SamajConnect: A Community Networking & Engagement Platform

B.Tech-CSE & ICT, Semester-VII

Prepared at

BISAG-N
ISO 9001:2015
ISO 27001:2022
CMMI LEVEL-5

Bhaskaracharya National Institute for Space Applications & Geo-informatics
Ministry of Electronics and Information Technology, Govt. of India.
Gandhinagar

Prepared By

Het Virani
ID No. 25BISAG0185

Prit Patel
ID No. 25BISAG0186

Guided By:
Dr. Shakti Mishra, Dr. Pallab Kumar Nath
Department of CSE & ICT
PDEU, Gandhinagar.

External Guide:
Nirav Parekh
Project Manager,
BISAG-N, Gandhinagar

SUBMITTED TO

Pandit Deendayal Energy University

PANDIT DEENDAYAL ENERGY UNIVERSITY
GANDHINAGAR, GUJARAT
Formerly Pandit Deendayal Petroleum University (PDPU)

School Of Technology, PDEU, Gandhinagar

Bhaskaracharya National Institute for Space Applications and Geo-informatics
MeitY, Government of India
Phone: 079 - 23213081 Fax: 079 - 23213091
E-mail: info@bisag.gujarat.gov.in, website: https://bisag-n.gov.in/

ISO 9001:2015
ISO 27001:2022
CMMI LEVEL-5

CERTIFICATE

This is to certify that the project report compiled by ***Mr. Het Virani*** *student of 7th Semester* ***B.Tech-ICT*** *and* ***Mr. Prit Patel,*** *student of 7th Semester* ***B.Tech-CSE from School of Technology, Pandit Deendayal Energy University, Gandhinagar*** *have completed their Summer internship project satisfactorily. To the best of our knowledge, this is an original and bona fide work done by them. They have worked on Android-based application for "****SamajConnect: A Community Networking & Engagement****", starting from May 20th, 2025 to July 20th, 2025.*

During their tenure at this institute, they were found to be sincere and meticulous in their work. We appreciate their enthusiasm & dedication towards the work assigned to them.

We wish them every success.

Nirav Parekh
Project Manager,
BISAG- N, Gandhinagar

Punit Lalwani
Additional Director cum CISO,
BISAG- N, Gandhinagar

CERTIFICATE

*This is to certify that the 7th Semester Summer Internship Project entitled "**SamajConnect: A Community Network & Engagement**" has been carried out by **Het Virani** under my guidance in fulfilment of the degree of Bachelor of Technology in Information & Communication Technology (7th Semester) of Pandit Deendayal Energy University during the academic year 2025-2026.*

Internal Guide

Dr. Pallab Kumar Nath

Assistant Professor

Dept. of ECE

PDEU

Head Of Department

Dr. Pawan Sharma

Head of Department ICT

PDEU

CERTIFICATE

This is to certify that the 7th Semester Summer Internship Project entitled " ***SamajConnect: A Community Network & Engagement****" has been carried out by* ***Prit Patel*** *under my guidance in fulfilment of the degree of Bachelor of Technology in Computer Science & Engineering (7th Semester) of Pandit Deendayal Energy University during the academic year 2025-2026.*

Internal Guide

Dr. Shakti Mishra

Head of Department CSE

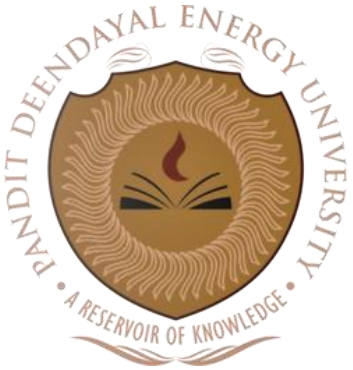
PDEU

Head Of Department

Dr. Shakti Mishra

Head of Department CSE

PDEU



About BISAG- N

ABOUT THE INSTITUTE

Modern day planning for inclusive development and growth calls for transparent, efficient, effective, responsive and low cost decision making systems involving multi-disciplinary information such that it not only encourages people's participation, ensuring equitable development but also takes into account the sustainability of natural resources. The applications of space technology and Geo–informatics have contributed significantly towards the socio-economic development. Taking cognizance of the need of geo-spatial information for developmental planning and management of resources, the department of Ministry of Electronics and Information Technology, Government of India, established "Bhaskaracharya National Institute for Space Applications and Geo-informatics" (BISAG- N). BISAG- N is an ISO 9001:2015, ISO 27001:2022 and CMMI: 5 certified institute. BISAG- N which was initially set up to carryout space technology applications, has evolved into a centre of excellence, where research and innovations are combined with the requirements of users and thus acts as a value added service provider, a technology developer and as a facilitator for providing direct benefits of space technologies to the grass root level functions/functionaries.

BISAG- N's Enduring Growth

Since its foundation, the Institute has experienced extensive growth in the sphere of Space technology and Geo-informatics. The objective with which BISAG- N was established is manifested in the extent of services it renders to almost all departments of the State. Year after year the institute has been endeavouring to increase its outreach to disseminate the use of geo-informatics up to grassroots level. In this span of nine years, BISAG- N has assumed multi-dimensional roles and achieved several milestones to become an integral part of the development process of the Gujarat State.

BISAG- N Journey

2003-04

Gujarat SATCOM Network

2007-08

Centre for Geo-informatics Applications

2010-11

Academy of Geo-informatics for Sustainable Development

2012-13

A full-fledged Campus

Activities

Satellite Communication..

for promotion and facilitation of the use of broadcast and teleconferencing networks for distant interactive training, education and extension.

Remote Sensing..

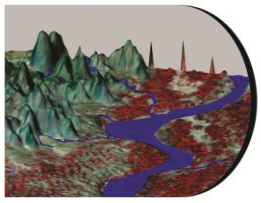
for Inventory, Mapping, Developmental planning and Monitoring of natural & man-made resources.

Geographic Information System..

for conceptualization, creation and organization of multi purpose common digital database for sectoral/integrated decision support systems.

Global Navigation Satellite System..

for Location based Services, Geo-referencing, Engineering Applications and Research.



Photogrammetry..

for Creation of Digital Elevation Model, Terrain Characteristic, Resource planning.

Cartography..

for thematic mapping, value added maps.

Software Development..

for wider usage of Geo-spatial applications, Decision Support Systems (desktop as well as web based), ERP solutions.

Education, Research and Training..

for providing Education, Research, Training & Technology Transfer to large number of students, end users & collaborators.

Applications of Geospatial Technology for Good Governance: Institutionalization

Through the geospatial technology, the actual situation on the ground can be accessed. The real life data collected through the technology forms the strong foundation for development of effective social welfare programs benefiting directly the grass root level people. The geospatial data collected by the space borne sensors along with powerful software support through Geographic Information System (GIS), the vital spatio-temporal maps, tables, and various statistics are being generated which feed into Decision Support System (DSS).

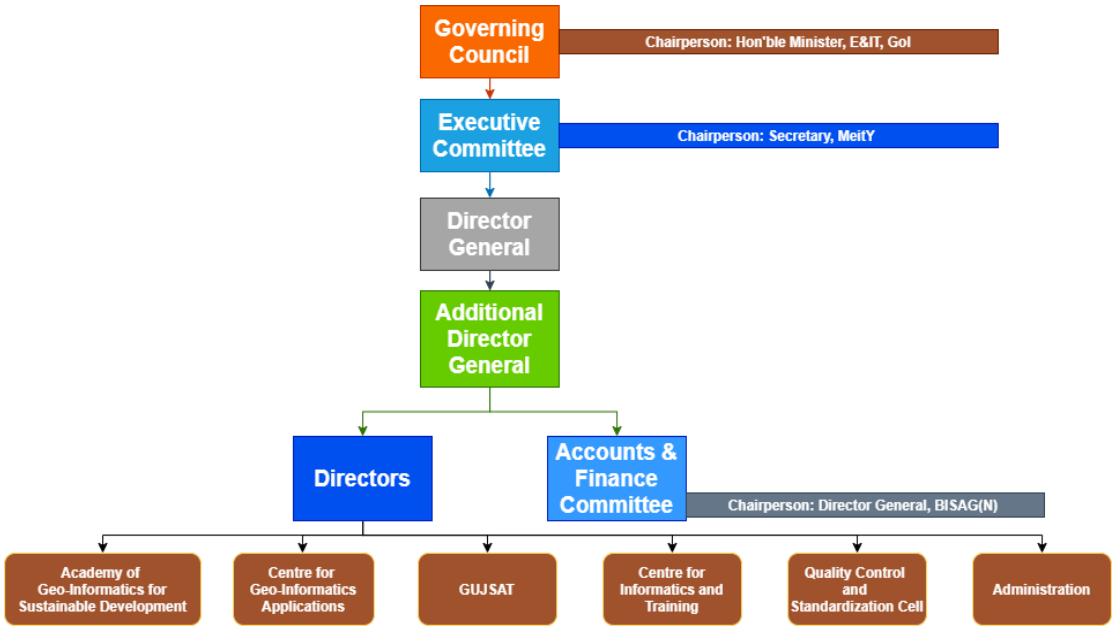
A multi-threaded approach is followed in the process of institutionalization of development of such applications. The 5 common threads which run through all the processes are: *Acceptability, Adaptability, Affordability, Availability and Assimilability*.
These are the "Watch Words" which any application developer has to meet. The "acceptability" addresses the issue that the application developed has met the wide acceptability among the users departments and the ultimate end beneficiary by way of providing all necessary data and statistics required. The "affordability" addresses the issue of the application product being cost effective. The "availability" aspect looks into aspect of easily accessible across any platform, anywhere and anytime. The applications should have inbuilt capability of easy adaptability to the changing spatio- and temporal resolutions of data, new aspects of requirements arising from time to time from users. The assimilability aspect ensures that the data from various sources / resolutions and technologies can be seamlessly integrated.

ACCEPTABILITY	▪ Problem definition by users • Proof of Concept development without financial liability on users ▪ Execution through collaboration under user's ownership
ADOPTABILITY	▪ Applications as per present systems & database ▪ Maximum Automation ▪ Minimum capacity building requirement at the user end
AFFORDABILITY :	▪ Multipurpose geo-spatial database, common, compatible, standardized (100s of layers) ▪ In house developed/open source software ▪ Full Utilization of available assets
AVAILABILITY:	▪ Departmental /Integrated DSS ▪ Desired Product delivery anytime, anywhere in the country
ASSIMILABILITY	▪ Integration of Various technologies like RS, GIS, GPS, Web MIS, Mobile etc.

Organizational Setup

The Institute is responsible for providing information and technical support to different Departments and Organizations. The Governing Body and the Empowered Executive Committee govern the functioning of BISAG- N. The Institute is registered under the Societies Registration Act 1860. Considering the scope and extent of activities of BISAG- N, its organizational structure has been charted out with defined functions.

Organizational Setup of BISAG- N



Governing Body

For smoother, easier and faster institutionalization of Remote Sensing and GIS technology, decision makers of the state were brought together to form the Governing Body. It is the supreme executive authority of the Institute. The Governing Body comprises of ex-officio members from various Government departments and Institutes.

