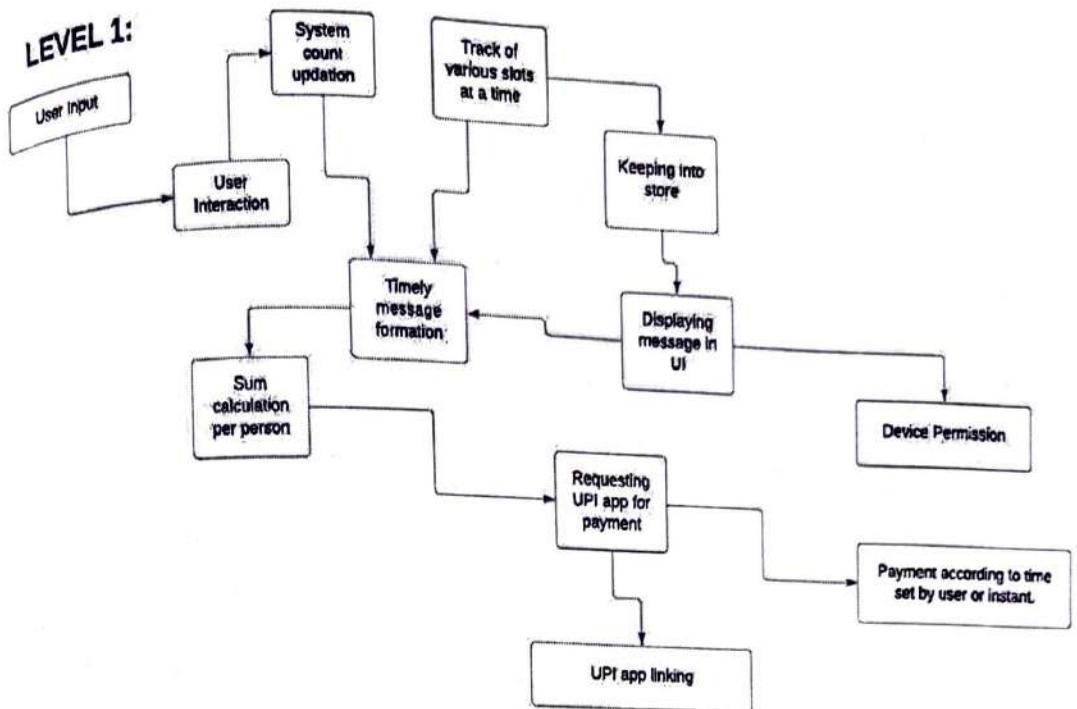
Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Data Flow Diagram

LEVEL 0:





Result:

Thus, the data flow diagrams have been created for the Bill Splitting and expense management system.



INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University u/s 3 of UGC Act, 1956)

School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	9
Title of Experiment	Design a Sequence and Collaboration Diagram
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	30-03-23

