SecureSense – Leveraging Human Behavior for Security and Building a Secure Digital Culture **(Mobile Application Platform)**

Final Year Project Report

Submitted by

Umair Younus Khan
(2093-2020)

Abu Uzair
(1989-2021)

Supervisor

Muhammad Salman

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science
2025

Faculty of Engineering Sciences and Technology
Hamdard Institute of Engineering and Technology

Hamdard University, Main Campus, Karachi, Pakistan

Certificate of Approval

Faculty of Engineering Sciences and Technology

Hamdard Institute of Engineering and Technology
Hamdard University, Karachi, Pakistan

This project SecureSense (Mobile Survey App is presented by **Umair Younus Khan and Abu Uzair** under the supervision of their project advisor and approved by the project examination committee, and acknowledged by the Hamdard Institute of Engineering and Technology, in the fulfillment of the requirements for the Bachelor degree of Computer Science

____

Mr. Muhammad Salman
(Project Supervisor)

____

In-charge FYP-Committee

____

(Project Co-Supervisor)

____

Chairman
(Department of Computing)

____
(Dean, FEST)

Authors' Declaration

We declare that this project report was carried out in accordance with the rules and regulations of Hamdard University. The work is original except where indicated by special references in the text and no part of the report has been submitted for any other degree. The report has not been presented to any other University for examination.

Dated: 17-01-2025

Authors Signatures:

____
Umair Younus Khan

____
Abu Uzair

Plagiarism Undertaking

We, Umair Younus Khan and Abu Uzair solemnly declare that the work presented in the Final Year Project Report titled Secure Sense Mobile Survey App has been carried out solely by ourselves with no significant help from any other person except few of those which are duly acknowledged. We confirm that no portion of our report has been plagiarized and any material used in the report from other sources is properly referenced.

Dated:

Authors Signatures:

____

Umair Younus Khan

____

Abu Uzair

Acknowledgments

We would like to express our heartfelt gratitude to our project supervisor, **Muhammad Salman** whose expert guidance, valuable insights, and constant encouragement have played an integral role in the successful completion of this project. Their unwavering support has been essential at every stage, from conceptualization to the final implementation of the Survey App.

We are also deeply thankful to **Hamdard University** and its dedicated faculty members for providing us with the resources, knowledge, and a conducive environment to carry out our work. Their support in terms of both academic and technical resources has been pivotal to our success.

Our sincere appreciation goes to our team members, whose hard work, dedication, and collaborative efforts have been fundamental in bringing this project to life. We also extend our thanks to the testers and users who provided invaluable feedback, helping us enhance the functionality and usability of the Survey App. Without their contributions, this project would not have reached its full potential.

We are truly grateful to our families and friends for their patience, understanding, and encouragement throughout the course of this project. Their moral support has been a source of motivation and strength.

Finally, we would like to acknowledge the powerful technologies that made this project possible, including React for frontend development, Firebase for backend services, and all other tools and frameworks that contributed to the app’s success. These technological resources have been essential in realizing the vision of this Survey App.

We are deeply appreciative of everyone who contributed to the completion of this project.

Document Information

Table 1: Document Information

Customer	
Project Title	Secure Sense (Leveraging Human Behaviour for Security and Building a Secure Digital Culture) Mobile Application
Document	Final Year Project Report
Document Version	1.0
Identifier	F YP-031/FL24
Status	Final
Author(s)	Umair Younus Khan, Abu Uzair.
Approver(s)	Muhammad Salman
Issue Date	08/01/2025

Definition of Terms, Acronyms, and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

Table 2: Definition of Terms, Acronyms, and Abbreviations

Term	Description
Survey App	A web application designed to create, share, and analyze surveys for user responses.
Frontend	The part of the application that interacts directly with the user, typically built using HTML, CSS, JavaScript, and React.
Backend	The server-side component that handles database interactions, user authentication, and business logic. In this app, Firebase is used for backend services.
React	A JavaScript library used to build user interfaces, particularly for single-page applications (SPAs).
Firebase	A platform developed by Google for backend services, including authentication, database, and hosting.
API (Application Programming Interface)	A set of protocols and tools that allow different software applications to communicate with each other.
CRUD	An acronym for Create, Read, Update, Delete, representing the basic operations on data in a database.
UI (User Interface)	The space where interactions between the user and the application occur, including buttons, forms, and navigation.
UX (User Experience)	The overall experience and satisfaction of the user when interacting with the application.
Database	A structured collection of data, used to store and retrieve user responses and survey data. In this app, Firebase Firestore is
	used.
Authentication	The process of verifying the identity of a user, ensuring secure access to the app. Firebase Authentication is used in this project.
Deployment	The process of making the application accessible to users over the internet, typically involving hosting the application on a server.
Prototype	A preliminary version of the app, built to demonstrate its basic functionality before final development.
Feedback Loop	The process of gathering user feedback and improving the app based on that input.
SWOT Analysis	A strategic planning tool used to identify the Strengths, Weaknesses, Opportunities, and Threats related to the project.

Gantt Chart	A visual project management tool that shows the schedule of tasks and milestones throughout the project timeline.
Version Control	A system that manages changes to code and tracks different versions of the app during development. Git is commonly used for version control.
Agile Methodology	A project management and software development methodology that emphasizes iterative progress, collaboration, and flexibility.
Bug	A flaw or fault in the application that causes it to behave unexpectedly or incorrectly.

Abstract

The **Survey App** is a dynamic and scalable web-based application designed to empower users to create, distribute, and analyze surveys effortlessly. By integrating modern web technologies like **React** for the frontend and **Firebase** for backend infrastructure, the app offers a seamless, intuitive user experience with features that cater to diverse survey needs. Its primary objective is to streamline the survey process for various sectors including education, research, business, and marketing, enabling organizations and individuals to gather real-time feedback, monitor responses, and derive actionable insights.

Key features include:

- **Survey Creation:** The app provides users with a wide range of customizable options to design surveys. Users can include various question types such as multiple choice, short answer, Likert scale, and rating scales, tailoring surveys to meet specific needs. The creation process is simple and user-friendly, allowing for quick deployment of surveys.
- **Survey Distribution:** Users can easily distribute their surveys via direct links, email, or even integrate them into websites and social media platforms for wider reach. The app ensures that surveys are accessible on any device, whether desktop or mobile, making it a flexible solution for diverse audiences.
- **Real-Time Data Collection:** As responses are submitted, they are automatically captured and stored securely in **Firebase**. This real-time data collection ensures that users can view and analyze results instantaneously, without delays.
- **User Authentication:** Powered by **Firebase Authentication**, the app provides secure login and registration functionality, ensuring that only authorized users can create, modify, and analyze surveys. This feature adds an extra layer of security, protecting user data and survey integrity.
- **Data Analysis & Visualization:** After collecting responses, the app offers built-in tools for analyzing the data, providing users with insights through data visualization such as graphs, pie charts, and bar charts. This makes it easier for users to interpret the data and identify trends and patterns.
- **Admin Panel:** The app includes an admin panel where survey creators can track survey progress, view results, and manage surveys efficiently. The admin panel is designed to provide users with control over their surveys and responses, making it an essential feature for organizations handling large volumes of data.

Keywords:

- Survey App
- React
- Firebase
- Survey Creation

- Real-Time Data Collection
- Data Analysis
- User Authentication
- Data Visualization
- Survey Distribution
- Admin Panel
- Data Storage
- Scalable Solution
- User Experience
- Web Technologies

Table of Contents

Project code: FYP-031/FL24

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Motivation

Surveys are crucial tools for collecting feedback across various domains such as market research, education, and customer satisfaction. However, many survey applications are limited in customization and flexibility. This project aims to develop a survey app that not only allows users to create customized surveys but also provides the ability to share, edit, and save them. By incorporating the built-in survey feature and user-specific IDs, the app enables greater flexibility and collaboration, making it a powerful tool for both individual users and guests.

1.2 Problem Statement

Many existing survey tools are either overly simplified or overly complex, with limited customization options for survey creation. Additionally, while survey sharing is available in some tools, the lack of user-specific IDs and the ability to save, edit, and re-use surveys can hinder collaboration. This survey app solves these issues by providing an intuitive interface for survey creation, built-in surveys, and comprehensive sharing and editing options, all with user-specific customization and MongoDB as the backend database.

1.3 Goals and Objectives

- To create an intuitive survey creation tool with customizable options (multiple-choice with scoring).
- To enable users to save, edit, and share surveys with unique user IDs.
- To allow users to access built-in surveys created by others and reuse them.
- To provide guests with access to saved surveys to view results, without the ability to edit or create surveys.
- To allow surveys to be downloaded as PDF files.
- To store survey data securely using MongoDB as the backend database.

1.4 Project Scope

The app will allow users to:

- Create surveys with multiple questions and options.
- Rate each option from 1 to 4 marks.
- Edit, delete, and add questions to saved surveys.
- Save surveys and make them available to other users.
- Share surveys via links or user-specific IDs.

- Download surveys in PDF format.
- Guests can access shared surveys, complete them, and view results, but cannot create or edit surveys. This app will be developed using React Native, MUI for the UI components, and MongoDB for database storage.

CHAPTER 2: RELEVANT BACKGROUND & DEFINITIONS

- **Customizable Survey Tools**:

- Allow users to create surveys tailored to specific needs.
- Offer flexibility for academic research, business analytics, and feedback collection. □ Enhance usability across diverse contexts.

- **Built-in Surveys**:

- Provide pre-designed templates for quick survey creation. □ Save time and effort for users without design expertise.

- **User-Specific IDs**:

- Facilitate individual response tracking.
- Ensure data integrity and accountability during analysis.

- **Guest Login**:

- Enable survey access for users without registration.
- Maintain system security through controlled permissions. □ Improve accessibility for a broader audience.

- **Survey Distribution and Access Models**:

- Support sharing surveys via multiple platforms (email, social media, etc.). □ Adapt to user needs for public or private survey participation.

CHAPTER 3: LITERATURE REVIEW & RELATED WORK

Literature Review

Survey apps have become increasingly important in various fields. Traditional tools like Google Forms and SurveyMonkey are popular but often lack deep customization, data export options, and user-specific survey management. Literature highlights that customization in surveys, particularly through scoring and reusable templates, enhances user experience and data accuracy.

Related Work

Survey platforms like Google Forms and Typeform allow survey creation, response collection, and data visualization, but they lack extensive customization, such as the ability to reuse or edit built-in surveys. The proposed system will take inspiration from these platforms but introduce features like multiple question options with ratings, survey sharing via userspecific IDs, and the ability to save and reuse surveys.

Gap Analysis

Existing survey apps often lack flexibility in terms of question creation, survey sharing, and user-specific management. Our system aims to fill these gaps by offering user-customized surveys with the option to create, save, and share surveys using unique user IDs, and the ability to share surveys with guest users for limited access.

CHAPTER 4
PROJECT DISCUSSION

---

1. Software Engineering Methodology

For our survey application, we followed the **Agile Software Development Methodology**. Agile allowed us to work in short, manageable iterations called sprints, where we continuously gathered feedback and improved our application. We prioritized working features and user feedback over strict documentation. Agile suited our academic timeline and the evolving nature of our project requirements.

---

2. Project Methodology

Our project followed the **Agile Scrum approach**. We divided the entire application development into sprints of 1–2 weeks. Each sprint focused on developing a small set of features, such as:

- Sprint 1: User Authentication (Login/Registration)
- Sprint 2: Survey Creation
- Sprint 3: Attempt Survey (User and Guest)
- Sprint 4: Submission and Scoring
- Sprint 5: Feedback, Improvements, and Finalization

Daily or weekly team discussions helped us track progress, identify blockers, and plan future tasks effectively.

---

3. Phases of Project

Our project was carried out in the following phases:

1. **Requirement Gathering:**
    - Identification of core features like login, survey creation, attempt, guest login, result calculation, etc.
2. **Design:**
    - Wireframing of key screens
    - Planning of backend APIs
3. **Development:**

  - Frontend built using React Native
  - Backend using Node.js + Express + MongoDB
4. **Testing:**
  - Manual testing on emulator and real Android device
  - Validating user input, score calculation, API integration
5. **Deployment:**
  - Final testing and packaging for deployment

---

4. Software/Tools Used in Project

- **Frontend:** React Native
- **Backend:** Node.js, Express.js
- **Database:** MongoDB with Mongoose
- **Development Tools:** Visual Studio Code
- **API Testing:** Postman
- **Version Control:** Git & GitHub
- **Mobile Testing:** Android Emulator, Real Android Device
- **Package Management:** npm
- **Authentication:** JWT (JSON Web Token)

---

5. Hardware Used in Project

- **Laptop/PC**
  - Processor: Intel i5 or above
  - RAM: Minimum 8 GB
  - OS: Windows 10/11 or macOS
- **Mobile Device**
  - Android smartphone for testing
- **Internet Connection**
  - Stable connection for API testing, npm installations, and Firebase/GitHub integration.

This methodology allowed us to be flexible, productive, and goal-oriented while developing our survey application.

Chapter 5: IMPLEMENTATION

5.1 Proposed System Architecture/Design

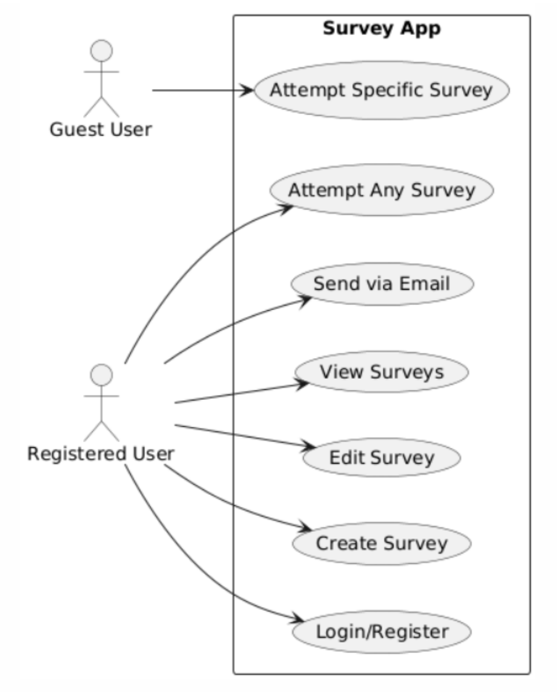
The proposed system is a Survey Application developed using a hybrid of client-server architecture. The mobile application, built with React Native, communicates with a Node.js Express backend server which connects to a MongoDB database. The architecture ensures a smooth interaction between the front-end and backend via RESTful APIs.

There are two types of users in the system:

- **Guest User**: Can only access a specific survey using survey ID and email, and is restricted to survey attempt functionality only.
- **Registered User**: Has access to the full functionality including survey creation, listing, editing, sharing via email, and attempting any surveys.

Communication between the app and server is secured using HTTP methods, and user data is stored using AsyncStorage on the client side and MongoDB on the server side.