- Hon'ble Minister of Electronics and Information Technology..... Chairperson (Ex-Officio)
- Hon'ble Minister of State Electronics and Information Technology.....……Deputy Chairperson (Ex-Officio)
- Secretary of Government of India: Ministry of Electronics and Information Technology..... .....Executive Vice Chairperson (Ex-Officio)
- Chief Executive Officer, Niti Aayog..... ……Member (Ex-Officio)
- Chairman, Indian Space Research Organization ..... ……Member (Ex-Officio)
- Secretary to Government of India: Department of Science and Technology ..... ……Member (Ex-Officio)
- Additional Secretary to Government of India: Ministry of Electronics and Technology .. ……Member (Ex-Officio)
- Chief Secretary to Government of Gujarat..... ……Member (Ex-Officio)
- President & Chief Executive Officer, National e-Governance Division, Ministry of Electronics
- and Information Technology ..... …… Member (Ex-Officio)
- Financial Advisor to Government of India: Ministry of Electronics and Information Technology…Member (Ex-Officio)
- Distinguished Professionals from the GIS field-Three (3) (To be nominated by the Chairperson)
- Director-General, Bhaskaracharya National Institute for Space Application and Geo-Informatics {BISAG(N)} ..... Member Secretary (Ex-Officio)

Centre for Geo-informatics Applications

Introduction

The objective of this technology group is to provide decision support to the sectoral stake holders through scientifically organized, comprehensive, multi-purpose, compatible and large scale (village level) geo-spatial databases and supporting analytical tools. These activities of this unit are executed by a well-trained team of multi-disciplinary scientists. The government has provided a modern infrastructure along with the state-of-the-art hardware and software. To study the land transformation and development over the years, a satellite digital data library of multiple sensors of last twenty years has been established and conventional data sets of departments have been co-registered with satellite data. The geo-spatial databases have been created using conventional maps, high resolution satellite 2D and 3D imagery and official datasets (attributes).The geo-spatial databases include terrain characteristics, natural and administrative systems, agriculture, water resources, city survey maps, village maps with survey numbers, water harvesting structures, water supply, irrigation, power, communications, ports, land utilization pattern, infrastructure, urbanization, environment data, forests, sanctuaries, mining areas, industries. They also include social infrastructure like the locations of schools, health centres, institutions, aganwadies, local government infrastructure etc. The geospatial database of nagar-palikas includes properties and amenities captured on city and town planning maps with 1000 GIS layers. Similar work for villages has been initiated as a pilot project.

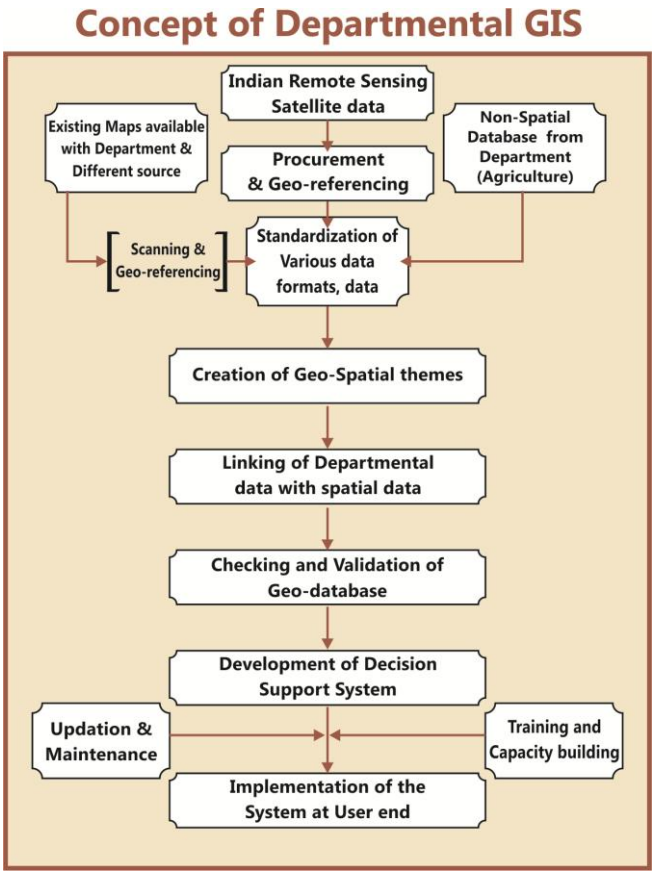
The applications of space technology and geo-informatics have been operational in almost all the development sectors of the state. Remote sensing and GIS applications have provided impetus to planning and developmental activities at grass root level as well as monitoring and management in various disciplines.

The GIS based Applications Development

The GIS software is a powerful tool to handle, manipulate and integrate both the spatial and non-spatial data. The GIS system operates on the powerful backend data base and Sequential Query Language (SQL) to inquiry the data bases. It has the capability to handle large volume of data and process to yield values of parameters which can be input to very important government activity as Decision Support System (DSS). Its mapping capabilities help the users and specialists in generating single and multi-theme wise maps.

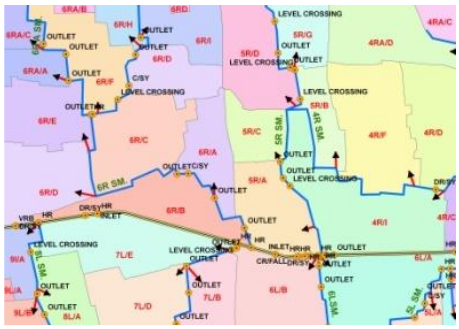
The GIS based applications development has been institutionalized in BISAG- N. This process can be listed as (Refer Figure for Details)

- Making the users aware of the GIS capabilities through introductory training programme and by exposing to already developed projects as success stories.
- Helping the users in defining the GIS based projects.
- Digitizing the data available with the users and encouraging them to collect any additional data as may be required.
- Generating the appropriate data bases with the full involvement of the users following the data bases standards



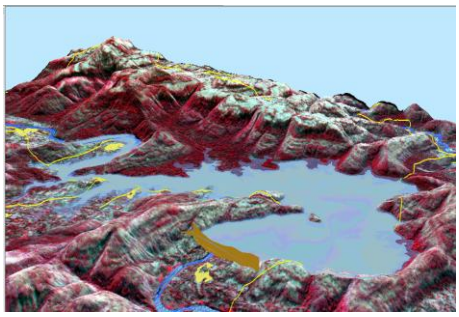
Remote Sensing and GIS Sectoral Applications:

Geo-informatics based Irrigation Management and Monitoring System



- The Geo-spatial information system for Irrigation water Management and Monitoring system for command areas in Sardar Sarovar Narmada Nigam Limited (SSNL) has been developed. Satellite image-based Irrigation monitoring system has been developed in GIS. From the multi-spectral Satellite images of every month, the irrigated areas were extracted.
- The irrigated area were overlaid on the geo-referenced cadastral maps and the statistics of area irrigated has been estimated.
- The user friendly Customized Decision Support System (DSS) has been developed.

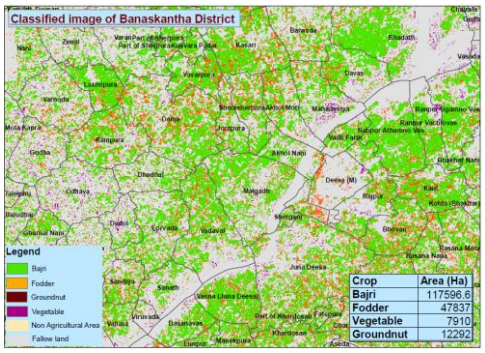
Preparation of DPR of Par–Tapi-Narmada Link using Geo-informatics for National Water development Agency (NWDA)



- The main objective of Par–Tapi-Narmada Link project is to divert surplus water available in west flowing rivers of south Gujarat and Maharashtra for utilization in the drought prone Saurashtra and Kachcha. On the request from NDWA, preparation of various maps for proposed DPR work was undertaken by the BISAG- N. Land use and submergence maps of proposed dams along with its statistics have been prepared by the BISAG- N. The detailed work consisted of generation of Digital Elevation Model (DEM), contour generation, Land use mapping, forest area generation of submergence extent at different levels etc.

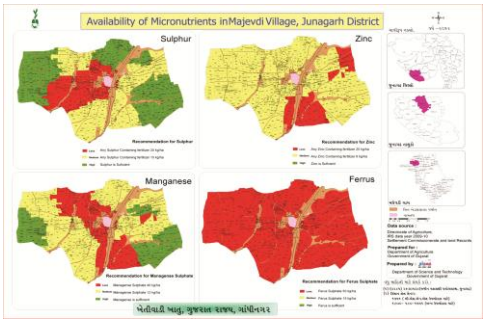
Agriculture

District and Village-level Crop Inventory



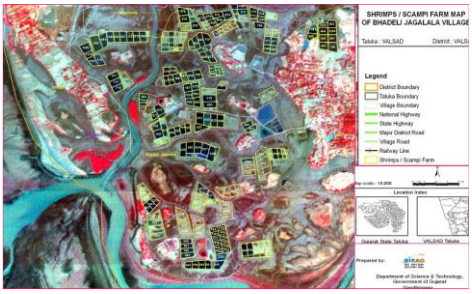
- Remote Sensing (RS) based Village-level Crop Acreage Estimation at was taken up in two villages of Anand and Mehsana districts of Gujarat state. The major objective of this study was to attempt village-level crop inventory during two crop seasons of Kharif (monsoon season) and Rabi (winter season) using single-date Indian Remote Sensing (IRS) LISS-III and LISS-IV digital data of maximum vegetative growth stage of major crops during each season.
- District-level crop acreage estimation during three cropping seasons namely Kharif, Rabi and Zaid (summer) seasons was also carried out in all the 26-districts of Gujarat State. Summer crop acreage estimation Gujarat State was carried out during 2012.

Spatial Variability Mapping of Soil Micro-Nutrients



- The spatial variability of soil micro-nutrients like Fe, Mn, Zn and Cu in various villages of different districts, Gujarat state was mapped using geo-informatics technology. The major objectives of this study were i) to quantify the variability of Mn, Fe, Cu and Zn concentration in soil; ii) to map the pattern of micro-nutrient variability in cadastral maps, iii) suggest proper application of micro-nutrients based on status of deficiency for proper crop management and iv) preparation of village-level atlases showing spatial variability of micro-nutrients.

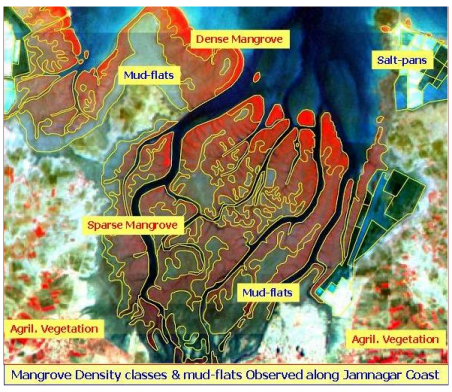
Geo-spatial Information System for Coastal Districts of Gujarat



- The project on development of Village-level Geo-spatial Information System for Shrimp Farms in Coastal Districts of Gujarat, was taken with major objective of development of Village-level Geo-spatial Information System for Shrimp/Scampi areas using Remote Sensing (RS) and GIS. This project was sponsored by the Marine Products Export Development Authority (MPEDA), Ministry of Commerce & Industry, Government of India for scientific management of Scampi farms in the coastal districts which can help fishermen to better their livelihood and increase the economic condition on sustainable basis. The customized query shell was developed using the open source software for sharing the information amongst the officers from MPEDA and potential users. This has helped the farmers to plan their processing and marketing operations so as to achieve better remunerations.