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	3
Total		10	7

P. Goutham 14/03/23
Staff Signature with date

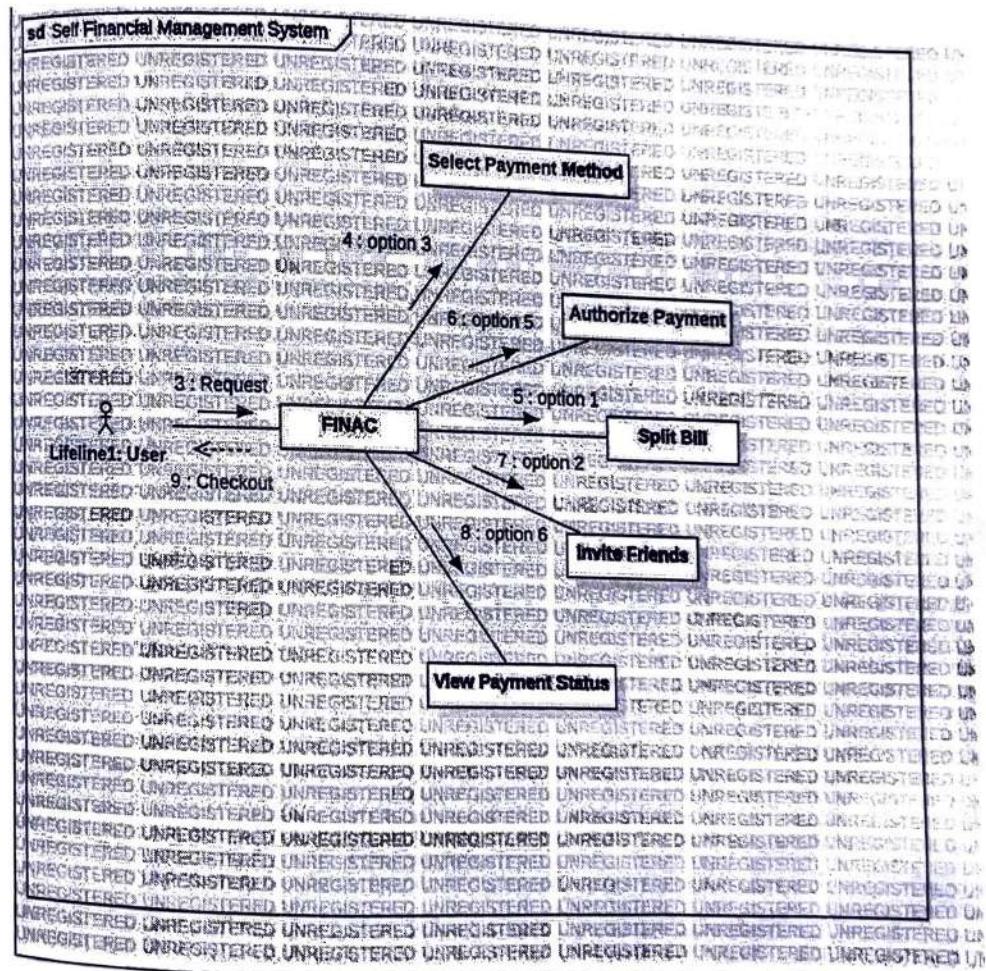
Aim

To create the sequence and collaboration diagram for the Finac - Self Financial Management System

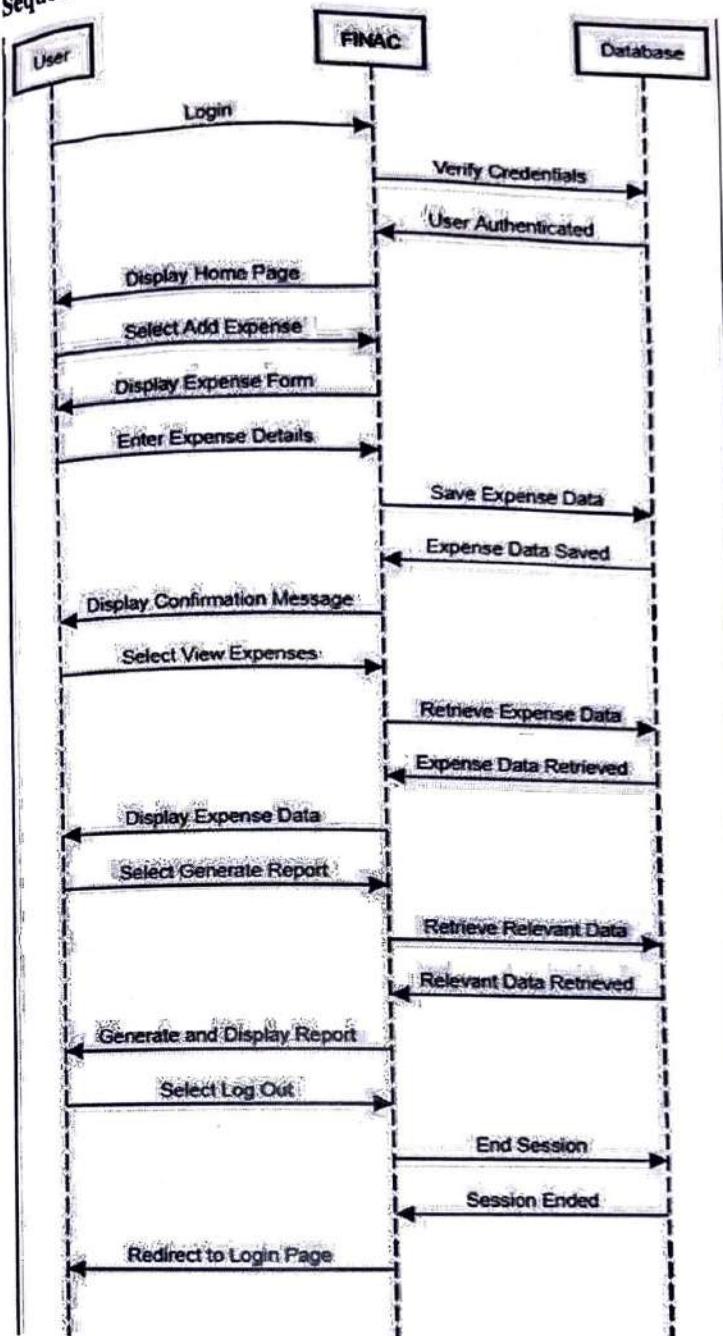
Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep/Member
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Collaboration Diagram:



Sequence Diagram:



Result:

Thus, the sequence and collaboration diagrams were created for the <project name>.



School of Computing

SRM IST, Kattankulathur - 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	10
Title of Experiment	Develop a Testing Framework/User Interface
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	30-03-23

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	5
2	Viva	5	5
Total		10	10

P. Gautham 24/4/2023

Staff Signature with date

Aim To develop the testing framework and/or user interface framework for the Finac

Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep/Member
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Executive Summary:

This document explains the various activities performed as part of the Testing of the 'Finac' application. 'Finac' is an app-based expense splitting and budget application. It is used for use for splitting and tracking the user's expense . There are several modules which are integrated to fulfill the purpose. Reactive approach is used

Test Plan

To create a test plan for FINAC, we need to consider different testing types, such as functional testing, integration testing, performance testing, and usability testing, among others. Here's a high-level test plan for an expense management program:

1. Functional Testing:

- Verify that all features of the expense management program are working as expected.
- Test the user interface to ensure that all buttons, menus, and tabs are functioning correctly.
- Verify that all input fields are correctly labeled and data is correctly validated.
- Test the application to ensure that data is being accurately saved and retrieved.
- Verify that reports and graphs are being generated correctly.

2. Integration Testing:

- Test the integration between the expense management program and other systems such as accounting software, project management tools, and HR systems.
- Verify that data is correctly synced between the expense management program and other systems.

3. Performance Testing:

- Test the system's response time, scalability, and load handling capabilities.
- Conduct load testing to ensure that the system can handle a large number of transactions simultaneously.
- Conduct stress testing to ensure that the system can handle heavy loads.

4. Usability Testing:

- Test the program's user interface to ensure that it is easy to use and navigate.
- Verify that the user interface is consistent throughout the application.
- Conduct user testing to get feedback on the application's usability.

5. Security Testing:

- Verify that the system is secure and that sensitive data is protected.
- Test authentication and authorization controls to ensure that only authorized users can access the system.
- Verify that all data is encrypted in transit and at rest.
- Conduct penetration testing to identify potential security vulnerabilities.

6. Localization Testing:

- Test the program's localization features to ensure that it works correctly with different languages, currencies, and regions.

7. Compatibility Testing:

- Test the application on different operating systems, browsers, and devices to ensure compatibility.

Overall, the goal of testing an expense management program is to ensure that it meets the user's needs, is easy to use, and functions correctly.

Scope of Testing

Functional:

1. Creating expense reports:

- Testing the ability to create expense reports in different formats, such as PDF or Excel.
- Testing the functionality to add expenses to a report, including the ability to upload receipts and attach files.
- Testing the ability to categorize expenses, such as by type, date, and project.

2. Approval workflows:

- Testing the approval workflow to ensure that the appropriate person approves each expense report.
- Testing the ability to configure different workflows based on the organization's requirements.
- Testing the ability to set up alerts and notifications for approval requests.

3. Budget tracking:

- Testing the ability to set and track budgets for different projects or departments.
- Testing the ability to view spending trends and identify areas of overspending or underspending.

4. Payment processing:

- Testing the ability to process payments for approved expenses, such as via bank transfer or credit card.
- Testing the ability to set up payment schedules and manage payment histories.

5. Reporting and analysis:

- Testing the ability to generate reports and graphs based on different criteria, such as by project, department, or date range.
- Testing the ability to analyze spending patterns and identify areas for cost savings.