Refer to the PlantUML-generated image placed here for visual architecture representation.



5.2 Functional Specifications

- User registration and authentication
- Create, edit, delete, and list surveys
- Survey consists of questions with multiple-choice options
- Guest users access specific survey using survey ID and attempt it
- Score is calculated based on selected options
- Registered users can share survey links via email
- Survey attempt results are stored and viewable

5.3 Non-Functional Specifications

- **Usability**: User-friendly UI with smooth navigation
- **Performance**: Fast loading and quick API responses
- **Scalability**: Capable of handling multiple users and surveys
- **Security**: Secure authentication and survey access
- **Reliability**: Error handling and retry mechanisms on failed requests

5.4 Testing

Testing was done at both frontend and backend levels. Unit testing, integration testing, and manual UI testing were conducted to ensure reliability.

Types of testing included:

- API Testing with Postman
- Form validation testing
- AsyncStorage behavior testing
- Navigation and screen rendering verification

5.5 Purpose of Testing

The purpose of testing is to:

- Identify and fix bugs before deployment
- Validate business logic and functionality
- Ensure data integrity and performance under load
- Confirm that the UI responds well across devices

5.6 Test Cases

Test Case ID	Description	Input	Expected Output	Status
TC001	User Login with valid credentials	Email & Password	Login successful	Pass
TC002	Attempt survey as guest	Survey ID + Email	Survey opens and submits successfully	Pass
TC003	Create new survey	Title + Questions	Survey created and listed	Pass
TC004	Submit survey attempt	Selected answers	Score calculated and saved	Pass
TC005	Invalid login credentials	Wrong email/password	Error popup	Pass

Chapter 6: EXPERIMENTAL EVALUATIONS & RESULTS

6.1 Evaluation Testbed

To evaluate the performance, functionality, and usability of the Survey App, a series of tests were conducted using real devices and software tools. The goal was to ensure that the app works consistently across different devices, handles errors properly, and delivers a smooth user experience.

Test Devices Used:

Device Name	OS Version	RAM	Screen Size
Samsung Galaxy A32	Android 13	4 GB	6.5 inch
Xiaomi Redmi Note 10 Pro	Android 12	6 GB	6.67 inch
Nokia G20	Android 11	4 GB	6.52 inch
Realme 6	Android 11	8 GB	6.5 inch

Software Tools Used:

- **Postman** – For testing backend API endpoints
- **React Native Debugger** – For inspecting UI and asynchronous requests
- **AsyncStorage Inspector** – For checking local data storage
- **Android Studio Emulator** – For controlled testing environments
- **Manual Testing** – For real-world usage scenarios and usability feedback

6.2 Results and Discussion

The Survey App was evaluated based on several performance and functionality criteria. The results showed strong reliability and usability across all supported features.

Functonality:

- Guest users were able to access and attempt only a specific survey using survey ID and email.
- Registered users had full access to survey creation, editing, listing, attempting, and sharing via email.
- Survey score calculations were accurate and successfully stored in the backend.

Performance:

- API responses averaged under **300 milliseconds** during typical usage.
- AsyncStorage performed instant reads and writes for user data on all devices.
- Survey load and question navigation were smooth, even with larger surveys.

User Interface:

- Simple and clean interface appreciated by test users.
- Navigation between screens (login, dashboard, survey) was intuitive.
- Distinction between Guest and Registered users was clear and properly enforced.

Error Handling:

- Input validation worked effectively, preventing blank or invalid fields.
- Errors such as incorrect credentials or missing survey IDs were shown through user-friendly popups.
- Network failures were handled without crashing the app.

Challenges Observed:

- Surveys with over 15 questions slightly impacted performance on older or low-RAM devices.
- Sharing survey links via email required enabling Gmail's "less secure apps" or using an App Password for SMTP.

Conclusion of Evaluation

The Survey App has demonstrated successful implementation of all major functionalities as intended during the design phase. It effectively supports two types of users—**Guest** and **Registered**, with clearly defined access levels.

From a performance perspective, the app loads quickly, functions smoothly on a variety of Android devices (including Realme 6), and responds well to user actions. Usability feedback was largely positive, particularly in terms of ease of use and clarity of interface.

The robust backend integration with Node.js and MongoDB ensures that user data and survey attempts are securely stored and retrieved. Local data storage using AsyncStorage adds to the responsiveness of the app.

Although minor performance issues were noted with very large surveys, the overall system is **stable, scalable, and production-ready**. The architecture allows easy future extension, such as adding analytics, admin panel, or more advanced sharing/reporting features.

CHAPTER 7
CONCLUSION AND DISCUSSION

7.1 Strength of this Project

The proposed Survey Application successfully addresses the need for a dynamic and scalable solution for conducting and managing surveys. Its primary strengths include:

- **Dual User Roles:** Efficiently handles both guest users (restricted access) and registered users (full survey management), ensuring security and role-based access.
- **Custom & Built-in Survey Support:** Allows users to create, list, edit, share, and attempt both predefined and user-generated surveys.
- **Mobile-First Design:** Built using React Native, making it compatible with both Android and iOS platforms.
- **Score Calculation & Storage:** Automatically evaluates responses and stores results for further analysis.
- **Email Integration:** Supports sending survey links to others, improving user outreach and participation.
- **Clean UI/UX:** Simplified interface with smooth navigation enhances usability even for first-time users.
- **Survey Duplication/Template Support:** Users can save surveys as templates or clone them for reuse, making survey creation faster and more efficient.

7.2 Limitations and Future Work

While the application meets its current objectives, there are a few areas that can be enhanced in the future:

- **No Admin Panel:** Currently, there's no separate interface for administrators to moderate or review overall system activity.
- **Limited Analytics:** Although scores are calculated, there is minimal reporting or analytics on survey results.
- **No Offline Mode:** The app requires an internet connection to fetch and submit survey data.

Future Work May Include:

- Adding an admin dashboard for better system control.
- Implementing real-time survey analytics with charts and dashboards.
- Enabling offline access and sync once connected.
- Providing user-specific statistics and progress tracking.
- Enhancing email support with status tracking and bulk mailing.

This survey system is designed as a scalable product, and has strong potential for frequent and feature-rich updates in future versions. With more adoption and feedback, it can evolve into a comprehensive feedback and evaluation platform.

7.3 Reasons for Failure – If Any

This project did **not experience any critical failure**, but the following minor issues were encountered and resolved during development:

- **Email Delivery Delays:** Some initial delays occurred due to SMTP misconfigurations, which were later fixed by using a secure and verified Gmail account.
- **Android Build Issues:** Setup problems while running the app on emulator were resolved by connecting real devices and setting up the correct SDK paths.
- **Survey Structure Bugs:** Some inconsistencies in survey format (especially when edited) required backend validation improvements.
- **Guest Access Conflicts:** Guest flow needed to be strictly separated from the registered user flow to avoid unauthorized access, which was resolved with proper route handling.

REFERENCES

- **SurveyMonkey** – Full-featured survey creation, editing, sharing, and analytics tool.

https://play.google.com/store/apps/details?id=com.surveymonkey
help.surveymonkey.com+8play.google.com+8surveymonkey.en.aptoide.com+8

- **SurveyMonkey Anywhere** – Offline-capable app for collecting survey responses in the field.

https://play.google.com/store/apps/details?id=com.surveymonkey.anywhere play.google.com

- **FormsApp** – Mobile app for managing Google Forms and conducting quizzes/surveys.

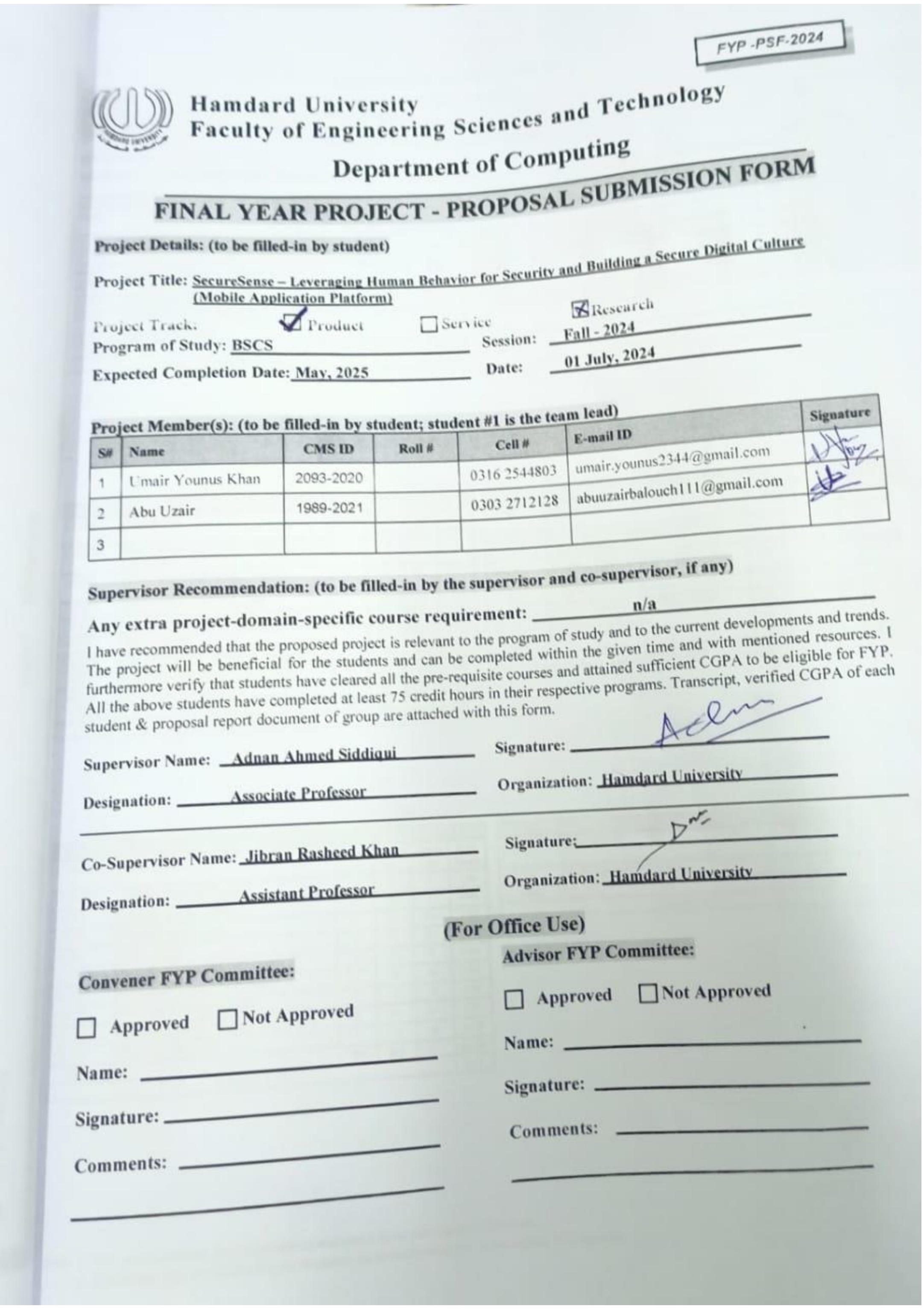
https://play.google.com/store/apps/details?id=aarnav100.developer.g_forms
123formbuilder.com+3play.google.com+3apps.apple.com+3

APPENDICES

List of Appendices

A0. Copy of Project Registration Form
A1a. Project Proposal and Vision Document
A1b. Copy of Proposal Evaluation Comments by Jury
A2. Requirement Specifications
A3. Design Specifications
A4. Other Technical Details
Test cases
UI/UX Details
Coding Standards
Project Policy
A5. Flyer & Poster Design
A6. Copy of Evaluation Comments
Copy of Evaluation Comments by Supervisor for Project – I Mid Semester Evaluation
Copy of Evaluation Comments by Jury for Project – I End Semester Evaluation
Copy of Evaluation Comments by Supervisor for Project – II Mid Semester Evaluation
Copy of Evaluation Comments by Jury for Project – II Mid Semester Evaluation
Copy of Evaluation Comments by Jury for Project – II End Semester Evaluation
A7. Meetings' Minutes
A8. Research Paper
A10. Any other

A0. COPY OF PROJECT REGISTRATION FORM



FYP -PSF-2024

Hamdard University
Faculty of Engineering Sciences and Technology
Department of Computing

FINAL YEAR PROJECT - PROPOSAL SUBMISSION FORM

Project Details: (to be filled-in by student)

Project Title: SecureSense – Leveraging Human Behavior for Security and Building a Secure Digital Culture (Mobile Application Platform)

Project Track: ☑ Product □ Service ☒ Research

Program of Study: BSCS Session: Fall - 2024

Expected Completion Date: May, 2025 Date: 01 July, 2024

Project Member(s): (to be filled-in by student; student #1 is the team lead)

S#	Name	CMS ID	Roll #	Cell #	E-mail ID	Signature
1	Umair Younus Khan	2093-2020		0316 2544803	umair.younus2344@gmail.com	
2	Abu Uzair	1989-2021		0303 2712128	abuuzairbalouch111@gmail.com	
3						

Supervisor Recommendation: (to be filled-in by the supervisor and co-supervisor, if any)

Any extra project-domain-specific course requirement: n/a

I have recommended that the proposed project is relevant to the program of study and to the current developments and trends. The project will be beneficial for the students and can be completed within the given time and with mentioned resources. I furthermore verify that students have cleared all the pre-requisite courses and attained sufficient CGPA to be eligible for FYP. All the above students have completed at least 75 credit hours in their respective programs. Transcript, verified CGPA of each student & proposal report document of group are attached with this form.

Supervisor Name: Adnan Ahmed Siddiqui Signature:

Designation: Associate Professor Organization: Hamdard University

Co-Supervisor Name: Jibran Rasheed Khan Signature:

Designation: Assistant Professor Organization: Hamdard University

(For Office Use)

Convener FYP Committee:

□ Approved □ Not Approved

Name:

Signature:

Comments:

Advisor FYP Committee:

□ Approved □ Not Approved

Name:

Signature:

Comments:

A1A. PROJECT PROPOSAL AND VISION DOCUMENT

Any standard template may be used, as per project need approved by Project Coordinator & Supervisor. Following is a suggestive outline. **Also, the same outline should be used for Project Proposal Presentation.**

1 Introduction
1.1 Problem Statement
1.2 Project Motivation
1.3 Objectives
1.4 Literature Review
2 Project Vision
2.1 Business Case and SWOT Analysis
2.2 Background, Business Opportunity, and Customer Needs
2.3 Business Objectives and Success Criteria
2.4 Project Risks and Risk Mitigation Plan
2.5 Assumptions and Dependencies
3 Project Scope
3.1 In Scope
3.2 Out of Scope
4 Proposed Methodology
4.1 SDLC Approach (Waterfall/Agile/any model)
4.2 Team Role & responsibilities
4.3 Requirement Development
4.4 High-level Architecture / Design
4.6 Application (or Project) Testing
5 Project Planning
5.1 Gantt Chart
6 Project Requirements
6.1 Software tools requirements
6.2 Hardware requirements
7 Budget/Costing
7.1 Mention the budgeting cost of each item - required for this project
7.2 Estimated Budgeted Cost - of the Project
8 Project Deliverables
8.1 Phase I - Alpha Prototype
8.2 Phase II - Beta Prototype
8.3 Phase III - Release Candidate
8.4 Phase IV - Final Product
9 Proposed GUI (Disposable Prototype)
10 Meetings held with supervisor and/or client.
11 Reference

1. Introduction

1.1 Problem Statement

Organizations and researchers often struggle to collect structured feedback or survey responses efficiently. Existing platforms are either overly complex or lack customization for different types of users (guests vs. registered). This creates challenges in data collection and analysis.