Environment and Forest

Mapping and Monitoring of Mangroves in the Coastal Districts of Gujarat State



Mangrove Density classes & mud-flats Observed along Jamnagar Coast

- Gujarat Ecology Commission, with technical inputs from the Bhaskaracharya National Institute for Space Applications and Geo-informatics - N (BISAG- N) made an attempt to publish Mangrove Atlas of the Gujarat state. Mangrove atlas for 13-coastal districts with 35-coastal talukas in Gujarat, have been prepared using Indian Remote sensing satellite images. The comparison of mangrove area estimates carried out by BISAG- N and Forest Survey of India (FSI) indicates a net increase in the area under mangrove cover. The present assessment by BISAG- N, has recorded 996.3 sq. km under mangrove cover, showing a steep rise to the tune of 88.03 sq. km. In addition to the existing Mangrove cover, the present assessment also gives the availability of potential area of 1153 sq. km, where mangrove regeneration program can be taken up.

Academy of Geo-informatics for Sustainable Development

Introduction

- Considering the requirement of high end research and development in the areas having relevance of geo-informatics technology for sustainable development, a separate infrastructure has been established. In collaboration with different institutes in the state as well as in the country, R&D activities are being carried out in the areas of climate change, environment, disaster management, natural resources management, infrastructure development, resources planning, coastal hazard and coastal zone management studies, etc. under the guidance of eminent scientists.

- Various innovative methodologies/models developed in this academy through the research process have helped in development of various applications. There are plans to enhance R&D activities manifold during coming years.
- This unit also provides training to more than 600 students every year in the field of Geo-informatics to the students from various backgrounds like water resources, urban planning, computer Engineering, IT, Agriculture in the areas of Remote sensing, GIS and their applications.

- This Academy has been established as a separate infrastructure for advanced research and development through following schools:
  - School of Geo-informatics
  - School of Climate & Environment
  - School of Integrated Coastal Zone Management

- School of Sustainable Development Studies
- School of Natural Resources and Bio-diversity
- School of Information Management of Disasters
- School of Communication and Society

During XIIth Five year Plan advance applied research through above schools shall be the main thrust area. Already M. Tech and Ph.D. students of other Universities/ Institutes are doing research in this academy in applied sciences under various collaborative programmes.

M. Tech. Students' Research Programme

The academy started M. Tech. students' research programme in a systematic way. It admitted 11 students from various colleges and universities in Gujarat, Rajasthan and Madhya Pradesh for period of 10 months from August 2011 to May 2012. All the students were paid stipend of Rs. 6000 per month during the tenure. The research covered the following areas:

- Cloud computing techniques
- Mobile communication
- Design of embedded systems
- Aquifer modelling
- Agricultural and Soils Remote Sensing
- Digital Image processing Techniques (Data Fusion and Image Classification).

The research resulted in various dissertations and publications in national and international journals.

• Now nine students, one from IIT, Kharagpur, three from GTU, one from M. S University, Vadodara and four from GU, are undergoing their Ph. D programme. Out of nine, two thesis have been submitted. Two students are from abroad. One each from Vietnam and Yemen. Since then (after approval of research programme from the Governing Body), 200+ papers have been published by the Academy.

CANDIDATE'S DECLARATION

We declare that the 7th semester internship project report entitled "**SamajConnect: A Community Networking & Engagement**" is our own work conducted under the supervision of the external guide **Mr. Nirav Parekh** from BISAG-N (Bhaskaracharya National Institute for Space Applications & Geo-informatics). We further declare that to the best of our knowledge, the report for this project does not contain any part of the work which has been submitted previously for such project either in this or any other institutions without proper citation.

Candidate 1's Signature

Het Virani

Student ID: 25BISAG0185

Candidate 2's Signature

Prit Patel

Student ID: 25BISAG0186

Submitted To:

School of Technology, Pandit Deendayal Energy University, Gandhinagar

ACKNOWLEDGMENT

We are grateful to **Shri T.P. Singh,** Director General (BISAG-N), for giving us this opportunity to work the guidance of renowned people of the field of GIS-Based Portal, also providing us with the required resources in the company.

We would like to express our endless thanks to our external guide, **Mr. Nirav Parekh,** and to the Training Cell, **Mr. Punit Lalwani & Mr. Sidhdharth Patel** at Bhaskaracharya National Institute of Space Application and Geo-informatics, for their sincere and dedicated guidance throughout the project development.

Also, our hearty gratitude to our Head of Department, **Dr. Shakti Mishra & Dr. Pawan Sharma** and our internal guide **Dr. Shakti Mishra & Dr. Pallab Kumar Nath** for giving us encouragement and technical support on the project.

Het Virani.
Student ID: 25BISAG0185

Prit Patel.
Student ID: 25BISAG0186

ABSTRACT

SamajConnect is an Android-based mobile application developed to bring people closer by allowing them to join and interact within their community groups, known as "Samaj." The app supports essential features like user registration with OTP verification, admin-controlled community creation, event management, and profile updates. It is built using Java for the frontend, Spring Boot for the backend, and PostgreSQL as the database, ensuring a secure and smooth experience for all users.

One of the most unique and powerful features of SamajConnect is the Relationship Tree. This feature allows users to visually explore and understand their family and social connections through a dynamic TreeView structure. Users can view relationships based on generation levels, send and accept relationship requests, and explore detailed member profiles. This visual representation makes it easy to trace and understand family links within the community, offering a clear and interactive way to strengthen social bonds.

The app also includes an event dashboard where users can view upcoming events, like or dislike them, and explore full-screen images with popularity indicators. Profile management, including personal details and image updates, is also supported. A built-in search function allows users to quickly find members using names, emails, or phone numbers.

With its focus on community engagement and clear relationship mapping, SamajConnect provides a complete and easy-to-use platform for connecting people within their Samaj. It lays the foundation for future improvements like event reminders, data insights, and real-time notifications.

Index

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Full Form
API	Application Programming Interface
JWT	JSON Web Token
OTP	One Time Password
UI	User Interface
UX	User Experience
REST	Representational State Transfer
CRUD	Create, Read, Update, Delete
DTO	Data Transfer Object
JPA	Java Persistence API
SMTP	Simple Mail Transfer Protocol

1. INTRODUCTION

1.1. Problem Statement

In today's digital age, maintaining strong community bonds and understanding family relationships within traditional social groups has become increasingly challenging. People often lose track of their extended family connections, community events, and social relationships due to geographical dispersion and busy lifestyles. Traditional methods of community management lack digital integration, making it difficult to:

- Track and visualize family relationships within the community
- Organize and manage community events effectively
- Maintain updated member profiles and contact information
- Facilitate communication between community members
- Preserve family lineage and relationship data for future generations

1.2. Purpose

SamajConnect : A Community Networking and Engagement Platform aims to bridge the gap between traditional community structures and modern digital solutions by providing a comprehensive platform that enables community members to:

- Maintain and visualize their family relationships through an interactive relationship tree
- Stay connected with community events and activities
- Manage their profiles and personal information
- Search and connect with community members
- Preserve their cultural and familial heritage digitally

1.3. Objective

The primary objectives of A Community Networking and Engagement Platform are:

1. Community Management: Provide a centralized platform for Samaj administration and member management
2. Relationship Mapping: Implement a visual family tree system to help users understand their connections
3. Event Management: Enable efficient creation, management, and participation in community events
4. User Engagement: Foster community interaction through profile management and search capabilities
5. Data Security: Ensure secure user authentication and data protection

1.4. Overview

SamajConnect : A Community Networking and Engagement Platform is a comprehensive Android-based mobile application designed to strengthen community bonds within traditional social groups. The application features a robust backend built with Spring Boot and PostgreSQL, ensuring reliable data management and security. The system supports two types of users: administrators who can create and manage Samaj communities, and individual members who can join existing communities and participate in various activities.

The application's standout feature is the Relationship Tree, which allows users to visualize and understand their family connections through an interactive tree structure. This feature, combined with event management, profile updates, and search functionality, creates a complete ecosystem for community engagement.

1.5. Tools and Technology Used

1) **Frontend:**
   - Java (Android SDK)
   - Android Studio Giraffe | 2022.3.1 Patch 1
   - Material Design Components
   - Volley for HTTP requests
   - GSON for JSON parsing
   - Glide for image loading
2) **Backend:**
   - Spring Boot Framework 3.1.0
   - Spring Security for authentication
   - Spring Data JPA for database operations
   - PostgreSQL Database 15
   - JWT for token-based authentication
   - Java Mail for email services
3) **Development Tools**
   - Git for version control
   - Postman for API testing
   - Android Emulator for testing
   - IntelliJ IDEA 2023.2 for backend development

2. PROJECT SCOPE

2.1. Features and Functionality

Core Features:

1. User Authentication System

- Email-based registration with OTP verification
- Secure login with JWT token authentication
- Password reset functionality
- Role-based access control (Admin/Member)

2. Samaj Management

- Admin-controlled community creation
- Community member management
- Samaj information and rules display
- Member search and filtering

3. Relationship Tree System

- Interactive family tree visualization
- Relationship request and approval system
- Generation-based relationship mapping
- Lineage context support (Paternal/Maternal)
- Visual node-based family connections

4. Event Management

- Event creation and editing (Admin only)
- Event listing with upcoming/recent filters
- Event reactions (Like/Dislike)
- Image upload for events
- Event details with full-screen image viewing

5. Profile Management

- Personal information updates
- Profile image upload and management
- Contact information management
- Gender and address details

6. Search and Discovery

- Member search by name, email, or phone
- Real-time search suggestions
- Filtered search results
- Member details viewing

2.2. Development Constraints

Technical Constraints:

- Android API level 24 (Android 7.0) minimum requirement
- Network dependency for all major functionalities
- Image size limitations for profile and event pictures
- Database storage limitations for Base64 encoded images

Time Constraints:

- Development timeline of 9 weeks
- Limited testing period due to internship schedule
- Concurrent development of frontend and backend

Resource Constraints:

- Two-person development team (Het Virani and Prit Patel)
- Limited access to diverse testing devices
- Budget constraints for cloud hosting and services

Platform Constraints:

- Android-only implementation (no iOS version)
- Requires stable internet connection
- Limited offline functionality

2.3. Deliverables

Primary Deliverables:

1. Complete Android application (APK file)
2. Backend REST API with Spring Boot
3. Database schema and setup scripts
4. Technical documentation
5. User manual and installation guide
6. Source code with version control history
7. Test cases and testing documentation

Secondary Deliverables:

1. API documentation
2. Database design documentation
3. UI/UX design specifications
4. Deployment guide
5. Project presentation materials

2.4. Out-of-Scope

Features Not Included:

1. iOS application development
2. Payment gateway integration
3. Advanced analytics and reporting
4. Multi-language support
5. Offline data synchronization
6. Push notifications
7. Social media integration
8. Advanced security features like biometric authentication

Technical Limitations:

1. No cloud storage for images (using database storage)
2. No automated backup systems
3. No load balancing or clustering
4. No advanced caching mechanisms
5. No third-party integrations

3. FEASIBILITY ANALYSIS

3.1. Technical Feasibility

Frontend Feasibility:

The Android platform provides robust support for the required features through its comprehensive SDK. Java programming language offers excellent documentation and community support. The chosen libraries (Volley, GSON, Glide) are well-established and reliable for HTTP requests, JSON parsing, and image loading respectively.