Overall, the functional scope of testing for FINAC would involve thoroughly testing all the program's features and functionalities to ensure that they are working as expected and meet the user's requirements.

Non-Functional:

1. Performance Testing:

- Testing the program's ability to handle large volumes of data and transactions without crashing or slowing down
- Testing the program's response time to user actions and requests

2. Usability Testing:

- Testing the program's user interface and user experience to ensure that it is intuitive and easy to use

- Testing the program's accessibility for users with disabilities

3. Security Testing:

- Testing the program's security features, such as encryption, authentication, and authorization, to ensure that sensitive data is protected
- Testing the program for vulnerabilities and conducting penetration testing to identify potential security issues

4. Compatibility Testing:

- Testing the program's compatibility with different operating systems, browsers, and devices to ensure that it works correctly on a variety of platforms
- Testing the program's compatibility with other systems, such as accounting software or HR systems

5. Localization Testing:

- Testing the program's ability to handle different languages, currencies, and regions to ensure that it works correctly for users in different locations

6. Reliability and Availability Testing:

- Testing the program's ability to run for long periods without crashing or failing
- Testing the program's availability to ensure that it is accessible to users at all times

Overall, the non-functional scope of testing for FINAC would involve testing different aspects of the program's performance, usability, security, compatibility, reliability, and availability to ensure that it meets the user's requirements and works effectively in different scenarios.

Types of Testing, Methodology, Tools

Black Box Testing: The Black Box Test is a test that only considers the external behavior of the system; the internal workings of the software are not taken into account. A tester provides input and observes the output generated by the system under test. Black box testing can be applied to three main types of tests: functional, non-functional, and regression testing.

White Box Testing: The White Box Test is a method used to test a software taking into consideration its internal functioning. It is carried out by testers. Test cases for white box testing are derived from the design phase of the software development lifecycle.

Category	Methodology	Tools Required
----------	-------------	----------------

Sign-up page test case	Black Box Testing	Selenium, TestComplete, Apache JMeter
Login page test cases	Black Box Testing	Selenium, TestComplete, Apache JMeter
home page test cases	Black Box Testing	Selenium, TestComplete, Apache JMeter
Money Calculation Test Cases	Black Box Testing	Selenium, TestComplete, Apache JMeter
Expense tracking test cases	Black Box Testing	Selenium, TestComplete, Apache JMeter
Frontend	White Box Testing	Jest, Enzyme, Cypress, Selenium WebDriver
Backend	White Box Testing	JUnit, TestNG, Mockito, EasyMock
Graphic user interface	White Box Testing	Figma, Sketch, Adobe XD, InVision, Balsamiq Mockups

Result:

Thus, the testing framework/user interface framework has been created for the Finac.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	11
Title of Experiment	Test Cases & Reporting
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	18/06/23

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	4
2	Viva	5	4
Total		10	8

P. Goutham 24/4/2023
Staff Signature with date

Aim To develop the test cases manual with manual test case report for the Self-Financial Management System.

Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Test Case

Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Login	Valid Login	1. Enter valid username and password 2. Click on Login button	User should be redirected to the Home page	User is redirected to the Home page	Pass	
2	Login	Invalid Login	1. Enter invalid username and password 2. Click on Login button	User should be shown an error message and prompted to try again	User is shown an error message and prompted to try again	Pass	
3	Add Expense	Valid Expense Entry	1. Click on Add Expense button 2. Fill in all required fields	Expense should be saved and added to the database	Expense is saved and added to the database	Pass	

			3. Click on Save button				
4	Add Expense	Invalid Expense Entry	1. Click on Add Expense button 2. Fill in some or all fields with invalid data 3. Click on Save button	User should be shown an error message and prompted to enter valid data	User is shown an error message and prompted to enter valid data	Pass	
5	View Expenses	Filtered View	1. Click on View Expenses button 2. Apply filters to the view (e.g. by date, category, amount) 3. Click on Filter button	Only expenses matching the selected filters should be displayed	Only expenses matching the selected filters are displayed	Pass	
6	View Expenses	Export Data	1. Click on View Expenses button 2. Select export option (e.g. CSV, Excel) 3. Click on Export button	Expense data should be exported in the selected format and saved	Expense data is exported in the selected format and saved to users	Pass	
7	Logout	Valid Logout	1. Click on Logout button	User should be logged out and redirected to the Login page	User is logged out and redirected to the Login page	Pass	