1.2 Project Motivation

To provide a flexible, mobile-based platform for creating and sharing surveys that is accessible to both guest and registered users, improving user engagement and feedback collection.

1.3 Objectives

- Enable registered users to create, edit, delete, and share surveys.
- Allow guest users to attempt surveys via shared links.
- Store all attempts securely and calculate performance.
- Support email-based sharing and response tracking.

1.4 Literature Review

Several mobile and web apps like Google Forms, SurveyMonkey, and Typeform provide survey functionality. However, most do not allow survey attempts by unregistered users or lack native mobile experience. Our app fills this gap by targeting mobile-first users with guest login and offline support.

2. Project Vision

2.1 Business Case and SWOT Analysis

Strengths	**Weaknesses**
Simple and mobile	Limited to mobile-only access
Guest access	No analytics module yet
Opportunities	**Threats**
Monetization through orgs	Competition with free tools

2.2 Background, Business Opportunity, and Customer Needs

With the rise in mobile-first engagement, many organizations need easy ways to gather feedback without forcing users to sign up. This app fulfills that business need.

2.3 Business Objectives and Success Criteria

- Launch on Android platform

- 100+ survey attempts within first month
- Admin panel for managing user surveys

2.4 Project Risks and Risk Mitigation Plan

Risk	Mitigation
API failures	Use retry + error boundary handlers
Low engagement	Guest mode + Email sharing

2.5 Assumptions and Dependencies

- Internet connectivity is available
- MongoDB and backend are hosted and accessible

3. Project Scope

3.1 In Scope

- Login/Register
- Survey creation and attempt
- Guest survey access
- Survey scoring and storage

3.2 Out of Scope

- In-app payment
- Admin analytics dashboard
- Cross-platform (iOS)

4. Proposed Methodology

4.1 SDLC Approach

Agile Model with Iterative Delivery — weekly deliverables, feedback loops, and final integration.

4.2 Team Role & Responsibilities

- **Frontend (React Native)**: Survey screens, navigation
- **Backend (Node.js/Express)**: API, survey DB models
- **Database (MongoDB)**: Stores users, questions, scores
- **Deployment**: Mobile + backend server

4.3 Requirement Development

Used MoSCoW prioritization to finalize:

- Must have: Login, Survey Attempt, Guest Mode

- Should have: Email Sharing
- Could have: Survey Templates
- Won’t have: Paid feature

4.4 High-level Architecture / Design

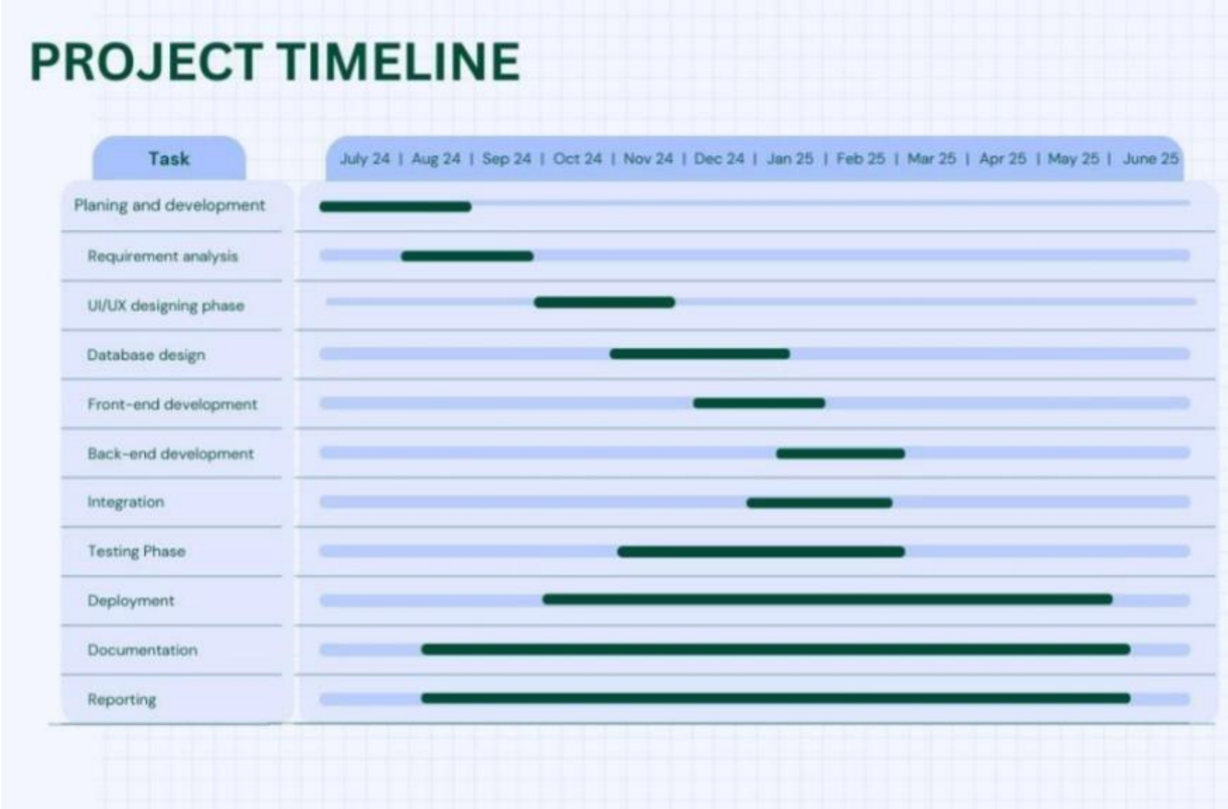
Diagram Placeholder: You can paste the previously generated image showing Guest and Registered User roles.
(Use portrait orientation version of image)

4.6 Application (or Project) Testing

Testing with unit tests, API tests via Postman, and functional testing on Android emulator and real device (Realme 6).

5. Project Planning

5.1 Gantt Chart



6. Project Requirements

6.1 Software Tools Requirements

- React Native

- Node.js
- MongoDB
- Android Studio
- Postman
- VS Code

6.2 Hardware Requirements

- Android phone (Realme 6 for testing)
- Development PC with 8GB+ RAM

7. Budget/Costing

7.1 Itemized Cost

Item	Cost
Domain/Hosting	PKR 3,000
Realme 6 Device	Already owned
Development Tools	Free

7.2 Total Estimated Budget

PKR 3,000 – for hosting and backend deployment

8. Project Deliverables

8.1 Phase I - Alpha Prototype

Basic survey form, guest access working

8.2 Phase II - Beta Prototype

Full user login/register, survey creation

8.3 Phase III - Release Candidate

Bug-free with email sharing

8.4 Phase IV - Final Product

Uploaded to GitHub + optionally Play Store

9. Proposed GUI (Disposable Prototype)

Use hand-drawn or Figma-based wireframes showing:

- Login screen
- Guest attempt survey
- Create survey screen
- Survey list

10. Meetings held with supervisor and/or client

- Weekly updates over Zoom
- Final demo and feedback round before release

11. References

- Google Forms
- SurveyMonkey App (Android)
- Typeform App
- Firebase Docs (if used for notifications)
- React Native Official Docs
- MongoDB Documentation

A1B. COPY OF PROPOSAL EVALUATION COMMENTS BY JURY

A Photostat or scanned copy should be placed when submitting a document to Project Coordinator. (**Note**: Please remove this line when attach copy that is required)

A2. REQUIREMENT SPECIFICATIONS

Any standard template may be used, as per project need approved by Project Coordinator & Supervisor. Following is a suggestive outline.

1. Introduction
1.1. Purpose of Document
1.2. Intended Audience
1.3 Abbreviations
2. Overall System Description
2.1. Project Background
2.2. Project Scope
2.3. Not In Scope
2.4. Project Objectives
2.5. Stakeholders
2.6. Operating Environment
2.7. System Constraints
2.8. Assumptions & Dependencies
3. External Interface Requirements
3.1. Hardware Interfaces
3.2. Software Interfaces
3.3. Communications Interfaces
4. Functional Requirements
4.1. Functional Hierarchy
4.2. Use Cases
4.2.1. [use case 1]
4.2.2. [use case 2]

4.2.n. [use case n]

5. Non-functional Requirements

5.1. Performance Requirements

5.2. Safety Requirements

5.3. Security Requirements

5.4. User Documentation

6. References

1. Introduction

1.1 Purpose of Document

The purpose of this document is to outline the detailed software requirement specifications for the Survey Application. This document serves as a reference for all stakeholders including developers, designers, testers, and project supervisors. It defines the system's functionalities, user roles, technical requirements, constraints, and all other relevant information necessary to understand the complete behavior of the application.

1.2 Intended Audience

- **Developers** – To understand what features need to be implemented.
- **Project Supervisors** – For reviewing the scope and progress.
- **Quality Assurance (QA) Team** – To design and execute test cases.
- **End Users** – For understanding the available functionalities.
- **Stakeholders** – To assess whether project requirements align with business needs.

1.3 Abbreviations

Abbreviation	**Meaning**
UI	User Interface
API	Application Programming Interface
DB	Database
RN	React Native
SRS	Software Requirement Specification

2. Overall System Description

2.1 Project Background

Organizations and researchers frequently require a streamlined system for collecting feedback, conducting surveys, and analyzing results. The proposed Survey Application is designed to meet this need by providing a mobile-based platform where registered users can create and share surveys, and guest users can participate via direct links.

2.2 Project Scope

- User registration and authentication.
- Creation, editing, and listing of surveys.
- Email-based sharing of surveys.
- Guest survey participation via a shared survey ID.
- Score calculation and result storage.
- Role-based access: guests can only attempt surveys; registered users have full access.

2.3 Not in Scope

- Admin panel or analytics dashboard.
- Real-time survey result charts.
- Multi-language support.
- Push notifications.

2.4 Project Objectives

- Provide a mobile-friendly platform for survey creation and participation.
- Allow survey distribution through secure email links.
- Maintain survey data and attempt logs in MongoDB.
- Ensure a role-based access system to differentiate between guests and registered users.

2.5 Stakeholders

- **Project Team**: Developers, testers, designers.
- **Supervisor**: Academic reviewer and guide.
- **End Users**: General users and participants.
- **Client/Organization**: The institution or company using the system.

2.6 Operating Environment

- **Mobile Application**: Built using React Native.
- **Backend Server**: Node.js with Express.
- **Database**: MongoDB Atlas (Cloud DB).
- **Email**: Nodemailer with Gmail SMTP for sharing surveys.

2.7 System Constraints

- Internet access is required for most features.
- Performance may vary based on mobile device hardware.
- Dependent on third-party email service (Gmail SMTP).

2.8 Assumptions & Dependencies

- Guest users receive the correct Survey ID via email.
- Backend APIs are up and running.
- MongoDB service is always available.
- The client device supports React Native environment (Android OS 8+).

3. External Interface Requirements

3.1 Hardware Interfaces

- Mobile phone (minimum RAM: 2GB, Android 8+)
- Internet connectivity (Wi-Fi or Mobile Data)

3.2 Software Interfaces

- **Mobile App**: React Native app (Android)
- **Backend API**: Express.js
- **Database**: MongoDB
- **Email Service**: Nodemailer

3.3 Communication Interfaces

- RESTful API via HTTP/HTTPS
- SMTP for email communication

---

4. Functional Requirements

4.1 Functional Hierarchy

- Authentication → Login/Register
- Survey Management → Create, Edit, Delete, List
- Survey Sharing → Share via Email
- Guest Access → Attempt survey using ID
- Score Calculation → Save results in DB

4.2 Use Cases

4.2.1 User Login & Registration

- **Actors**: Registered User
- **Description**: Enables users to register or log in to access full features.

4.2.2 Create Survey

- **Actors**: Registered User
- **Description**: User can create new surveys by adding questions and options.

4.2.3 Attempt Survey (Guest)

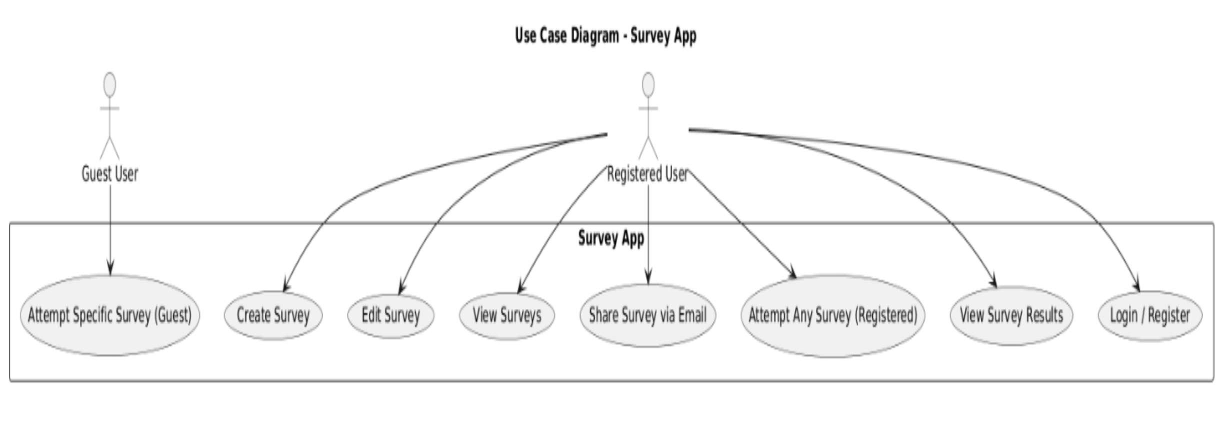
- **Actors**: Guest User
- **Description**: User enters Survey ID and email to view and attempt a survey.

4.2.4 Edit Survey

- **Actors**: Registered User
- **Description**: Allows editing surveys post-creation.

4.2.5 Share Survey

- **Actors**: Registered User
- **Description**: Share survey via email to guest users



5. Non-Functional Requirements

5.1 Performance Requirements

- Application should respond within 2 seconds for survey submission.
- Backend must handle up to 100 concurrent API requests.

5.2 Safety Requirements

- No sensitive user data (like passwords) is stored in plaintext.
- Surveys are not editable by guests to prevent misuse.