Backend Feasibility:

Spring Boot framework provides excellent support for REST API development with built-in security, data persistence, and email services. PostgreSQL database offers reliable data storage with support for complex relationships required for the family tree feature. JWT authentication provides secure and stateless user authentication.

Integration Feasibility:

The REST API architecture ensures clean separation between frontend and backend, making integration straightforward. JSON data format provides efficient data exchange between Android application and Spring Boot backend.

Technical Challenges Addressed:

- Complex relationship mapping solved using recursive algorithms
- Image storage handled through Base64 encoding
- Security implemented through JWT tokens and password encryption
- Email verification achieved through SMTP integration

3.2. Time Schedule Feasibility

Development Timeline: 9 Weeks

Week 1: Project Setup and Planning

- Requirements refinement and scope finalization
- Technology stack confirmation
- Development environment setup
- Database schema design

Week 2: Backend Core Development

- User authentication system
- Database models and repositories
- REST API endpoints for user management

Week 3: Frontend Foundation

- Android application structure
- User authentication screens
- Navigation framework
- API integration setup

Week 4: Core Backend Features

- Samaj management system
- Event management APIs
- Relationship management foundation

Week 5: Core Android Features

- Dashboard implementation
- Event listing screens
- Profile management

Week 6: Advanced Features - Relationship Tree

- Relationship tree implementation
- Relationship request system

Week 7: Advanced Features - Event Reactions and Search

- Event reactions system
- Member search functionality

Week 8: Integration and Testing

- Frontend-backend integration
- Comprehensive testing
- Bug fixes

Week 9: Finalization and Documentation

- System testing and bug fixes
- Documentation completion
- User manual creation
- Project presentation

Risk Mitigation:

- Parallel development of frontend and backend
- Regular integration testing
- Agile development approach with weekly sprints
- Prioritized feature implementation

3.3. Operational Feasibility

User Acceptance:

The application addresses real community needs for digital connection and relationship management. The intuitive interface design ensures easy adoption by users of varying technical expertise.

Administrative Feasibility:

The admin-controlled community creation ensures proper governance and prevents misuse. Role-based access control provides appropriate permissions to different user types.

Maintenance Feasibility:

The modular architecture and clean code practices ensure easy maintenance and updates. Comprehensive documentation supports future development and troubleshooting.

Scalability Considerations:

The system architecture supports horizontal scaling through additional server instances. Database design accommodates growing user base and relationship data.

3.4. Implementation Feasibility

Development Team Capability:

Two-person team (Me and Prit) with complementary skills in Android development and backend systems. Strong foundation in Java programming and database design.

Resource Availability:

- Development tools: Android Studio, IntelliJ IDEA (free/student licenses)
- Testing devices: Android emulators and personal devices
- Database: PostgreSQL (open source)
- Hosting: Local development environment

Technical Infrastructure:

- Version control: Git repository
- API testing: Postman
- Database management: pgAdmin 4
- Continuous integration: Local testing environment

3.5. Economic Feasibility

Development Costs:

- Personnel: Internship project (no salary costs)
- Software licenses: Free/open-source tools
- Hardware: Existing development machines
- Testing: Android emulators (free)

Operational Costs:

- Database hosting: Minimal for development
- Email service: Free tier sufficient for testing
- Domain and hosting: Not required for academic project

Cost-Benefit Analysis:

- Low development cost due to open-source technologies
- High potential value for community organizations
- Scalable architecture supports future monetization
- Educational value for team members

Return on Investment:

- Internship completion and learning experience
- Portfolio enhancement for career development

3.6. Nom-Functional Requirements

Performance Requirements:

- Application startup time: < 3 seconds
- API response time: < 2 seconds for standard operations
- Image loading time: < 5 seconds for high-resolution images
- Database query response: < 1 second for simple queries

Security Requirements:

- Password encryption using BCrypt
- JWT token-based authentication
- Input validation and sanitization
- SQL injection prevention through JPA
- XSS protection through proper data encoding

Usability Requirements:

- Intuitive user interface following Material Design guidelines
- Consistent navigation patterns
- Error messages and user feedback
- Accessibility support for different user abilities

Reliability Requirements:

- 99% uptime during testing period
- Graceful error handling and recovery
- Data consistency and integrity
- Backup and recovery procedures

Scalability Requirements:

- Support for multiple Samaj communities
- Horizontal scaling capability
- Database optimization for large datasets
- Efficient memory management

Compatibility Requirements:

- Android API level 24+ support
- Various screen sizes and resolutions
- Different Android device manufacturers
- Network connectivity variations

4. SOFTWARE AND HARDWARE REQUIREMENTS

4.1. Developer Software Requirements

Software	Version	Purpose
Android Studio	Giraffe (2022.3.1 Patch 1)	Android application development
IntelliJ IDEA	2023.2	Backend development
Java Development Kit	JDK 11	Programming language runtime
PostgreSQL	15	Database management system
Git	2.40	Version control system
Postman	10.0	API testing and documentation
pgAdmin	4.30	Database administration
Android SDK	API 24+	Android development framework

Table 4.1: List of Software Tools Used

Additional Tools:

- Gradle 8.0+ for build automation
- Maven 3.8+ for dependency management
- Chrome Browser for web-based testing
- Image editing software for UI assets

4.2. Developer Hardware Requirements

Minimum Requirements:

- Processor: Intel Core i5 or AMD Ryzen 5
- RAM: 8 GB DDR4
- Storage: 256 GB SSD
- Graphics: Integrated graphics sufficient
- Network: Stable internet connection
- Display: 1920x1080 resolution

Recommended Requirements:

- Processor: Intel Core i7 or AMD Ryzen 7
- RAM: 16 GB DDR4
- Storage: 512 GB SSD
- Graphics: Dedicated GPU for Android emulator
- Network: High-speed broadband
- Display: Dual monitor setup

Testing Devices:

- Android smartphone (API 24+)
- Android tablet for UI testing
- Various screen sizes for compatibility

4.3. User Software requirements

Mobile Device Requirements:

- Operating System: Android 7.0 (API level 24) or higher
- Available Storage: 100 MB for application installation
- RAM: 2 GB minimum, 4 GB recommended
- Network: 3G/4G/5G or Wi-Fi connectivity

4.4. User Hardware Requirements

Smartphone Specifications:

- Screen Size: 5.0 inches minimum
- Resolution: 720p (HD) minimum, 1080p recommended
- Processor: Quad-core 1.4 GHz minimum
- RAM: 2 GB minimum, 4 GB recommended
- Storage: 32 GB with 100 MB available space
- Camera: 5 MP for profile pictures (optional)
- Network: 3G/4G/5G or Wi-Fi capability

Tablet Specifications (Optional):

- Screen Size: 7.0 inches minimum
- Resolution: 1024x768 minimum
- Similar processor and RAM requirements as smartphone
- Enhanced viewing experience for family tree

Network Requirements:

- Internet Speed: 1 Mbps minimum for basic functionality
- Data Usage: Approximately 10-50 MB per month depending on usage
- Latency: < 500ms for optimal experience

5. PROJECT MODEL

5.1. Planning

Development Methodology: Agile with Spiral Model Elements

The project follows an agile development approach with spiral model characteristics, emphasizing iterative development, risk assessment, and continuous improvement.

Phase 1: Inception (Week 1)

- Project scope definition
- Stakeholder identification
- Initial requirements gathering
- Technology stack evaluation
- Team role assignment

Phase 2: Elaboration (Week 2)

- Detailed requirements analysis
- System architecture design
- Database schema design
- UI/UX wireframes
- Risk assessment and mitigation planning

Phase 3: Construction (Week 3-7)

- Backend API development
- Android application development
- Database implementation
- Integration testing
- Feature implementation in sprints

Phase 4: Transition (Week 8-9)

- System testing and bug fixes
- Documentation completion
- User manual creation

- Deployment preparation
- Project presentation

Sprint Planning:

- 1-week sprints with defined deliverables
- Weekly team meetings for progress review
- Daily standups for coordination
- Sprint retrospectives for continuous improvement

5.2. Project Scheduling (Gantt chart)

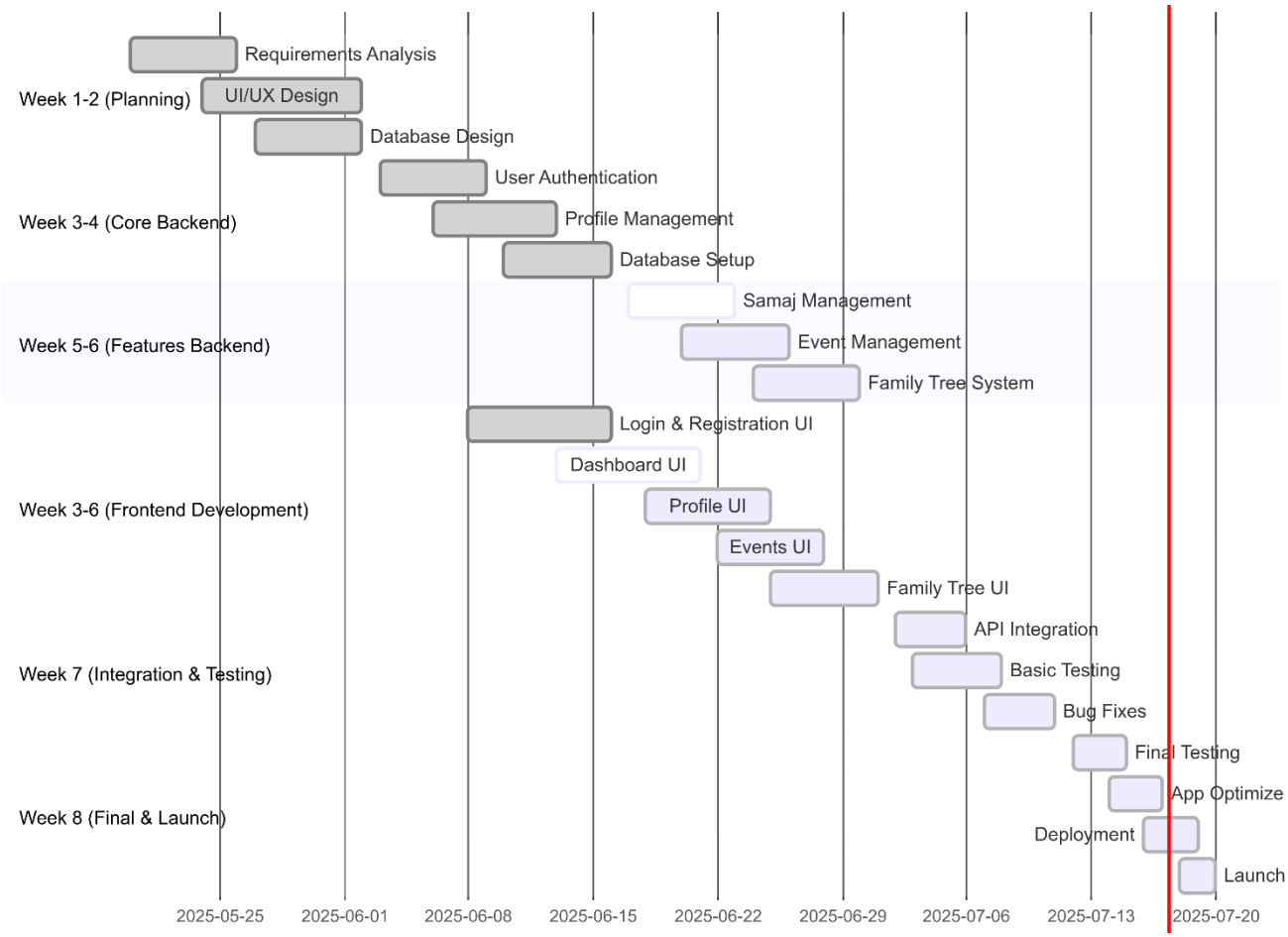


Fig 5.1 Gantt Chart

5.3. Risk Analysis

Technical Risks:

1. Database Performance Risk

- Probability: Medium
- Impact: High
- Mitigation: Database optimization, indexing, query optimization
- Contingency: Alternative database solutions, caching mechanisms

2. Android Compatibility Risk

- Probability: Medium
- Impact: Medium
- Mitigation: Extensive testing on multiple devices and API levels
- Contingency: Minimum API level adjustment, alternative UI components

3. Integration Complexity Risk

- Probability: High
- Impact: High
- Mitigation: Early integration testing, API documentation
- Contingency: Simplified integration approach, mock services

Project Risks:

1. Time Constraint Risk

- Probability: High
- Impact: High
- Mitigation: Agile development, feature prioritization
- Contingency: Scope reduction, MVP approach

2. Team Coordination Risk

- Probability: Medium
- Impact: Medium
- Mitigation: Regular communication, clear role definition
- Contingency: Increased meeting frequency, task redistribution

3. Requirement Changes Risk

- Probability: Medium
- Impact: Medium
- Mitigation: Agile methodology, stakeholder involvement
- Contingency: Change control process, impact assessment

External Risks:

1. Technology Obsolescence Risk

- Probability: Low
- Impact: Medium
- Mitigation: Current technology selection, update monitoring
- Contingency: Technology migration plan

2. Third-party Service Risk

- Probability: Low
- Impact: High
- Mitigation: Service reliability assessment, backup options
- Contingency: Alternative service providers, local implementations

5.4. Development & Design

Architecture Design Principles:

- Separation of concerns
- Modularity and reusability
- Scalability and maintainability
- Security by design
- User-centric design

Backend Architecture:

- RESTful API design
- Layered architecture (Controller, Service, Repository)
- Dependency injection
- Exception handling
- Input validation

Frontend Architecture:

- Model-View-Controller (MVC) pattern
- Activity and Fragment lifecycle management
- Asynchronous operations
- Memory management
- User interface responsiveness

Database Design:

- Normalized relational design
- Entity relationships modelling
- Data integrity constraints
- Performance optimization

Security Design:

- Authentication and authorization
- Data encryption
- Input sanitization
- SQL injection prevention
- Cross-site scripting protection

5.5. Evaluation

Evaluation Criteria

1. Functionality Assessment

- Feature completeness
- User requirement satisfaction
- System reliability
- Performance benchmarks

2. Quality Metrics

- Code quality and maintainability
- Test coverage
- Bug density
- User experience rating

3. Technical Evaluation

- Architecture adherence
- Security implementation
- Scalability potential
- Integration success

Evaluation Methods:

- Unit testing for individual components
- Integration testing for system interactions
- User acceptance testing for requirement validation
- Performance testing for system benchmarks

Success Criteria:

- All core features implemented and functional
- 95% test case pass rate
- User acceptance rating > 4.0/5.0
- Performance targets met
- Security requirements satisfied

6. PROJECT PLAN

6.1 Project Overview and Timeline

The A Community Networking and Engagement Platform project follows a structured 9-week development timeline designed to accommodate the constraints of an internship period while ensuring comprehensive feature implementation and thorough testing. This timeline represents an intensive development schedule that maximizes productivity through parallel development streams and agile methodologies.

The project timeline is strategically divided into three main phases: Foundation Phase (Weeks 1-3), Development Phase (Weeks 4-7), and Finalization Phase (Weeks 8-9). Each phase builds upon the previous one, ensuring a logical progression from initial setup to final deployment. The compressed timeline necessitates efficient resource allocation and clear milestone definitions to maintain project momentum and quality standards.

6.2 Detailed Weekly Breakdown

Week 1: Project Initiation and Foundation Setup

The first week establishes the groundwork for the entire development process. This phase focuses on comprehensive project planning, environment configuration, and team coordination. The week begins with detailed requirements analysis, where we refine the initial project scope based on stakeholder feedback and technical feasibility assessments. This involves creating detailed user stories, defining acceptance criteria, and establishing clear project boundaries.

During this week, both team members collaborate on setting up their respective development environments. This includes installing and configuring Android Studio Giraffe 2022.3.1 Patch 1, IntelliJ IDEA 2023.2, PostgreSQL 15, and all necessary SDKs and dependencies. Version control setup using Git ensures proper code management and collaboration workflows. The database schema design process begins with entity relationship modelling, focusing on the complex relationship structures required for the family tree functionality.

The week concludes with the creation of project documentation templates, coding standards definition, and communication protocols establishment. Risk assessment and mitigation strategies are developed to address potential challenges in the compressed timeline. Initial UI/UX wireframes are created to guide the frontend development process.

Week 2: System Architecture and Design Finalization

Week two concentrates on solidifying the technical architecture and design specifications. The backend architecture is finalized using Spring Boot 3.1.0 framework, with detailed API endpoint specifications created for all required functionalities. The database schema is refined and optimized, with particular attention to the relationship modeling required for the family tree feature. Indexing strategies and query optimization plans are developed to ensure system performance.

The frontend architecture is established using Android's Model-View-Controller pattern, with activity and fragment hierarchies defined. Material Design component selection and customization guidelines are established to ensure consistent user interface design. The integration strategy between frontend and backend is detailed, including error handling protocols and data synchronization mechanisms.

Security architecture is designed during this week, incorporating JWT token-based authentication, password encryption using BCrypt, and input validation strategies. API documentation is created using industry-standard formats, facilitating smooth integration between frontend and backend development streams. Testing strategies and frameworks are selected and configured for both unit and integration testing.

Week 3: Backend Core Development

The third week marks the beginning of intensive backend development. The Spring Boot application structure is implemented with proper layering including controllers, services, repositories, and data transfer objects. The user authentication system is developed as the foundation for all other features, incorporating secure registration, login, email verification with OTP, and password reset functionalities.

Database connectivity is established and tested, with initial entity models created for User, Samaj, Event, and Relationship structures. Repository interfaces are implemented using Spring Data JPA, providing efficient database operations. The email service integration is completed using JavaMail, enabling OTP delivery and notification capabilities.

Core API endpoints for user management are developed and thoroughly tested using Postman. This includes registration, authentication, profile management, and basic CRUD operations. Error handling mechanisms are implemented to provide meaningful feedback for various failure scenarios. The JWT token generation and validation system is integrated across all secured endpoints.

Week 4: Android Application Foundation

Week four focuses on establishing the Android application foundation and implementing core user interface components. The application structure is created with proper activity lifecycle management and navigation framework implementation. The authentication screens including login, registration, and OTP verification are developed with comprehensive input validation and user feedback mechanisms.

HTTP client configuration using Volley library is implemented to facilitate communication with the backend API. JSON parsing capabilities using Gson are integrated for efficient data serialization and deserialization. Image handling functionality using Glide library is implemented for profile pictures and event images.

The dashboard activity is developed as the central hub of the application, incorporating user profile display, navigation menu, and basic event listing functionality. Material Design components are implemented consistently across all screens, ensuring a professional and intuitive user interface. Basic error handling and loading states are implemented to enhance user experience.

Week 5: Core Feature Integration and Backend Enhancement

The fifth week involves intensive integration work between frontend and backend systems. The Samaj management system is implemented on the backend, including community creation, member management, and administrative functionalities. Event management APIs are developed with full CRUD capabilities, image upload support, and reaction systems.

On the frontend, the event listing and detail screens are implemented with comprehensive functionality including event creation forms, image upload capabilities, and reaction mechanisms. Profile management screens are developed allowing users to update personal information and profile pictures. The member search functionality is implemented with real-time search capabilities and result filtering.

API integration is completed for all core features, with comprehensive error handling and user feedback mechanisms. Data synchronization strategies are implemented to ensure consistent information across different application screens. Performance optimization begins with focus on image loading, API response handling, and memory management.

Week 6: Advanced Features - Relationship Tree Implementation

Week six is dedicated to implementing the most complex feature of the application - the relationship tree system. The backend relationship management system is developed with sophisticated algorithms for handling complex family structures, generation level calculations, and relationship validation. The relationship request system is implemented with approval workflows and notification mechanisms.

The Android relationship tree visualization is developed using custom view components and drawing algorithms. Both tree view and list view implementations are created to provide different perspectives on family relationships. Interactive features including node selection, tree navigation, and member detail display are implemented.

The relationship request management interface is developed, allowing users to send, approve, and reject relationship requests. Integration between the relationship system and user profiles is completed, ensuring seamless data flow and consistency. Complex validation logic is implemented to prevent invalid relationships and circular references.

Week 7: Feature Completion and Enhancement

The seventh week focuses on completing all remaining features and enhancing existing functionality. The event reaction system is finalized with real-time count updates and user

interaction feedback. Advanced search functionality is implemented with multiple search criteria and result optimization.

User interface enhancements are implemented including animations, transitions, and improved visual feedback. The auto-scroll functionality for the event carousel is developed and integrated. Image compression and optimization algorithms are implemented to improve performance and reduce storage requirements.

Administrative features are completed including community management tools and user administration capabilities. Data validation and security measures are enhanced across all application components. Performance optimization continues with focus on database query efficiency and mobile application responsiveness.

Week 8: Integration Testing and Quality Assurance

Week eight is dedicated to comprehensive testing and quality assurance activities. Integration testing is performed across all application components, ensuring seamless interaction between frontend and backend systems. User acceptance testing scenarios are executed with focus on real-world usage patterns and edge cases.

Performance testing is conducted to validate system responsiveness and scalability. Security testing is performed to identify and address potential vulnerabilities. Bug fixing and optimization activities are prioritized based on severity and impact assessments.

Code review processes are implemented to ensure code quality and maintainability standards. Documentation is updated to reflect final implementation details and system architecture. User interface refinements are made based on testing feedback and usability assessments.

Week 9: Finalization and Documentation

The final week focuses on project completion and comprehensive documentation. Final system testing is performed to ensure all features function correctly and meet specified requirements. Performance benchmarking is completed to validate system capabilities against defined metrics.

Comprehensive project documentation is finalized including technical specifications, user manuals, and deployment guides. The project presentation is prepared with demonstrations of key features and technical achievements. Final code cleanup and optimization are performed to ensure production readiness.

Deployment preparation activities are completed including environment configuration and system requirements documentation. Knowledge transfer materials are prepared for future maintenance and enhancement activities. Project retrospective is conducted to identify lessons learned and improvement opportunities.

6.3 Resource Allocation and Team Responsibilities

Team Structure and Role Definition

The two-person development team is structured to maximize efficiency and minimize overlap while ensuring comprehensive coverage of all project aspects. The team composition leverages individual strengths while maintaining collaborative development practices throughout the project lifecycle.