Non-Functional Test Cases

Test ID (#)	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
NF-1	Usability	User 1 Interface	Navigate through the website	Website is easy to navigate and use	User finds it difficult to navigate the website	Fail	Improve website navigation
NF-2	Performance	Page Load Time	Open website	Website loads quickly	Website takes a long time to load	Fail	Improve website performance
NF-3	Security	Authentication	Log in to the website	User is able to successfully log in	Unauthorized access to the website	Fail	Improve website security measures
NF-4	Compatibility	Browser Compatibility	Access website from different browsers	Website is accessible from all major browsers	Website is not accessible from some browsers	Fail	Improve website compatibility with different browsers
NF-5	Scalability	Concurrent Users	Simulate multiple users accessing the website at the same time	Website is able to handle concurrent user requests without crashing	Website crashes or slows down when multiple users access it simultaneously	Fail	Improve website scalability

Category	Progress Against Plan	Status
Functional Testing	Amber	In-Progress
Non-Functional Testing	Red	Not-Completed

Functional	Test Case Coverage (%)	Status
Login with valid credentials	100%	Pass
Login with invalid credentials	100%	Pass
Create a new account	100%	Pass
View account balance	90%	Pass
View transaction history	90%	Pass
Add a new expense	95%	Pass
Add new income	95%	
Edit an expense	80%	Fail
Delete an expense	80%	Fail
Edit an income	80%	Fail
Delete an income	80%	Fail
Generate a report	85%	Pass

Result:

Thus, the test case manual and report has been created for the Self-Financial Management System.



School of Computing

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC206J

Course Name: Software Engineering and Project Management

Experiment No	12
Title of Experiment	Provide the details of Architecture Design/Framework/Implementation
Name of the candidate	Arnav Srivastava
Team Members	Anish Bharat, Aswin Sujith Varghese
Register Number	RA2111030010066
Date of Experiment	24-04-23

Mark Split Up

S. No	Description	Maximum Mark	Mark Obtained
1	Exercise	5	5
2	Viva	5	4
	Total	10	9

P. Gautham 28/4/2023
Staff Signature with date

Aim

To provide the details of architectural design/framework/implementation

Team Members:

S No	Register No	Name	Role
1	RA2111030010066	Arnav Srivastava	Rep/Member
2	RA2111030010079	Aswin Sujith Varghese	Member
3	RA2111030010092	Anish Bharat	Member

Initial Screen after opening the app for the first time:

This would just include the terms and conditions for the app and has a "get started" button.



Signup/Login Page:

After going through the conditions it would direct you to the signup/login page to enter your details accordingly.



Account settings:

This has option to edit your profile and it has all the general settings. In the "Edit Profile" option the user will have access to edit his card details, email, name etc.



Expense Handling and report:

This displays the users cards added to the app and the recent transaction the app will detect through messages and add it to the report. The user has option to transfer, top-up, pay, withdraw. In "more" there would be an option to split expense with your friends or add any additional expenses to your report.

The bottom bar has an option for "Stats" which would give a much more detailed information in categories of where money was spent. In the "notifications" tab it would mention transaction alerts or new expense generated for a split bill.



Result:

Thus, the details of architectural design/framework/implementation along with the screenshots were provided.

CONCLUSION

In summary, this project report has provided an overview of a bill splitting app that can prove to be a convenient tool for groups of people who want to split expenses evenly and fairly. The aim was to create an application that is not only functional and feature-rich but also user-friendly.

With a meticulous approach to planning and effective communication, we were able to overcome obstacles and deliver a final product that met the needs of our target audience. As such, we will try our best to frequently update and maintain it to ensure that it continues to meet the needs of its users.

We believe this application can help reduce the hassle and potential conflict of manually calculating and dividing up expenses. They offer a variety of features, such as the ability to track expenses, split bills in different ways, and even send payment requests to group members.

REFERENCES

- <https://www.geeksforgeeks.org/>
- <https://www.lucidchart.com/pages>
- <https://www.visual-paradigm.com/>
- <https://www.javatpoint.com/>
- <https://creately.com/>
- <https://www.quora.com/>