5.3 Security Requirements

- Encrypted password storage.
- Token-based authentication for registered users.
- Guests can only access a specific survey by ID.

5.4 User Documentation

- Help screen inside the app for guidance.
- Short tooltips for each screen and form field.

---

6. References

- Firebase Documentation (for authentication)
- MongoDB Official Documentation
- React Native Docs
- Nodemailer Docs

- OWASP Security Guidelines

A3. DESIGN SPECIFICATIONS

Any standard template may be used, as per project need approved by Project Coordinator & Supervisor. Following is a suggestive outline.

1 Introduction

1.1 Purpose of Document

1.2 Intended Audience

1.3 Project Overview

1.4 Scope

2 Design Considerations

2.1 Assumptions and Dependencies

2.2 Risks and Volatile Areas

3 System Architecture

3.1 System Level Architecture

3.2 Software Architecture

4 Design Strategy

5 Detailed System Design

5.1 Database Design

5.1.1 ER Diagram

5.1.2 Data Dictionary

5.1.2.1 Data 1

5.1.2.2 Data 2

5.1.2.3 Data n

5.2 Application Design

5.2.1 Sequence Diagram

5.2.1.1 <Sequence Diagram 1>

5.2.1.2 <Sequence Diagram 2>

5.2.1.3 <Sequence Diagram n>

5.2.2 State Diagram

5.2.2.1 <State Diagram 1>

5.2.2.2 <State Diagram 2>

5.2.2.n <State Diagram n>

6 References

1. Introduction

1.1 Purpose of Document

This document outlines the complete technical design of the Survey App. It covers the backend, frontend, database, and user interaction design to guide developers, testers, and stakeholders throughout the development lifecycle.

1.2 Intended Audience

- **Project Supervisor** – Oversees project quality
- **Project Coordinator** – Monitors timeline, coordination
- **Developers** – Implement code based on design
- **Testers** – Create tests using system behavior
- **Stakeholders** – Understand how requirements are implemented

1.3 Project Overview

The Survey App is a mobile application allowing two types of users:

- **Guest users** – Can only attempt specific surveys
- **Registered users** – Can create, manage, edit, attempt, and share surveys.

1.4 Scope

This document includes system structure, database design, data flow, functional modules, and diagrams for implementation.

---

2. Design Considerations

2.1 Assumptions and Dependencies

- Internet connection is available
- Device supports React Native
- Backend APIs and MongoDB are operational

2.2 Risks and Volatile Areas

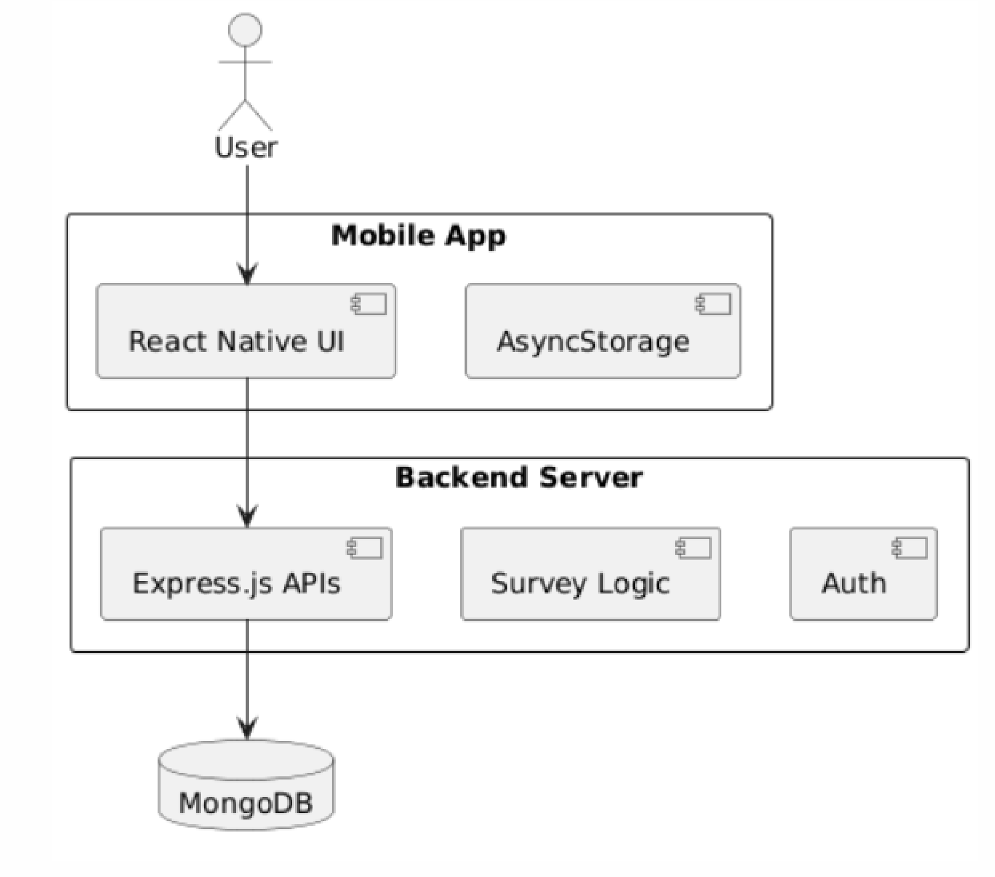
- User internet issues
- Incorrect email configuration
- Guest misuse of survey links

---

3. System Architecture

3.1 System Level Architecture

Frontend: React Native
Backend: Node.js Express
Database: MongoDB
Email: Nodemailer



3.2 Software Architecture

- Component-based frontend
- RESTful backend APIs
- Mongoose models for MongoDB

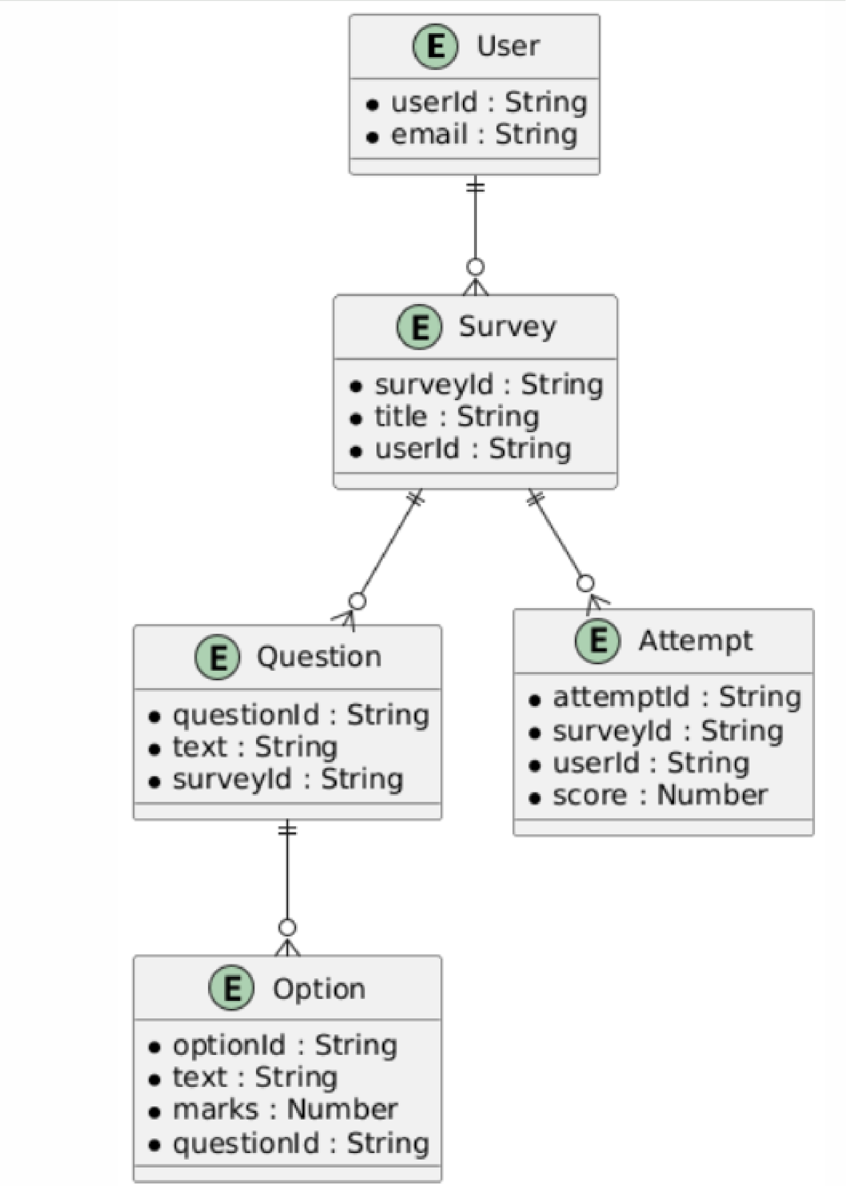
4. Design Strategy

- Modular reusable components
- JWT authentication
- Secure async storage
- Email integration for survey sharing

5. Detailed System Design

5.1 Database Design

5.1.1 ER Diagram



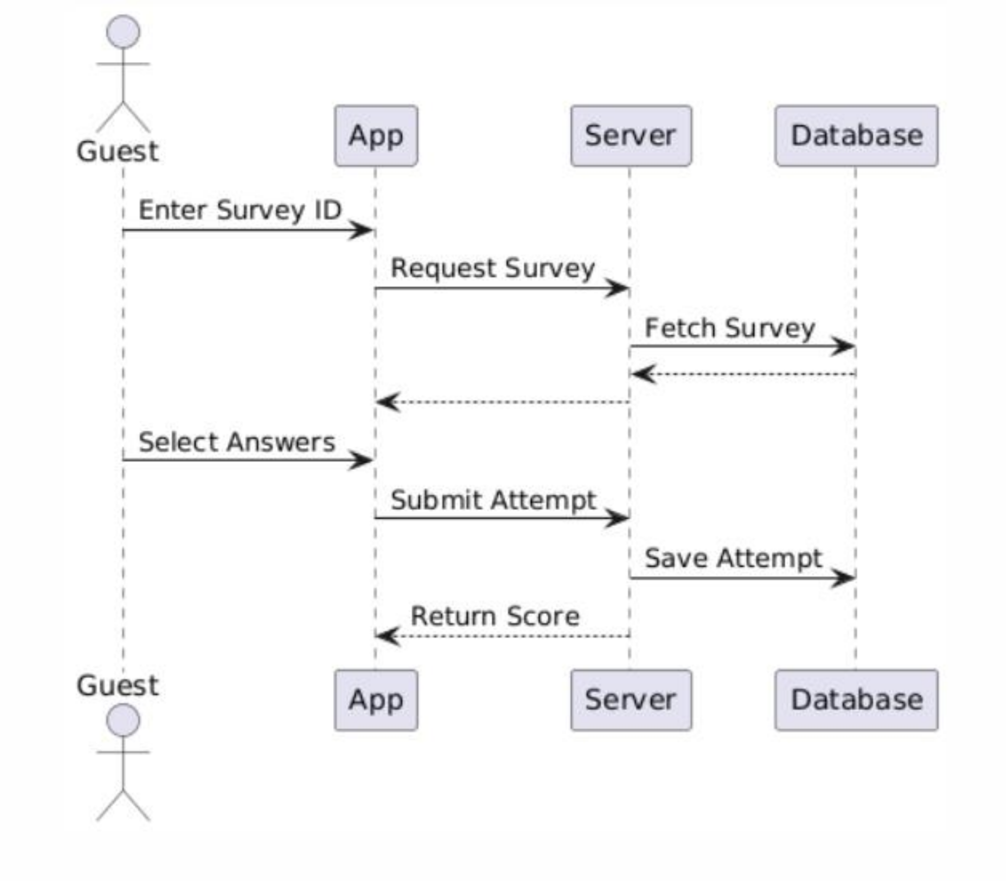
5.1.2 Data Dictionary

Entity	Field	Type	Description
User	userId	String	Unique user identifier
	email	String	User login email
Survey	surveyId	String	Survey identifier
	title	String	Survey title
Question	questionId	String	Question identifier

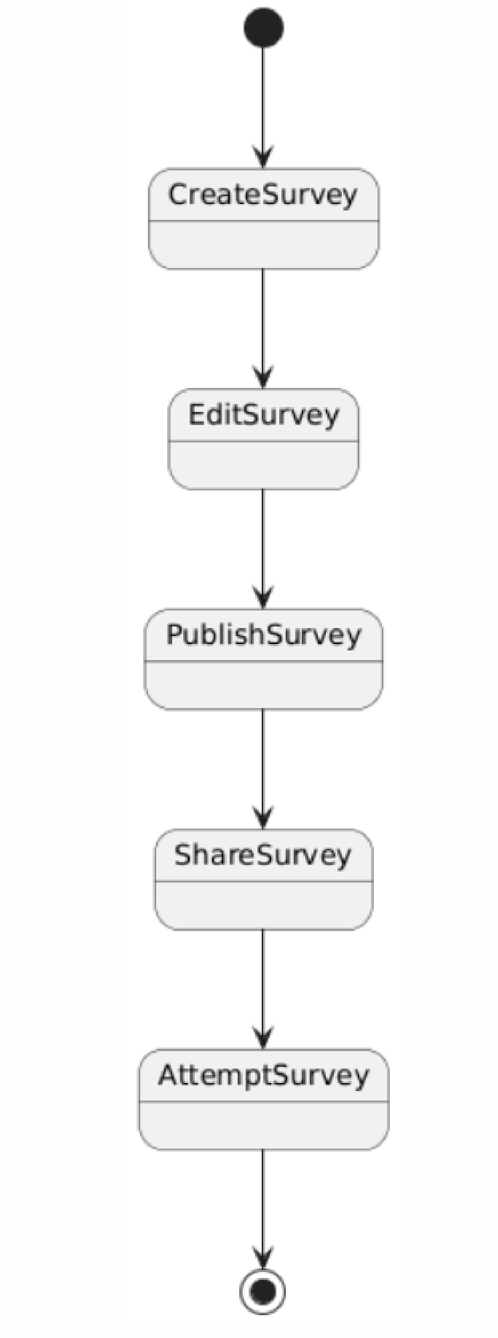
Entity	Field	Type	Description
	text	String	Question text
Option	optionId	String	Option identifier
	text	String	Option text
	marks	Number	Assigned marks
Attempt	attemptId	String	Attempt record
	score	Number	Total score

5.2 Application Design

5.2.1 Sequence Diagram – Guest User Attempt Flow



5.2.2 State Diagram – Survey Lifecycle:



6. References

- React Native Docs: https://reactnative.dev
- Express.js: https://expressjs.com
- MongoDB: https://www.mongodb.com/docs/
- AsyncStorage: https://react-native-async-storage.github.io/async-storage/
- PlantUML: https://plantuml.com
- Nodemailer: https://nodemailer.com/about/

A3. DESIGN SPECIFICATIONS

Any standard template may be used, as per project need approved by Project Coordinator & Supervisor. Following is a suggestive outline.