Team Member 1 (Backend - Me): Primary responsibility for backend development using Spring Boot framework, database design and implementation, API development and testing, security implementation, and system architecture design. Secondary responsibilities include integration testing, deployment configuration, and technical documentation. This role requires deep understanding of server-side technologies, database management, and security protocols.

Team Member 2 (Frontend - Prit): Primary responsibility for Android application development, user interface design and implementation, API integration, user experience optimization, and mobile-specific functionality implementation. Secondary responsibilities include user testing coordination, mobile performance optimization, and user documentation creation. This role requires expertise in Android development, UI/UX design principles, and mobile application best practices.

6.4 Risk Management and Mitigation Strategies

Technical Risk Mitigation

The compressed 9-week timeline presents significant technical risks that require proactive management. Database performance risks are mitigated through early optimization, proper indexing strategies, and query performance monitoring. Android compatibility risks are addressed through comprehensive testing on multiple devices and API levels, with fallback strategies for problematic scenarios.

Integration complexity risks are managed through early and frequent integration testing, comprehensive API documentation, and mock service implementation for parallel development. Performance risks are mitigated through continuous monitoring, optimization checkpoints, and scalable architecture design.

Project Management Risk Mitigation

Time constraint risks are managed through agile development practices, feature prioritization, and scope management flexibility. Team coordination risks are addressed through daily communication protocols, clear responsibility definitions, and regular progress reviews. Requirement change risks are mitigated through stakeholder engagement, change control processes, and impact assessment procedures.

6.5 Quality Assurance and Testing Strategy

Testing Framework Implementation

The testing strategy encompasses multiple levels of validation to ensure system reliability and user satisfaction. Unit testing frameworks are implemented for both backend (JUnit 5) and frontend (Android Testing Framework) components. Integration testing protocols are established for API validation and system interaction verification.

User acceptance testing procedures are defined with specific scenarios and success criteria. Performance testing benchmarks are established for response times, system throughput, and resource utilization. Security testing protocols are implemented to validate authentication, authorization, and data protection mechanisms.

Continuous Quality Monitoring

Quality metrics are monitored throughout the development process including code coverage, bug density, and performance benchmarks. Regular code reviews ensure adherence to coding standards and best practices. Automated testing integration provides continuous validation of system functionality and regression prevention.

This comprehensive project plan ensures systematic development progress while maintaining quality standards within the 9-week internship timeframe. The detailed weekly breakdown provides clear milestones and deliverables, while risk management strategies address potential challenges proactively. The collaborative team structure maximizes individual strengths while ensuring comprehensive project coverage and successful completion.

7. SYSTEM DESIGN

7.1. System Architecture Design

This chapter provides a comprehensive overview of the architectural and structural components of the A Community Networking and Engagement Platform. Unified Modelling Language (UML) diagrams are used to visually represent different aspects of the system's functionality, flow, and deployment.

7.2. UML Diagrams

7.2.1. E-R Diagram

The ER diagram of **SamajConnect** models a community networking platform where users belong to a *Samaj*, create and react to events, and establish family-like relationships. The Users table connects with Samajs, Events, UserRelationships, and EventReactions to manage profiles, event participation, and social connections. Relationship requests flow through RelationshipRequests.

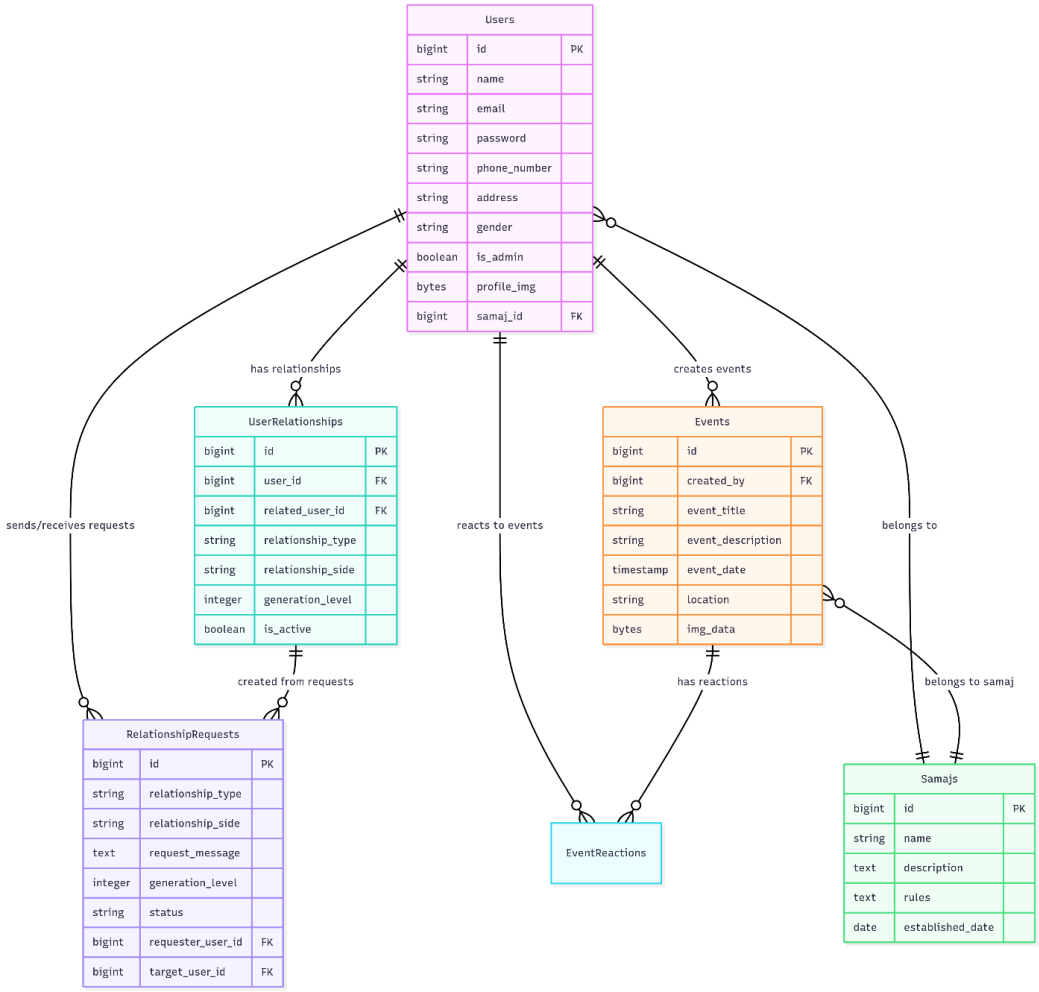


Fig 7.1 E-R Diagram

7.2.2. Use Case Diagram

The Use Case Diagram for *SamajConnect* represents interactions between users (both admins and members) and the system. Admins can create events, manage users, and approve relationship requests. Regular users can register/login, join a Samaj, send relationship requests, view events, and react to them. The diagram highlights key functionalities that promote networking and engagement within the community.