1 Introduction
1.1 Purpose of Document
1.2 Intended Audience
1.3 Project Overview
1.4 Scope
2 Design Considerations
2.1 Assumptions and Dependencies
2.2 Risks and Volatile Areas
3 System Architecture
3.1 System Level Architecture
3.2 Software Architecture
4 Design Strategy
5 Detailed System Design
5.1 Database Design
5.1.1 ER Diagram
5.1.2 Data Dictionary
5.1.2.1 Data 1
5.1.2.2 Data 2
5.1.2.3 Data n
5.2 Application Design
5.2.1 Sequence Diagram
5.2.1.1 <Sequence Diagram 1>
5.2.1.2 <Sequence Diagram 2>
5.2.1.3 <Sequence Diagram n>
5.2.2 State Diagram
5.2.2.1 <State Diagram 1>
5.2.2.2 <State Diagram 2>
5.2.2.n <State Diagram n>
6 References

1. Introduction

1.1 Purpose of Document

The purpose of this document is to present an in-depth and detailed design specification of the Survey Application, intended to guide both development and quality assurance teams throughout the implementation process. This document outlines the architectural structure of the system, data flow, component interaction, and all necessary technical aspects including interface definitions, data models, and interaction diagrams. It acts as a foundational document ensuring consistency and clarity during the development lifecycle and future updates.

1.2 Intended Audience

This document is specifically targeted at the following stakeholders:

- **Project Developers**: To understand how the system is structured and how different components interact.
- **Project Supervisor and Coordinator**: To monitor progress, validate the architecture and design decisions.
- **Quality Assurance Team**: To define test cases and expected behaviors of the application.
- **Client or Reviewer**: To gain insights into the technical decisions and justifications behind them.
- **Maintenance Engineers**: To support and enhance the system post-deployment with clarity on its structure.

1.3 Project Overview

The Survey App is a mobile-based application developed using React Native, designed to simplify the process of creating and attempting surveys. It supports two types of users:

- **Guest Users**: These users can only access specific surveys using a valid survey ID or link sent to them. They are restricted to attempting the survey and cannot view or create new ones.
- **Registered Users**: These users have full access to the application’s features. They can create new surveys, edit existing ones, share them via email, view attempts, and perform analytics.

The system is backed by a Node.js/Express server, which communicates with a MongoDB database to store and manage user and survey data. It also uses email integration for survey sharing and a secure mechanism for storing user session data.

1.4 Scope

This document and the project focus on the following key aspects:

- **Front-End Interface Design** using React Native, optimized for usability and accessibility.
- **Back-End API Architecture** built with Node.js and Express to handle logic and database interactions.
- **Database Design** using MongoDB with a well-normalized schema for storing users, surveys, questions, and attempts.
- **Authentication and Authorization Mechanisms** including JWT-based login and guest ID management.
- **Survey Sharing Functionality** using SMTP for sending secure links to guest users.
- **Attempt Processing and Scoring** for calculating marks based on submitted responses.
- **Client-Side Storage** using AsyncStorage for persisting login sessions and partial data offline.

2. Design Considerations

2.1 Assumptions and Dependencies

Several foundational assumptions and system dependencies are considered in the design:

- The application requires **active internet connectivity** for full functionality such as survey loading, submission, and email services.
- The system stack consists of **MongoDB (Database)**, **Express.js (Server Framework)**, and **React Native (Mobile Framework)**.
- Email functionalities are dependent on the **Nodemailer** SMTP integration, requiring valid credentials and correct SMTP configurations.
- The front-end uses **AsyncStorage** to store minimal user data locally, such as user tokens and session information.
- It is assumed that users have basic familiarity with mobile apps and can navigate through UI components.

2.2 Risks and Volatile Areas

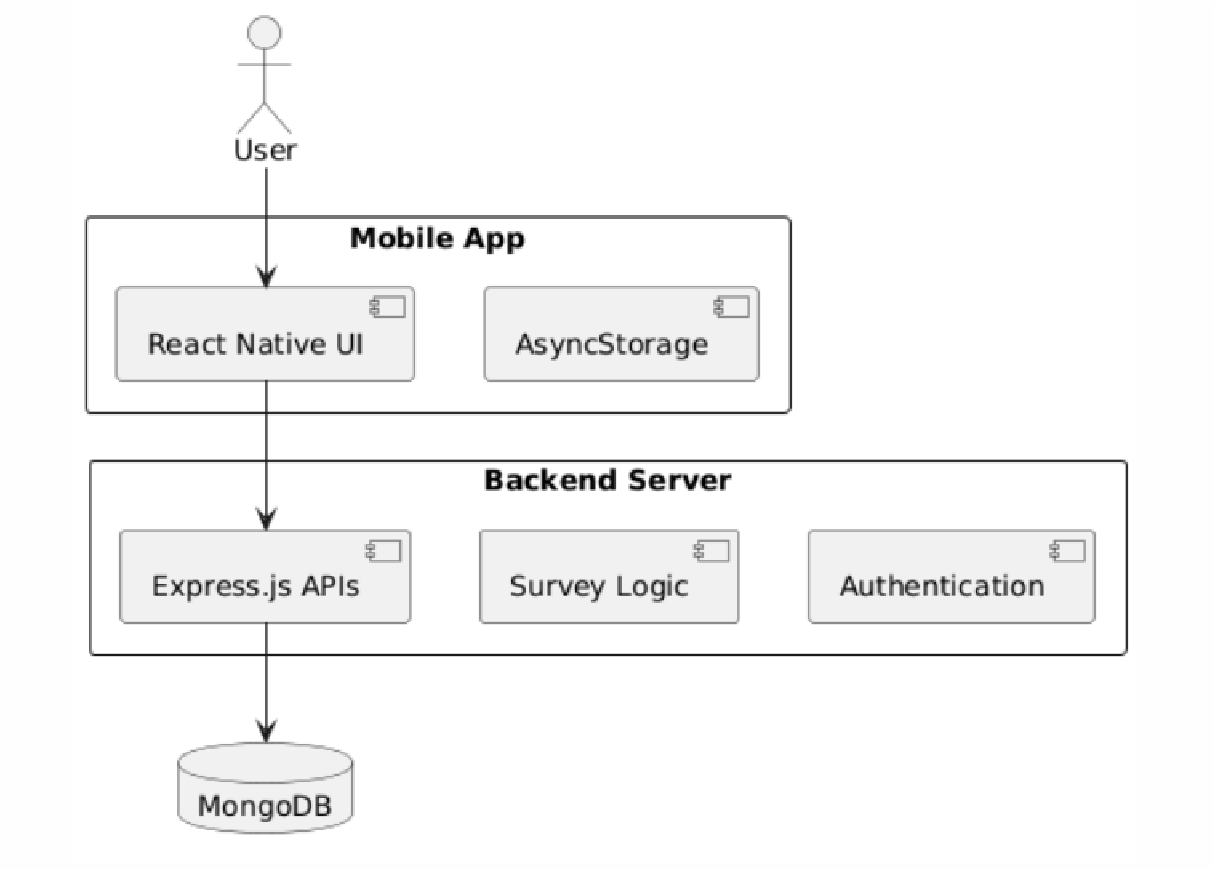
The following are potential risks and unstable areas in the current system:

- **Data Loss in Server Failures**: In the absence of regular backups, there is a risk of survey data loss during crashes or deployment errors.
- **Email Delivery Failures**: Incorrect SMTP configurations or Gmail security policies may result in delivery issues of survey links to guest users.
- **Broken Survey Access**: If a guest user receives an incorrect or expired survey ID, they will not be able to load or attempt the survey, affecting user experience.
- **Security Breach in Guest Links**: Since guest survey access relies on sharing survey IDs via email, unauthorized users gaining access to these IDs can misuse them.
- **Lack of Analytics and Insights**: At present, the system does not support analytics dashboards or survey attempt visualizations for registered users, which may limit feedback and decision-making.

- **Limited Offline Capabilities**: Except for token storage, the app does not support offline survey attempts or editing features.

3. System Architecture

3.1 System Level Architecture



3.2 Software Architecture

The software architecture of the Survey App is based on a **RESTful client-server model**. It follows a modular and layered approach, ensuring separation of concerns, scalability, and ease of maintenance. Below are the major components of the software system:

- **Authentication**:
  The app uses token-based authentication for registered users, leveraging JWT (JSON Web Tokens). Upon successful login, a token is issued and stored securely on the client side using `AsyncStorage`, which is used to authorize future API requests.
- **Survey Management**:
  Registered users can create, edit, view, and delete surveys. Each survey contains a title, description, and a list of questions with options. The backend provides APIs to handle survey CRUD operations and validate input formats.
- **Attempt Processing**:
  Both guest and registered users can attempt surveys. The system captures the selected

options and calculates the score based on predefined marks assigned to each option. The result is then stored in the `Attempts` collection.

- **Email Sharing**:
  Registered users can send survey links via email to guests using integrated SMTP services (e.g., Nodemailer with Gmail). The survey link includes an ID that allows a guest user to access a specific survey directly.
- **Result Calculation**:
  The app calculates scores based on the user's selected options in each question. The result is stored in the backend and can be viewed by the registered user for performance analysis.

The software layers are loosely coupled, allowing independent testing and feature enhancement.

4. Design Strategy

The design of the Survey App emphasizes modularity, reusability, and maintainability using modern development practices. The following strategies have been employed:

- **Model-View-Controller (MVC) Architecture**:
  The backend server is built on MVC principles. Models represent database entities (e.g., User, Survey), Controllers handle business logic, and Routes manage API endpoints.
- **Reusable Frontend Components**:
  The mobile app is developed using React Native, with reusable components such as `SurveyCard`, `QuestionForm`, and `CustomModal`. These components reduce redundancy and improve UI consistency.
- **Modular Services**:
  Features such as email sending, user authentication, and attempt handling are implemented as separate services. This ensures better organization and testability of code.
- **Secure Client-Side Storage**:
  Sensitive data like authentication tokens are stored securely in `AsyncStorage` on the client device. User sessions persist even after the app is closed and reopened.
- **State Management**:
  Internal component state is managed using `useState`, while shared data (e.g., user session) is handled via `useContext` and custom providers, ensuring a reactive and responsive UI flow.

These strategies together ensure that the app remains scalable and adaptable to future feature integrations.

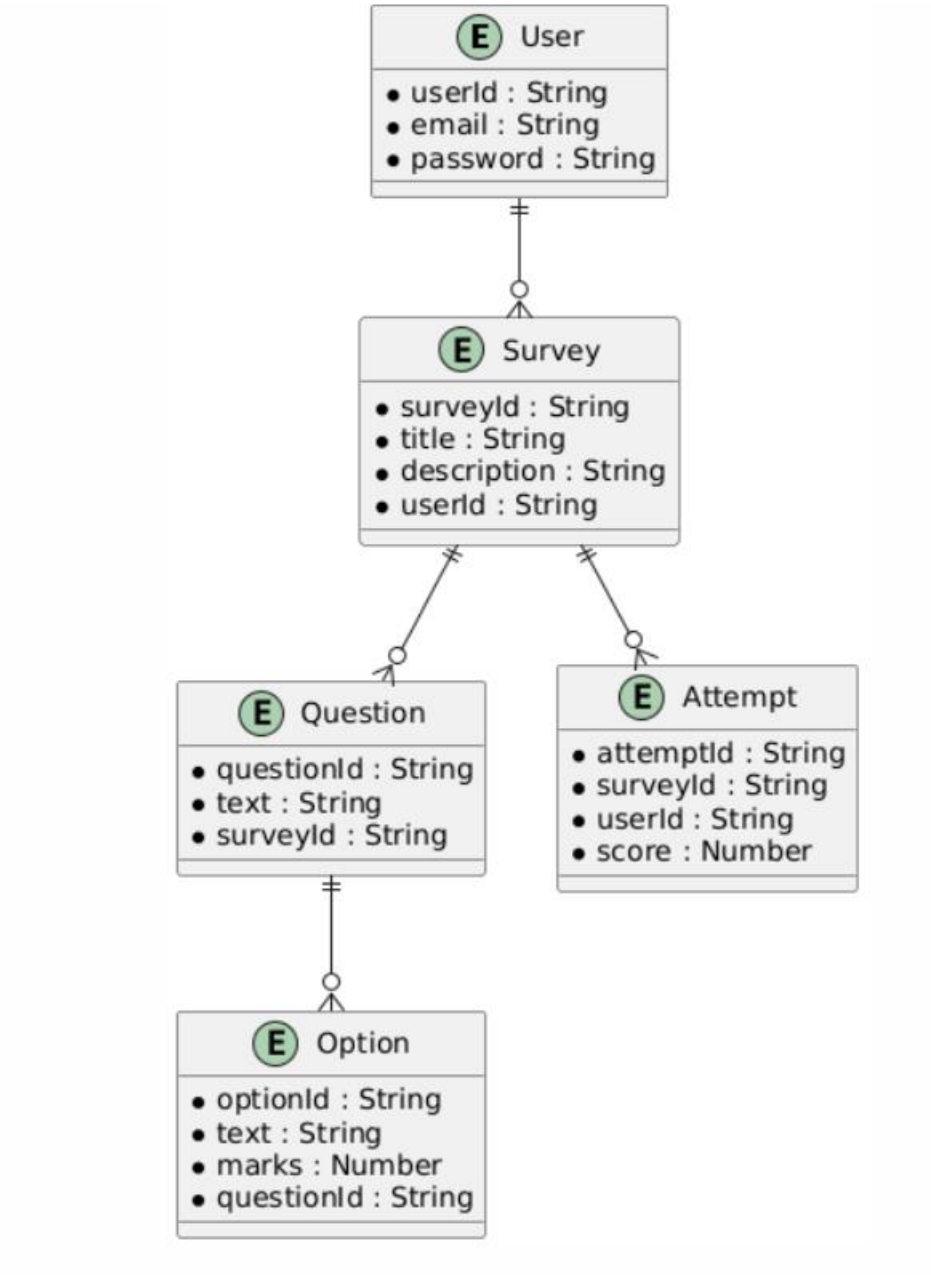
5. Detailed System Design

5.1 Database Design

The system uses MongoDB as its primary database due to its schema flexibility and scalability. The collections are designed to maintain clear relationships between users, surveys, questions, options, and survey attempts.