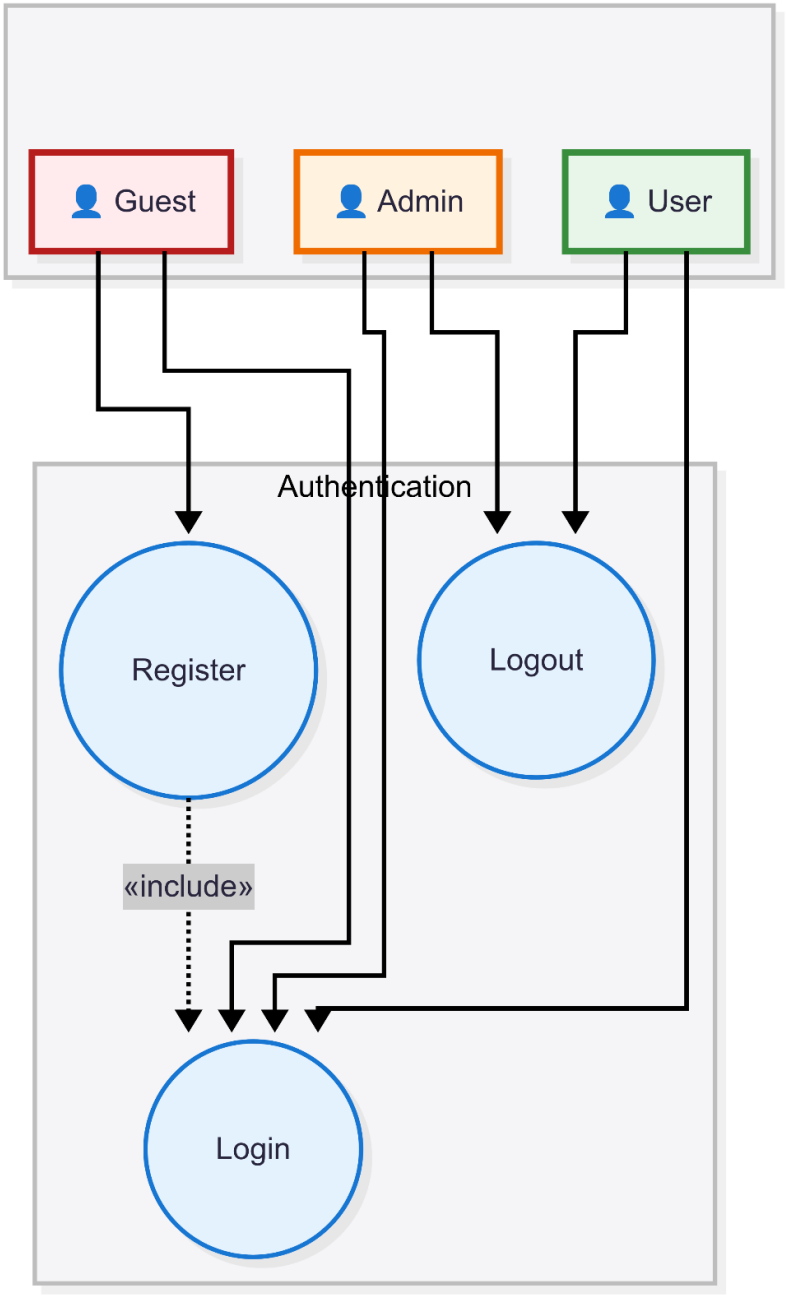


Fig 7.2 Use Case of Login/Register/Log-Out

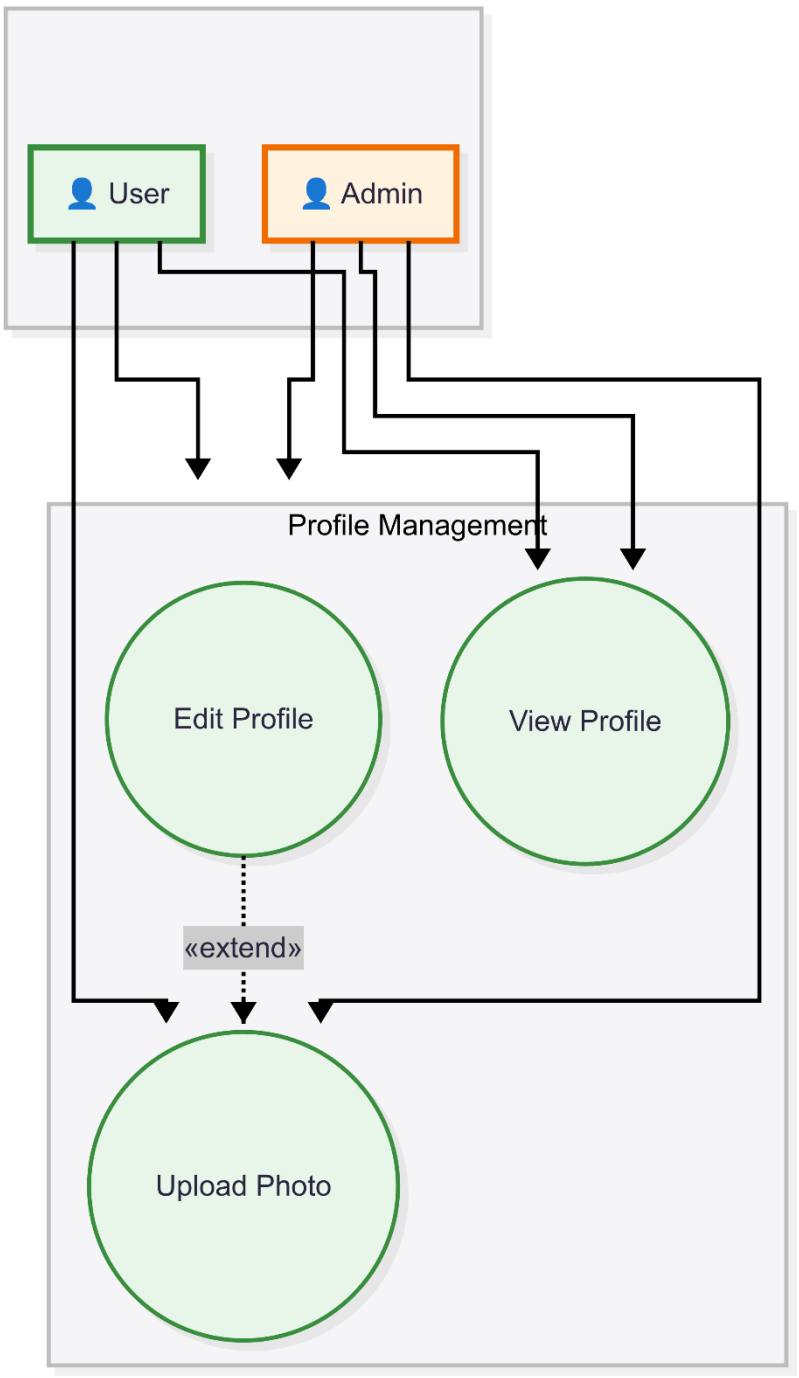


Fig 7.3 Use Case Diagram of Profile

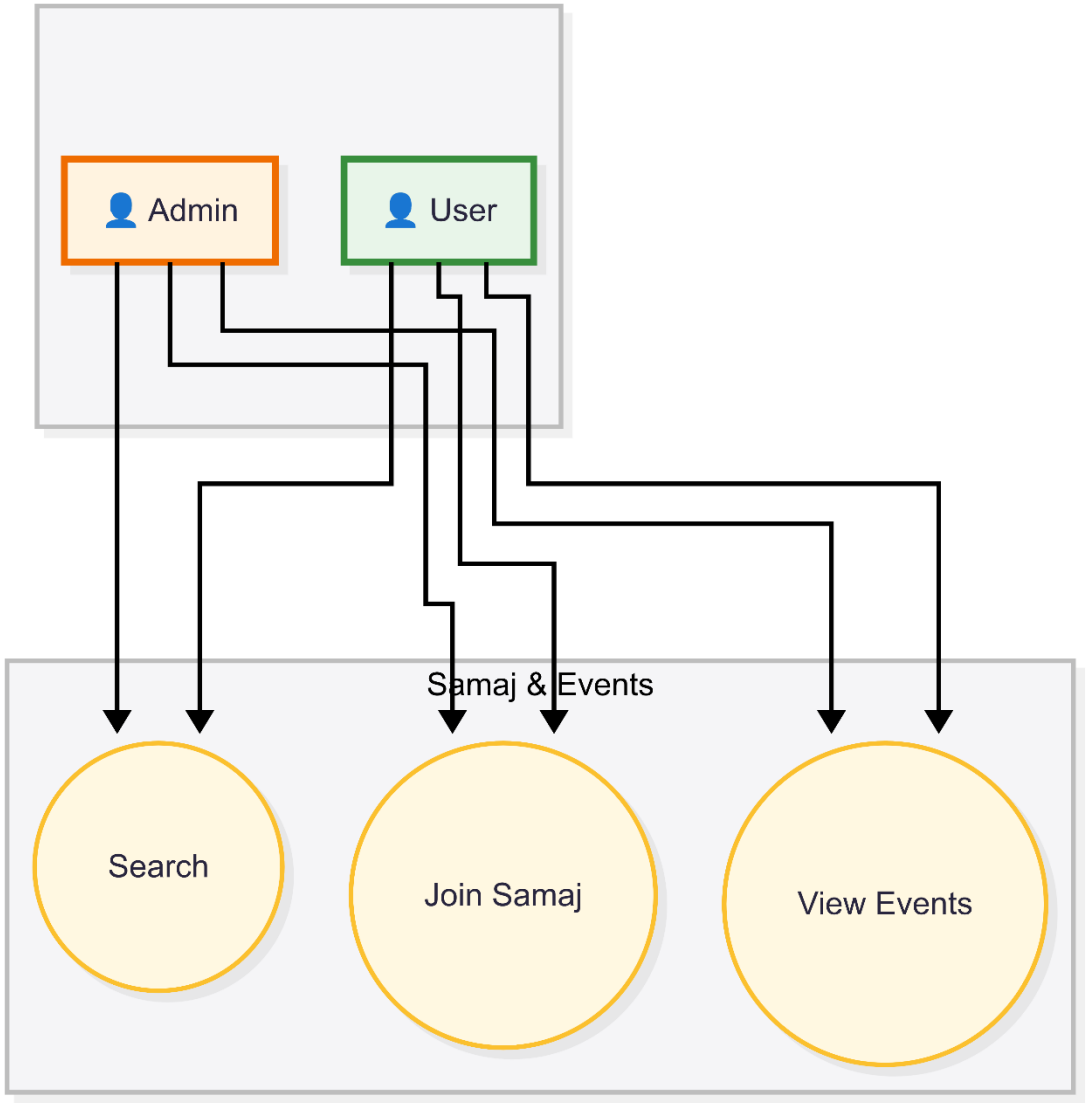


Fig 7.4 Use Case of Join Community

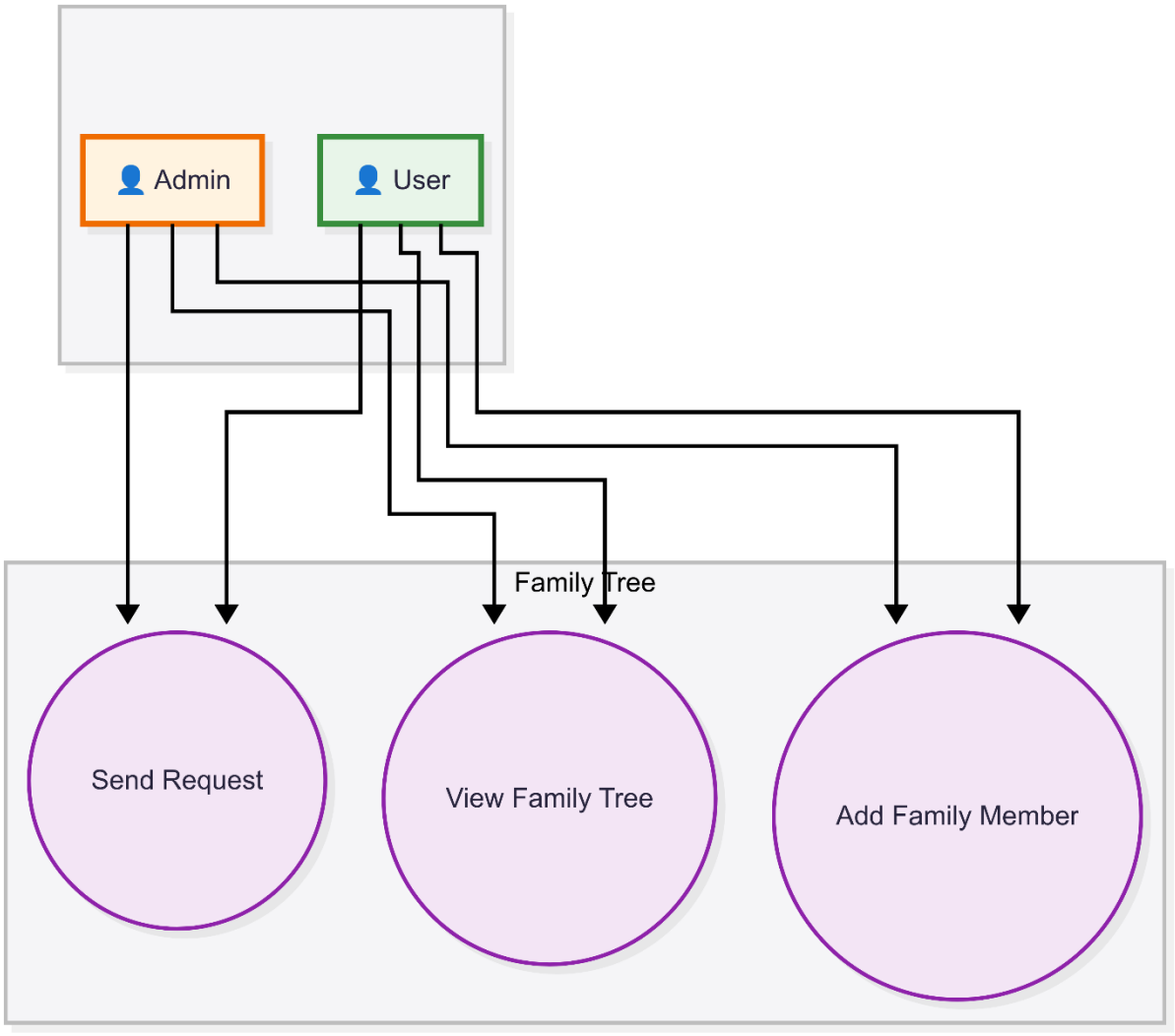


Fig 7.5 Use Case Diagram of Rekationship

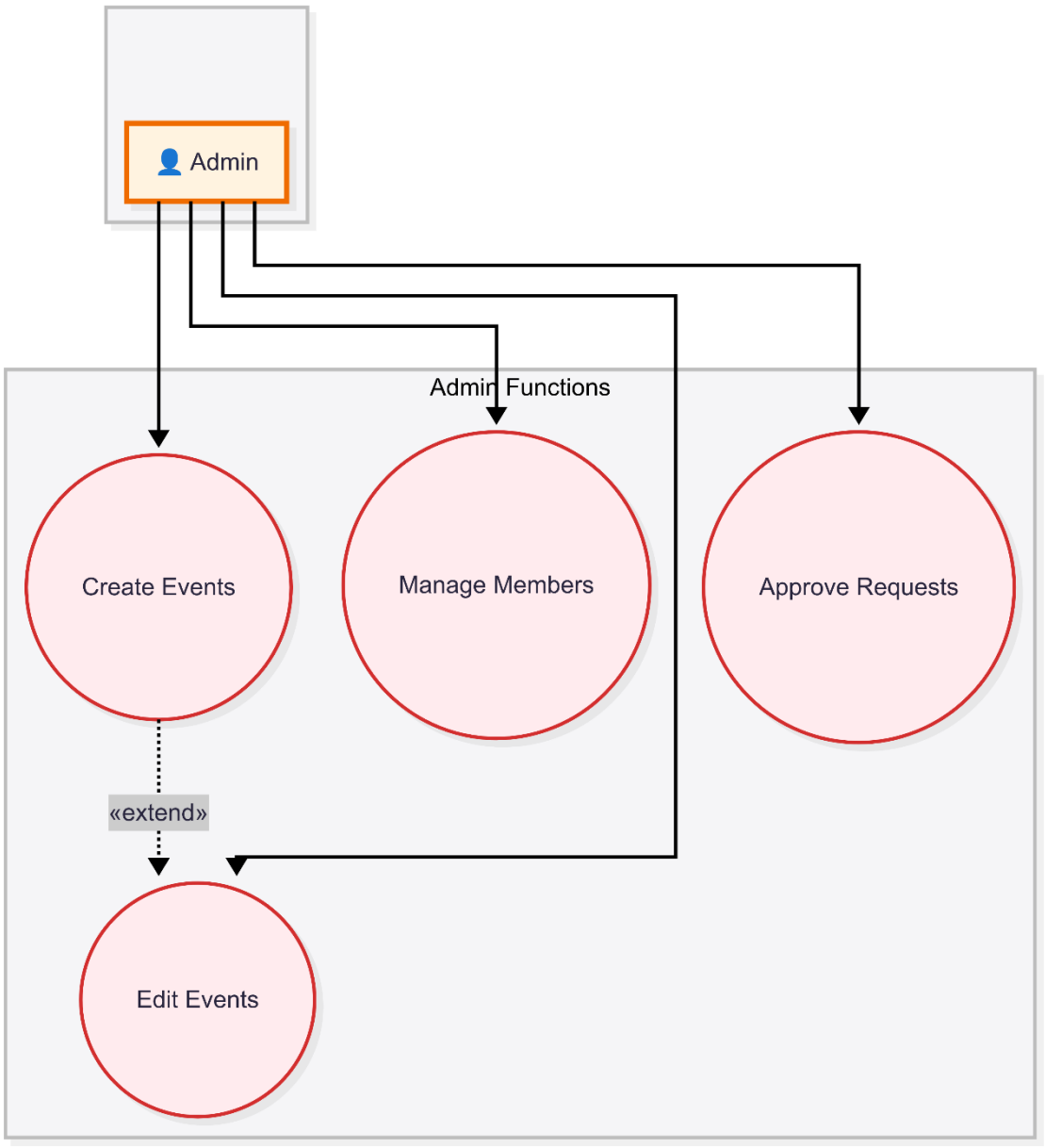


Fig 7.6 Use Case Diagram of Events

7.2.3. Class Diagram

The class diagram defines the structure of the application with classes like User, Samaj, Event, UserRelationship, and RelationshipRequest. It shows attributes and methods, along with relationships