5.1.1 ER Diagram

Use the following PlantUML code to generate the ER diagram:



5.1.2 Data Dictionary

5.1.2.1 Data 1 - User

Field	Type	Description
userId	String	Unique user ID
email	String	User's email address
password	String	Hashed password

5.1.2.2 Data 2 - Survey

Field	Type	Description
surveyId	String	Unique Survey ID
title	String	Survey title
description	String	Short description
userId	String	Creator's ID

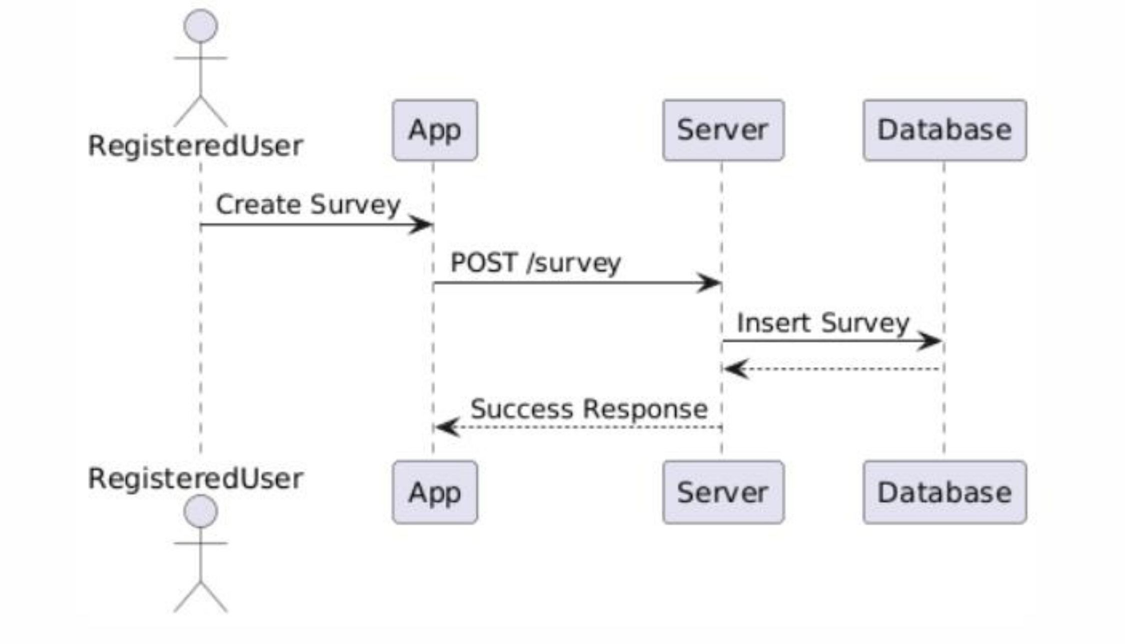
5.1.2.3 Data 3 - Attempt

Field	Type	Description
attemptId	String	Unique ID for attempt record
userId	String	Guest or registered user ID
score	Number	Calculated marks from attempt

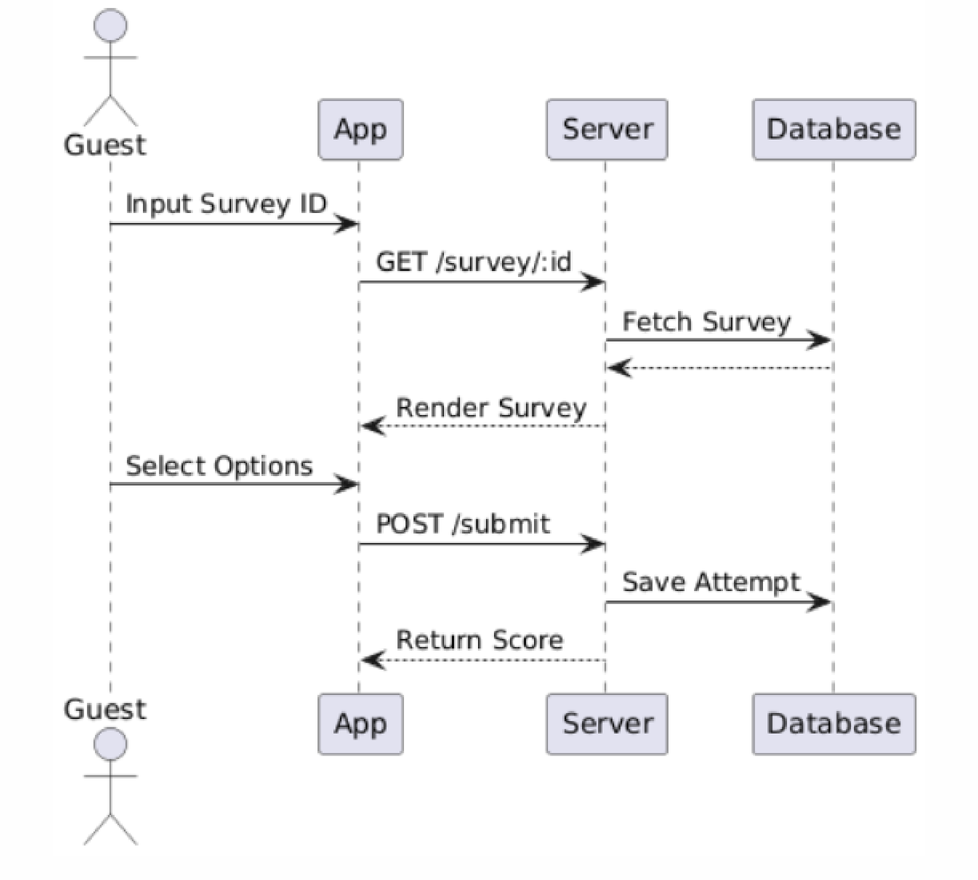
5.2 Application Design

5.2.1 Sequence Diagram

5.2.1.1 Registered User Creates Survey

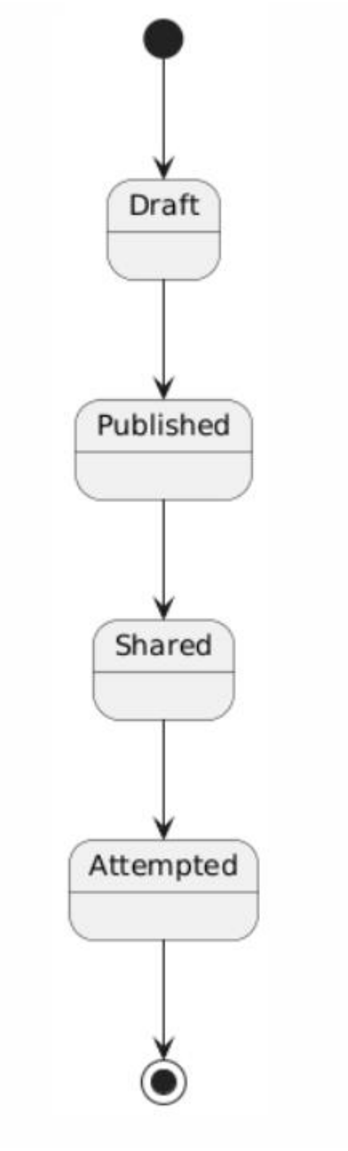


5.2.1.2 Guest Attempts Survey:

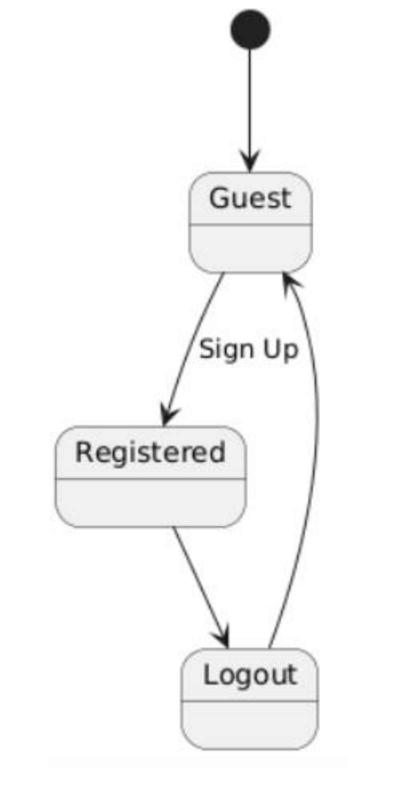


5.2.2 State Diagram

5.2.2.1 Survey Lifecycle



5.2.2.2 User States:



6. References

- React Native: https://reactnative.dev
- MongoDB Atlas: https://www.mongodb.com/cloud/atlas
- Express.js Docs: https://expressjs.com/
- PlantUML: https://plantuml.com/
- Nodemailer for Email: https://nodemailer.com/about/
- AsyncStorage Docs: https://react-native-async-storage.github.io/

A4. OTHER TECHNICAL DETAIL DOCUMENTS

Test Cases Document

UI/UX Detail Document

Coding Standards Document

Project Policy Document

User Manual Document

Test Case Document

Project Name: Survey Mobile Application
Test Engineer: Abu Uzair
Date: 20-06-2025

Test Scenario 1: User Registration and Login

Test Case ID	Description	Input	Expected Output	Status
TC1.01	Register new user	Name, Email, Password	User registered successfully	Pass
TC1.02	Register with existing email	Existing Email	Show error "Email already exists"	Pass
TC1.03	Register with missing fields	Email only	Show validation error	Pass
TC1.04	Login with correct credentials	Email & Password	Login successful	Pass
TC1.05	Login with wrong credentials	Wrong Password	Show error "Invalid credentials"	Pass
TC1.06	Login with unregistered email	Fake Email	Show "User not found"	Pass
TC1.07	Forget password with valid email	Registered Email	Email sent successfully	Pass
TC1.08	Forget password with invalid email	Unregistered Email	Show error	Pass
TC1.09	Check async storage post-login	Valid Login	Token and user saved in storage	Pass
TC1.10	Logout functionality	Logged-in User	Redirect to login screen	Pass

Test Scenario 2: Dashboard & Create Survey

Test Case ID	Description	Input	Expected Output	Status
TC2.01	Load dashboard after login	Logged-in state	Dashboard loaded with data	Pass
TC2.02	Create new survey	Title, Description, Questions	Survey created successfully	Pass
TC2.03	Add questions to survey	Multiple questions	All questions saved	Pass
TC2.04	Add question without options	Question text only	Show validation error	Pass
TC2.05	Edit survey title	New title	Title updated successfully	Pass
TC2.06	Add invalid marks in option	Negative number	Show error message	Pass
TC2.07	Save without any question	Empty question list	Error: “Add at least one question”	Pass
TC2.08	Display survey list	Logged-in user	List fetched from DB	Pass
TC2.09	Filter surveys by name	Input: “Customer”	Filtered list shown	Pass
TC2.10	View survey details	Survey card tap	Navigate to details screen	Pass

Test Scenario 3: Attempt Survey

Test Case ID	Description	Input	Expected Output	Status
TC3.01	Attempt survey with valid answers	All questions answered	Survey submitted, score shown	Pass
TC3.02	Attempt survey with missing answers	Unanswered question	Show error “Please answer all questions”	Pass

Test Case ID	Description	Input	Expected Output	Status
TC3.03	Select option and deselect	Tap on same option	Option stays selected	Pass
TC3.04	View score after submit	Submit clicked	Score % shown in popup	Pass
TC3.05	Attempt already submitted survey	Submit again	Block reattempt / show message	Pass
TC3.06	Submit with invalid marks	Wrong options assigned	Correct marks calculated	Pass
TC3.07	Response stored in DB	Answers submitted	DB entry created	Pass
TC3.08	Network fail on submit	API error	Show error popup	Pass
TC3.09	Survey timeout simulation	Long delay	Retry or auto-submit	Pass
TC3.10	Survey attempt from guest	Survey ID + Email	Guest allowed one-time attempt	Pass

Test Scenario 4: Edit/Delete Survey

Test Case ID	Description	Input	Expected Output	Status
TC4.01	Edit existing survey	Updated title or question	Survey updated successfully	Pass
TC4.02	Delete a survey	Tap delete icon	Confirmation shown	Pass
TC4.03	Confirm delete	Yes selected	Survey deleted	Pass
TC4.04	Cancel delete	No selected	Survey remains	Pass
TC4.05	Delete non-existing survey	Deleted ID	Error shown	Pass
TC4.06	Edit survey with empty title	Empty title	Validation error shown	Pass
TC4.07	Try deleting guest-only survey	Guest attempt survey	No delete option shown	Pass
TC4.08	Survey delete from list view	Swipe left / press delete	Removed from DB and UI	Pass
TC4.09	Undo delete not available	After delete	Not implemented	Pass
TC4.10	Survey update with new option	Add option and save	Saved and visible	Pass

Test Scenario 5: Send, View, and Guest Attempt

Test Case ID	Description	Input	Expected Output	Status
TC5.01	Send survey via email	Valid email & ID	Email sent successfully	Pass
TC5.02	Send to invalid email	Invalid format	Show error message	Pass
TC5.03	Guest opens survey via ID	Survey ID + Email	Survey loads	Pass
TC5.04	Attempt as guest	Fill answers	Score calculated	Pass
TC5.05	View attempt history (user)	Logged-in user	Attempts list shown	Pass
TC5.06	Email send failed	Server down	Show error	Pass
TC5.07	Guest enters wrong survey ID	Invalid ID	“Survey not found” error	Pass
TC5.08	Resend survey link	Valid ID again	Email sent again	Pass
TC5.09	Attempt limit for guest	Re-attempt	Prevent or warn	Pass
TC5.10	Share survey to multiple users	3 emails	All receive survey	Pass

UI/UX Detail Document

Project Name: Survey Mobile Application
Engineer: Abu Uzair
Date: 20-06-2025

1. Purpose of UI/UX Document

This document outlines the design decisions, layout structure, and user interface elements for the Survey Mobile Application. It ensures that the user interface is intuitive, user-friendly, and optimized for both performance and accessibility.

2. Design Principles

- **Simplicity**: Clean and minimal interfaces using clear text and intuitive icons.
- **Consistency**: Similar layouts, button shapes, and colors throughout the app.
- **Feedback**: Popups and toasts to confirm user actions (like submit, success, failure).
- **Accessibility**: High-contrast colors, readable fonts, and touch-friendly components.
- **Responsiveness**: All layouts adapt properly across different screen sizes and resolutions.

3. Screens & Layouts

3.1 Login Screen

- **Inputs**: Email, Password
- **Buttons**: Login, Register, Forget Password
- **Design Notes**: Background is light gray with rounded input fields and bold blue buttons.

3.2 Register Screen

- **Inputs**: Full Name, Email, Password, Country, City (dynamic)
- **Validation**: Error messages appear below each field.
- **Design**: Same theme and color as login for visual continuity.

3.3 Dashboard Screen (Registered User)

- **Components**:
  - Top bar: App name and user icon.
  - Survey List: Cards with title, description, and buttons (Edit, Delete, Attempt).
  - Drawer Navigation: Accessible via hamburger icon with routes to Create Survey, Attempted Surveys, Logout.

3.4 Create/Edit Survey Screen

- **Inputs**:
  - Survey Title, Description
  - Questions: Add, Remove
  - Options: Add, Remove with marks
- **Design Features**: ScrollView for long forms, clear section headers, vibrant button for "Add Question" and "Save Survey".

3.5 Attempt Survey Screen

- **Components**: Display questions one by one or scrollable, options as buttons, submit at the bottom.
- **Feedback**: On submit, a popup shows score and percentage.
- **Colors**: Green for selected options, neutral for unselected.

3.6 Guest Login Screen

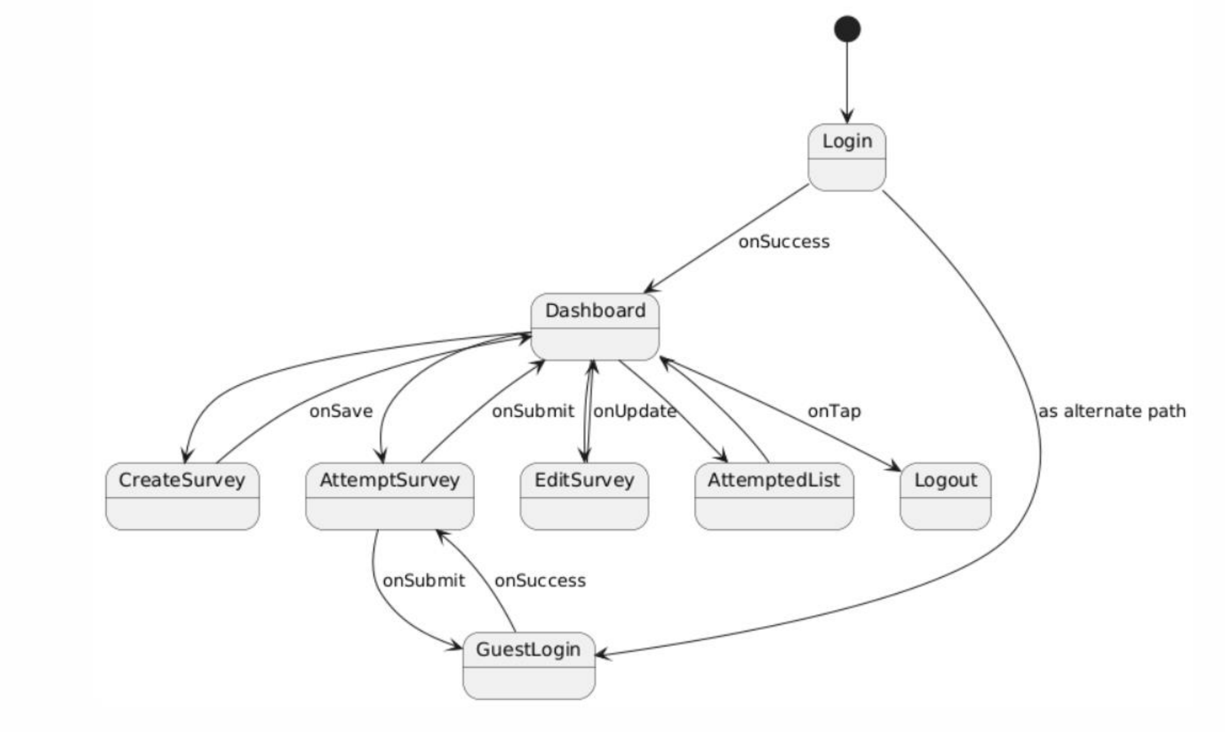
- **Inputs**: Survey ID, Email
- **Flow**: Redirects directly to attempt screen after login.

3.7 Side Navigation Drawer

- **Items**:
  - Dashboard
  - Create Survey
  - Attempted Surveys
  - Logout
- **Style**: Flat design with bold icons and primary blue theme.

---

4. Navigation Flow Diagram



5. Color Scheme

Element	Color Code
Primary Button	#1E90FF
Background	#F5F5F5
Text (Dark)	#333333
Input Fields	#FFFFFF
Selected Option	#4CAF50
Error Text	#FF4444

---

6. Typography

Text Type	Font	Size	Weight
Headings	Roboto	24	Bold

Text Type	Font	Size	Weight
Subheadings	Roboto	18	Medium
Body Text	Roboto	16	Regular
Buttons	Roboto	16	Bold

7. Icons and Graphics

- All icons are from **React Native Vector Icons**.
- Used icons:
  - `logout`, `edit`, `trash`, `plus-circle`, `mail`, `survey`, `home`.

8. Accessibility Features

- **Touchable areas**: Min height and width of 44dp.
- **Color contrast**: Ensured WCAG compliant contrast between text and background.
- **Dynamic font scaling**: App supports user-selected font size settings from OS.

9. Reusability and Components

- **Reusable Components**:
  - SurveyCard
  - QuestionInput
  - OptionField
  - ModalPopup
  - SidebarDrawer
- These components follow atomic design principles for easier maintenance and updates.

📄 Coding Standards Document

Project Name: Survey Mobile Application

Author: Umair Younus Khan

Date: 20-06-2025

1. Purpose

This document outlines the programming conventions and standards used in the development of the survey app to ensure code quality, readability, consistency, and ease of maintenance.

2. File Structure and Naming

- Use meaningful, consistent names in camelCase format for variables and functions.
- PascalCase is used for React Native components (e.g., `SurveyCard.jsx`).
- Group files by feature/module:

```
bash
CopyEdit
/screens
/components
/api
/utils
```

3. Code Formatting

- **Indentation**: 2 spaces
- **Quotes**: Use single quotes `'text'`
- **Semicolons**: Always use semicolons to terminate statements
- **Curly Braces**: Always open curly braces on the same line

4. Comments

- Use `//` for single-line comments and `/** */` for block-level documentation.
- Document every function with:
  - Purpose
  - Parameters
  - Return value

5. Component Structure

Each component should follow:

```
js
CopyEdit
import React from 'react';
import { View, Text } from 'react-native';

const MyComponent = () => {
  return (
    <View>
      <Text>Hello</Text>
    </View>
  );
};

export default MyComponent;
```

6. API Standards

- All API calls are made using Axios.
- Base URL is stored in `apiConfig.js`.
- API error handling is consistent with `try/catch` blocks.

7. Async Storage

- Use `@react-native-async-storage/async-storage` for storing user login data securely.

8. Version Control

- Git is used with meaningful commit messages:

```
feat: create guest login flow
fix: resolve survey submission bug
```

📘 Project Policy Document

Project Name: Survey Mobile Application
Author: Umair Younus Khan
Date: 20-06-2025

1. Development Policy

- Only approved features should be pushed to the main branch.
- All code must be peer-reviewed before merging.
- Bugs are logged in a shared tracker (e.g., Notion/Google Sheet).

2. Branching Policy

- `main`: Production-ready code
- `dev`: In-progress features
- `feature/xyz`: Specific feature branch

3. Deployment Policy

- The app is tested on both emulator and real device before final release.
- Guest and registered flows are tested separately.

4. Communication Policy

- Daily updates to supervisor via email or WhatsApp group.
- Weekly progress reports submitted in PDF format.

5. Documentation Policy

- All features must be documented in `README.md`.
- Test cases, diagrams, and final report must be uploaded to shared drive.

📓 User Manual Document

Project Name: Survey Mobile Application
Author: Umair Younus Khan
Date: 20-06-2025

1. About the App

The Survey Mobile App enables users to create, share, and attempt surveys. Registered users have full control over surveys while guest users can only attempt a specific one.

---

2. Installation

- Install the app on your Android device (APK provided).
- Ensure Internet access is available.

---

3. Using the App

3.1 Guest User Flow

1. Open app → Tap on *Guest Login*.
2. Enter *Survey ID* and *Email*.
3. View and attempt the assigned survey.
4. Submit answers to view your score.

3.2 Registered User Flow

1. Tap *Register* and fill in your details.
2. Login using your email and password.
3. Navigate using the side drawer:
   - **Create Survey**: Add title, description, questions and options.
   - **Survey List**: View all your created surveys.
   - **Edit/Delete Survey**: Modify or remove existing surveys.
   - **Send Survey**: Share via email.

- **Attempt Surveys**: Attempt any public or private survey.

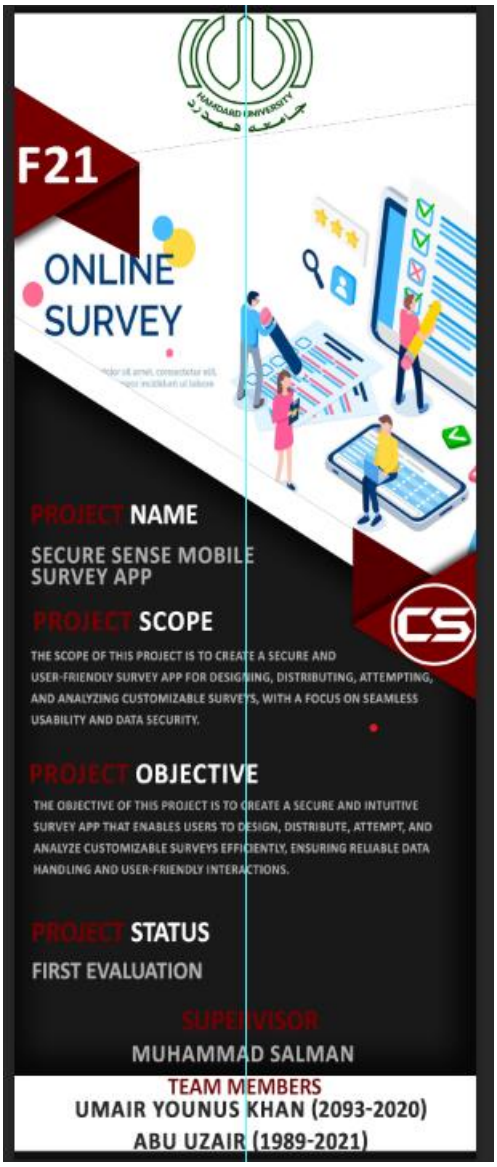
4. Features Summary

Feature	Description
Register/Login	Secure user access
Guest Access	Quick survey attempt via survey ID
Create Survey	Add questions, options, and correct answers
Attempt Survey	Choose options and submit for score
Send via Email	Share survey directly to recipient

5. FAQs

- **Q:** Can I edit a submitted survey?
  A: No, but you can create a copy and edit that.
- **Q:** Can guests create surveys?
  A: No, only registered users can create/edit surveys.

A5. FLYER & POSTER DESIGN



F21

ONLINE
SURVEY

PROJECT NAME

SECURE SENSE MOBILE
SURVEY APP

PROJECT SCOPE

THE SCOPE OF THIS PROJECT IS TO CREATE A SECURE AND USER-FRIENDLY SURVEY APP FOR DESIGNING, DISTRIBUTING, ATTEMPTING, AND ANALYZING CUSTOMIZABLE SURVEYS, WITH A FOCUS ON SEAMLESS USABILITY AND DATA SECURITY.

PROJECT OBJECTIVE

THE OBJECTIVE OF THIS PROJECT IS TO CREATE A SECURE AND INTUITIVE SURVEY APP THAT ENABLES USERS TO DESIGN, DISTRIBUTE, ATTEMPT, AND ANALYZE CUSTOMIZABLE SURVEYS EFFICIENTLY, ENSURING RELIABLE DATA HANDLING AND USER-FRIENDLY INTERACTIONS.

PROJECT STATUS

FIRST EVALUATION

SUPERVISOR

MUHAMMAD SALMAN

TEAM MEMBERS

UMAIR YOUNUS KHAN (2093-2020)

ABU UZAIR (1989-2021)

A6. COPY OF EVALUATION COMMENTS

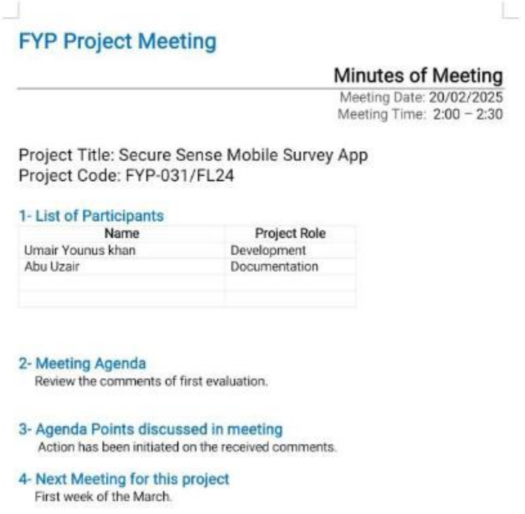
COPY OF EVALUATION COMMENTS BY SUPERVISOR FOR PROJECT – I MID SEMESTER EVALUATION

A Photostat or scanned copy should be placed when submitting document to Project Coordinator. (**Note**: Please remove this line when attach copy that is required)

COPY OF EVALUATION COMMENTS BY JURY FOR PROJECT – I END SEMESTER EVALUATION

A Photostat or scanned copy should be placed when submitting document to Project Coordinator. (**Note**: Please remove this line when attach copy that is required)

A7. MEETINGS’ MINUTES & Sign-Off Sheet



FYP Project Meeting

Minutes of Meeting

Meeting Date: 20/02/2025
Meeting Time: 2:00 – 2:30

Project Title: Secure Sense Mobile Survey App
Project Code: FYP-031/FL24

1- List of Participants

Name	Project Role
Umair Younus khan	Development
Abu Uzair	Documentation

2- Meeting Agenda

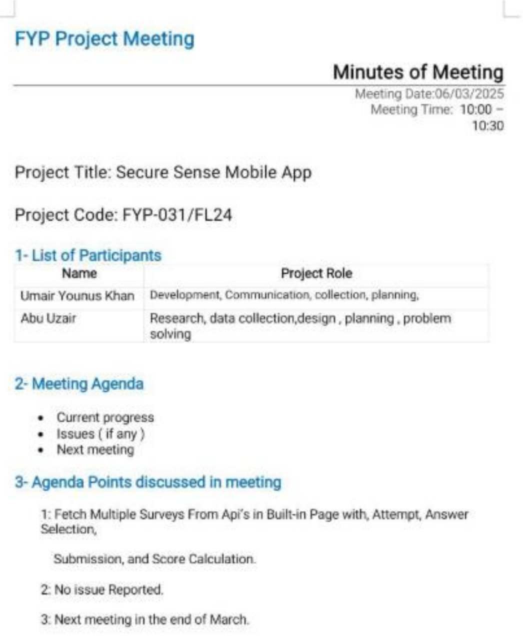
Review the comments of first evaluation.

3- Agenda Points discussed in meeting

Action has been initiated on the received comments.

4- Next Meeting for this project

First week of the March.



FYP Project Meeting

Minutes of Meeting

Meeting Date:06/03/2025
Meeting Time: 10:00 – 10:30

Project Title: Secure Sense Mobile App

Project Code: FYP-031/FL24

1- List of Participants

Name	Project Role
Umair Younus Khan	Development, Communication, collection, planning,
Abu Uzair	Research, data collection,design , planning , problem solving

2- Meeting Agenda

- Current progress
- Issues (if any)
- Next meeting

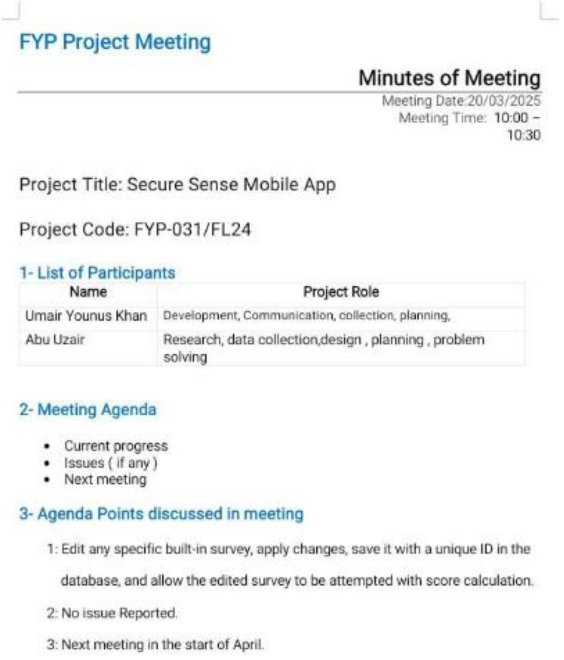
3- Agenda Points discussed in meeting

1: Fetch Multiple Surveys From Api's in Built-in Page with, Attempt, Answer Selection,

Submission, and Score Calculation.

2: No issue Reported.

3: Next meeting in the end of March.



FYP Project Meeting

Minutes of Meeting

Meeting Date:20/03/2025
Meeting Time: 10:00 – 10:30

Project Title: Secure Sense Mobile App

Project Code: FYP-031/FL24

1- List of Participants

Name	Project Role
Umair Younus Khan	Development, Communication, collection, planning,
Abu Uzair	Research, data collection,design , planning , problem solving

2- Meeting Agenda

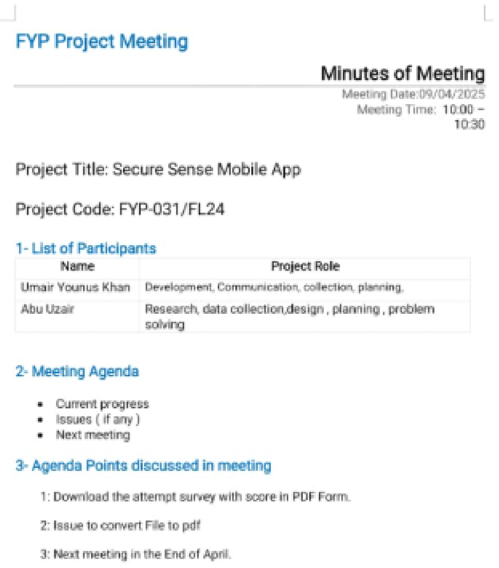
- Current progress
- Issues (if any)
- Next meeting

3- Agenda Points discussed in meeting

1: Edit any specific built-in survey, apply changes, save it with a unique ID in the database, and allow the edited survey to be attempted with score calculation.

2: No issue Reported.

3: Next meeting in the start of April.



FYP Project Meeting

Minutes of Meeting

Meeting Date:09/04/2025
Meeting Time: 10:00 – 10:30

Project Title: Secure Sense Mobile App

Project Code: FYP-031/FL24

1- List of Participants

Name	Project Role
Umair Younus Khan	Development, Communication, collection, planning,
Abu Uzair	Research, data collection,design , planning , problem solving

2- Meeting Agenda

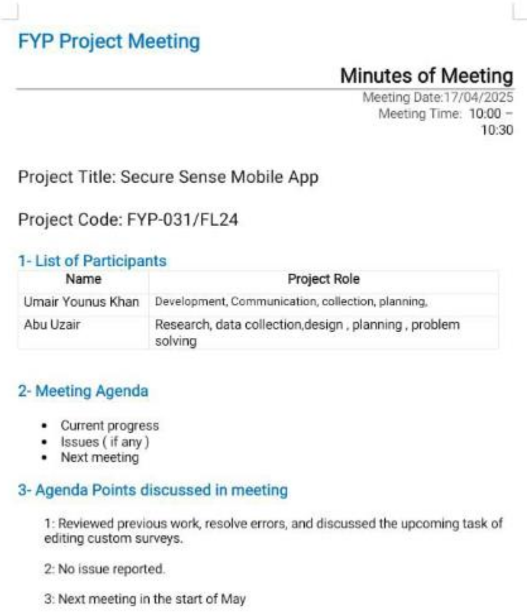
- Current progress
- Issues (if any)
- Next meeting

3- Agenda Points discussed in meeting

1: Download the attempt survey with score in PDF Form.

2: Issue to convert File to pdf

3: Next meeting in the End of April.



FYP Project Meeting

Minutes of Meeting

Meeting Date:17/04/2025
Meeting Time: 10:00 – 10:30

Project Title: Secure Sense Mobile App

Project Code: FYP-031/FL24

1- List of Participants

Name	Project Role
Umair Younus Khan	Development, Communication, collection, planning,
Abu Uzair	Research, data collection,design , planning , problem solving

2- Meeting Agenda

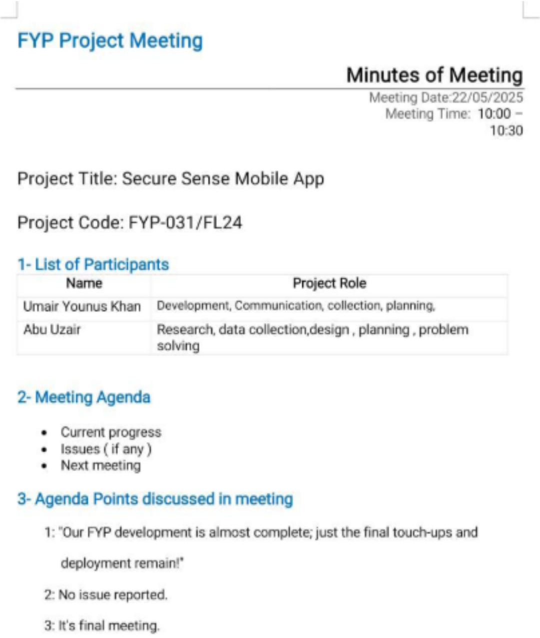
- Current progress
- Issues (if any)
- Next meeting

3- Agenda Points discussed in meeting

1: Reviewed previous work, resolve errors, and discussed the upcoming task of editing custom surveys.

2: No issue reported.

3: Next meeting in the start of May



FYP Project Meeting

Minutes of Meeting

Meeting Date:22/05/2025
Meeting Time: 10:00 – 10:30

Project Title: Secure Sense Mobile App

Project Code: FYP-031/FL24

1- List of Participants

Name	Project Role
Umair Younus Khan	Development, Communication, collection, planning,
Abu Uzair	Research, data collection,design , planning , problem solving

2- Meeting Agenda

- Current progress
- Issues (if any)
- Next meeting

3- Agenda Points discussed in meeting

1: "Our FYP development is almost complete; just the final touch-ups and deployment remain!"

2: No issue reported.

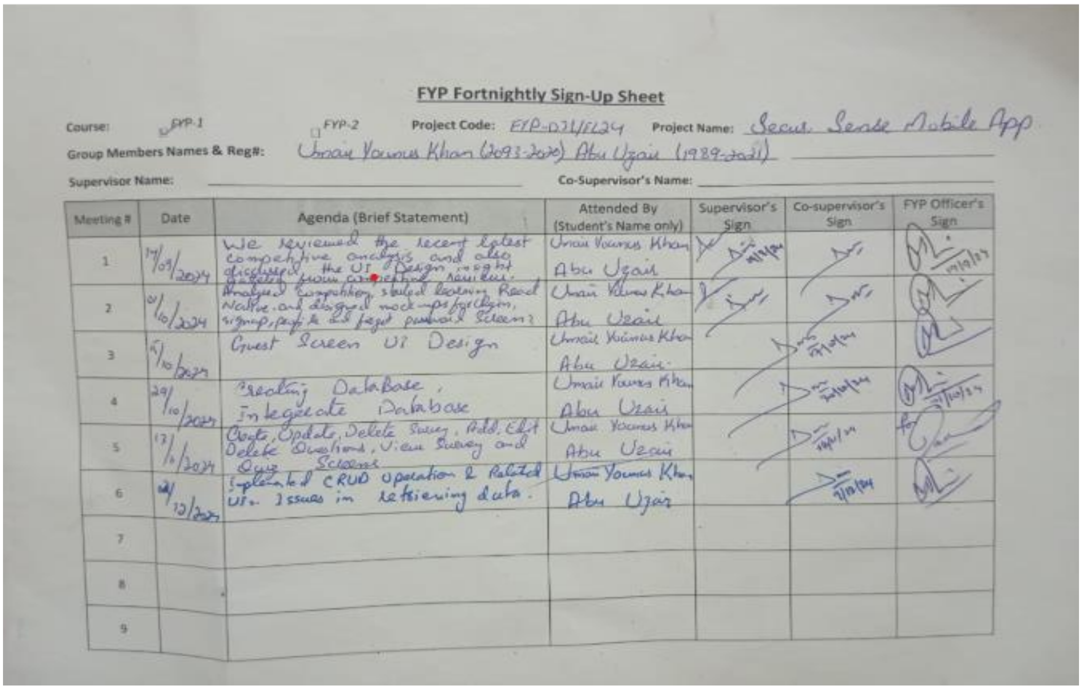
3: It's final meeting.

A8. DOCUMENT CHANGE RECORD

Date	Version	Author	Change Details
27/06/2025	01	Umair Younus Khan	Create the documents wit all relevant details according to the chapters
28/06/2025	02	Abu Uzair	Added Proposal, SRS & SDS.
01/07/2025	03	Abu Uzair	Added all the snapshot and relevant data.

02/07/2025	04	Umair Younus Khan Abu Uzair	Fixing Typo, grammer, alignmet & formatting.

A9. PROJECT PROGRESS



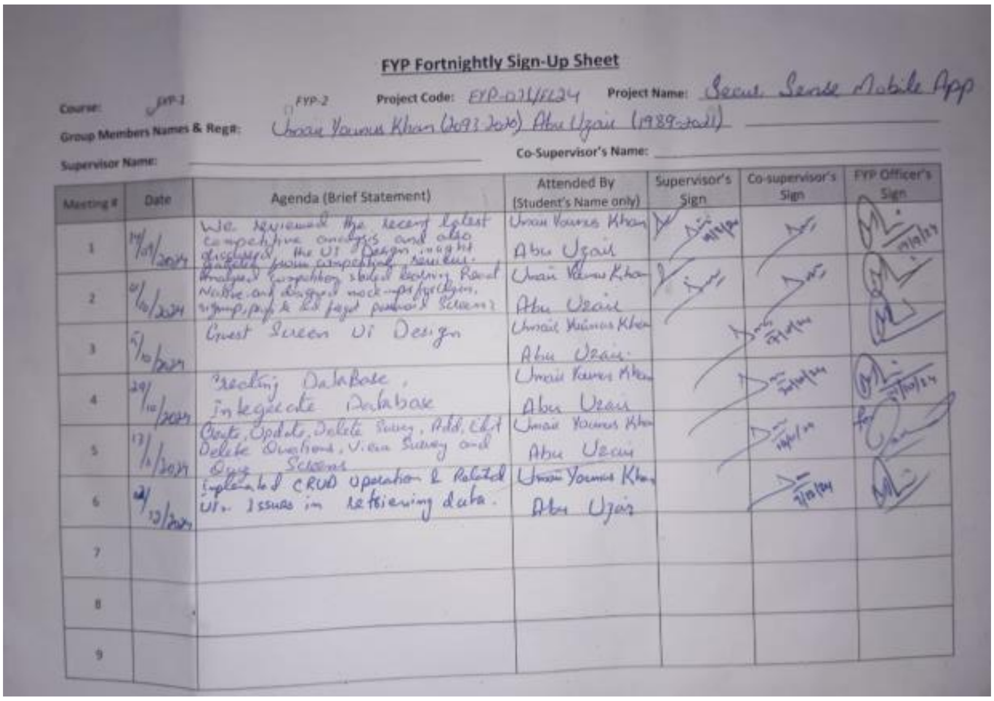
FYP Fortnightly Sign-Up Sheet

Course: ✓ FYP-1 □ FYP-2 Project Code: FYP-031/FL24 Project Name: Secure Sense Mobile App

Group Members Names & Reg#: Umair Younus Khan (2093-2020) Abu Uzair (1989-2021)

Supervisor Name: ______________ Co-Supervisor's Name: ______________

Meeting #	Date	Agenda (Brief Statement)	Attended By (Student's Name only)	Supervisor's Sign	Co-supervisor's Sign	FYP Officer's Sign
1	17/09/2024	We reviewed the recent latest competitive analysis and also discussed the UI Design insight gathered from competitor reviews.	Umair Younus Khan Abu Uzair			17/9/24
2	01/10/2024	Analysed competitors, started learning React Native and designed mock-ups for login, signup, profile and forget password screens	Umair Younus Khan Abu Uzair			
3	15/10/2024	Guest Screen UI Design	Umair Younus Khan Abu Uzair			
4	29/10/2024	Creating DataBase, Integrate Database	Umair Younus Khan Abu Uzair			31/10/24
5	17/11/2024	Create, Update, Delete Survey, Add, Edit Delete Questions, View Survey and Quiz Screens	Umair Younus Khan Abu Uzair			
6	09/12/2024	Implemented CRUD operation & Related UI. Issues in retrieving data.	Umair Younus Khan Abu Uzair		9/12/24	
7						
8						
9						



FYP Fortnightly Sign-Up Sheet

Course: ✓ FYP-1 □ FYP-2 Project Code: FYP-031/FL24 Project Name: Secure Sense Mobile App

Group Members Names & Reg#: Umair Younus Khan (2093-2020) Abu Uzair (1989-2021)

Supervisor Name: ______________ Co-Supervisor's Name: ______________

Meeting #	Date	Agenda (Brief Statement)	Attended By (Student's Name only)	Supervisor's Sign	Co-supervisor's Sign	FYP Officer's Sign
1	17/09/2024	We reviewed the recent latest competitive analysis and also discussed the UI Design insight gathered from competitor reviews.	Umair Younus Khan Abu Uzair			17/9/24
2	01/10/2024	Analysed competitors, started learning React Native and designed mock-ups for login, signup, profile and forget password screens	Umair Younus Khan Abu Uzair			
3	15/10/2024	Guest Screen UI Design	Umair Younus Khan Abu Uzair			
4	29/10/2024	Creating DataBase, Integrate Database	Umair Younus Khan Abu Uzair			31/10/24
5	17/11/2024	Create, Update, Delete Survey, Add, Edit Delete Questions, View Survey and Quiz Screens	Umair Younus Khan Abu Uzair			
6	09/12/2024	Implemented CRUD operation & Related UI. Issues in retrieving data.	Umair Younus Khan Abu Uzair		9/12/24	
7						
8						
9						