such as associations, inheritance, and compositions. This diagram provides a blueprint of how entities interact within the system.

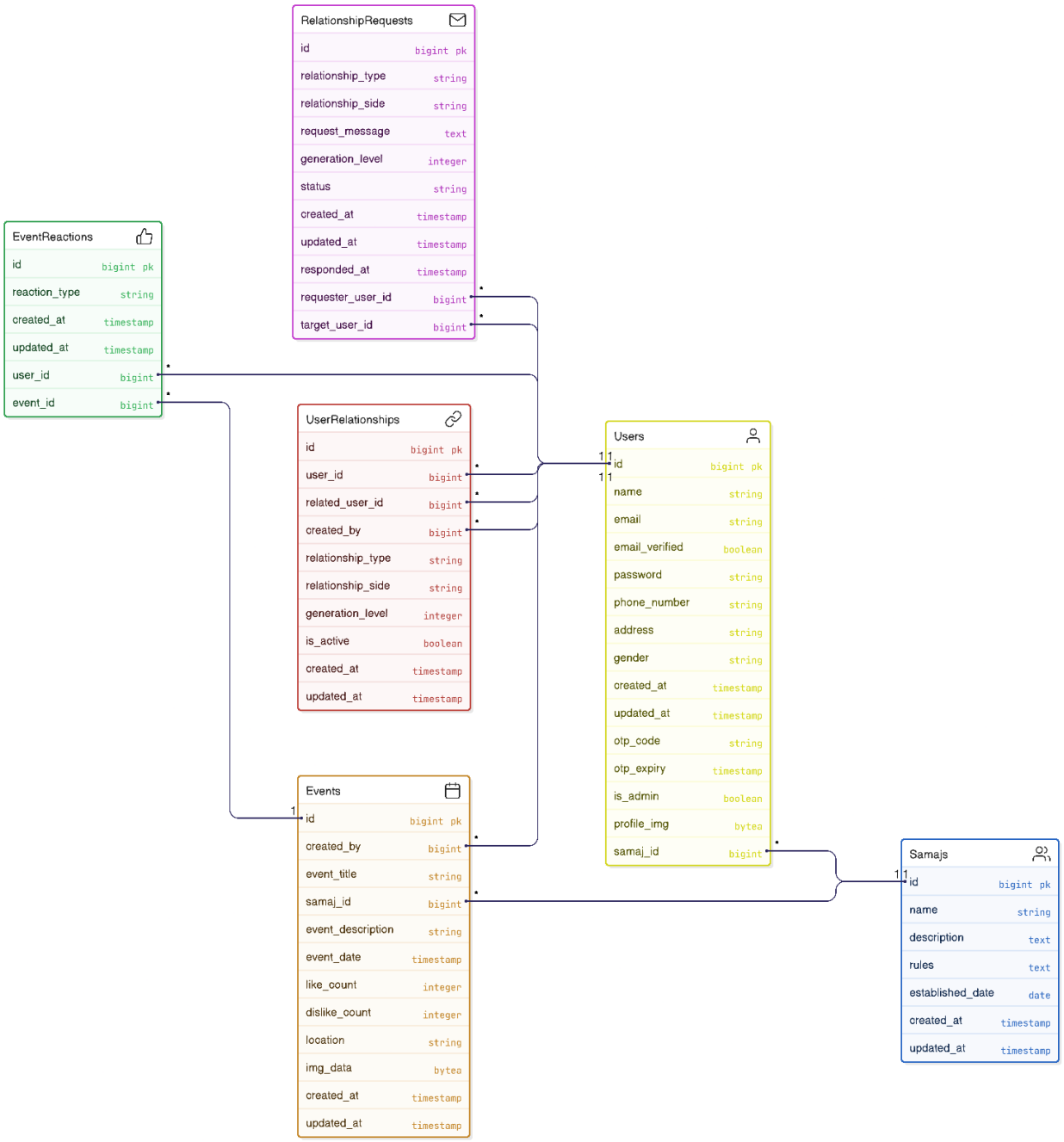


Fig 7.7 Class Diagram

7.2.4. Sequence Diagram

The sequence diagram illustrates how objects interact over time for specific operations like sending a relationship request or reacting to an event. It captures the order of method calls between users, the system, and the database, showcasing the dynamic behavior of the application during execution.

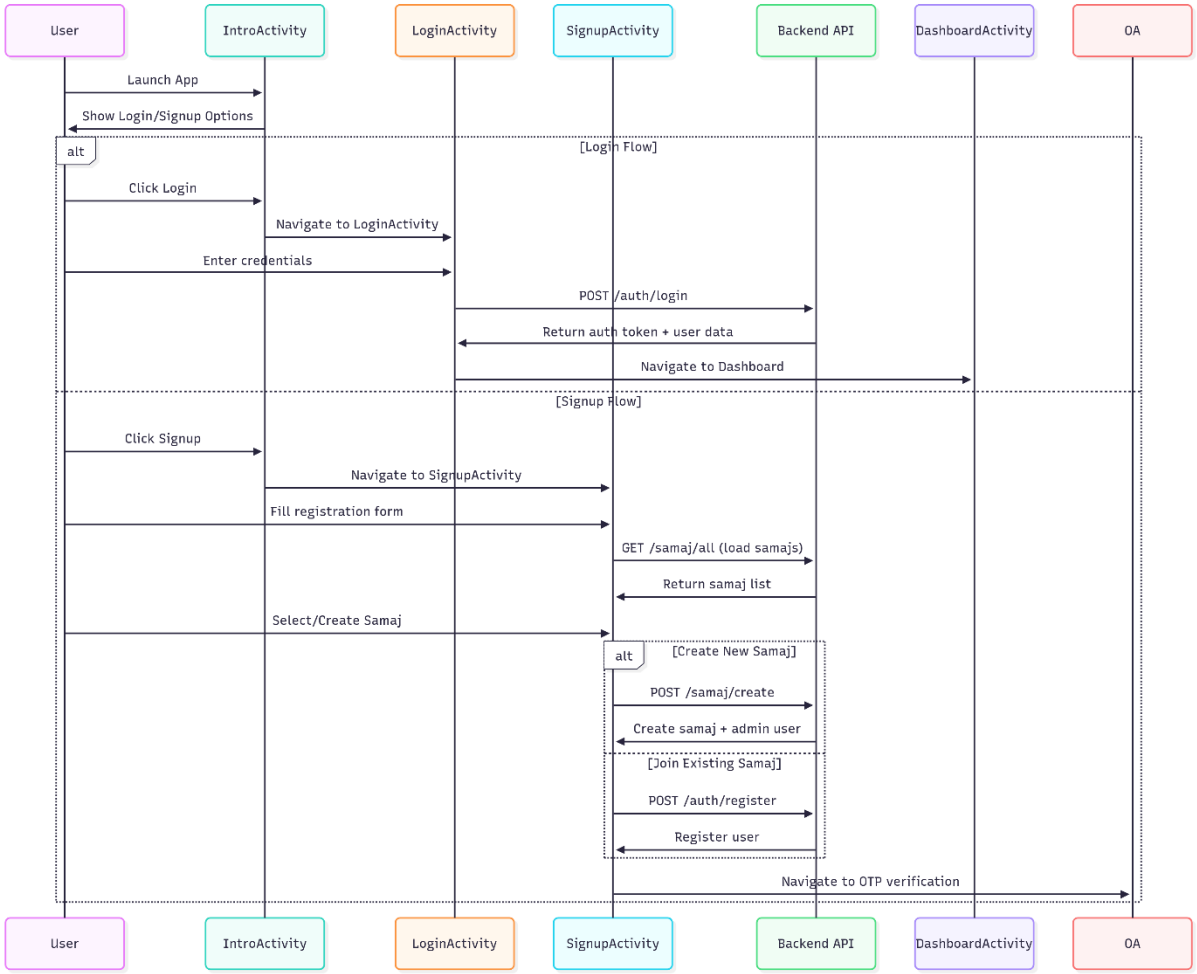


Fig 7.8 Sequence Diagram of User_Auth_Sequence_Diagram

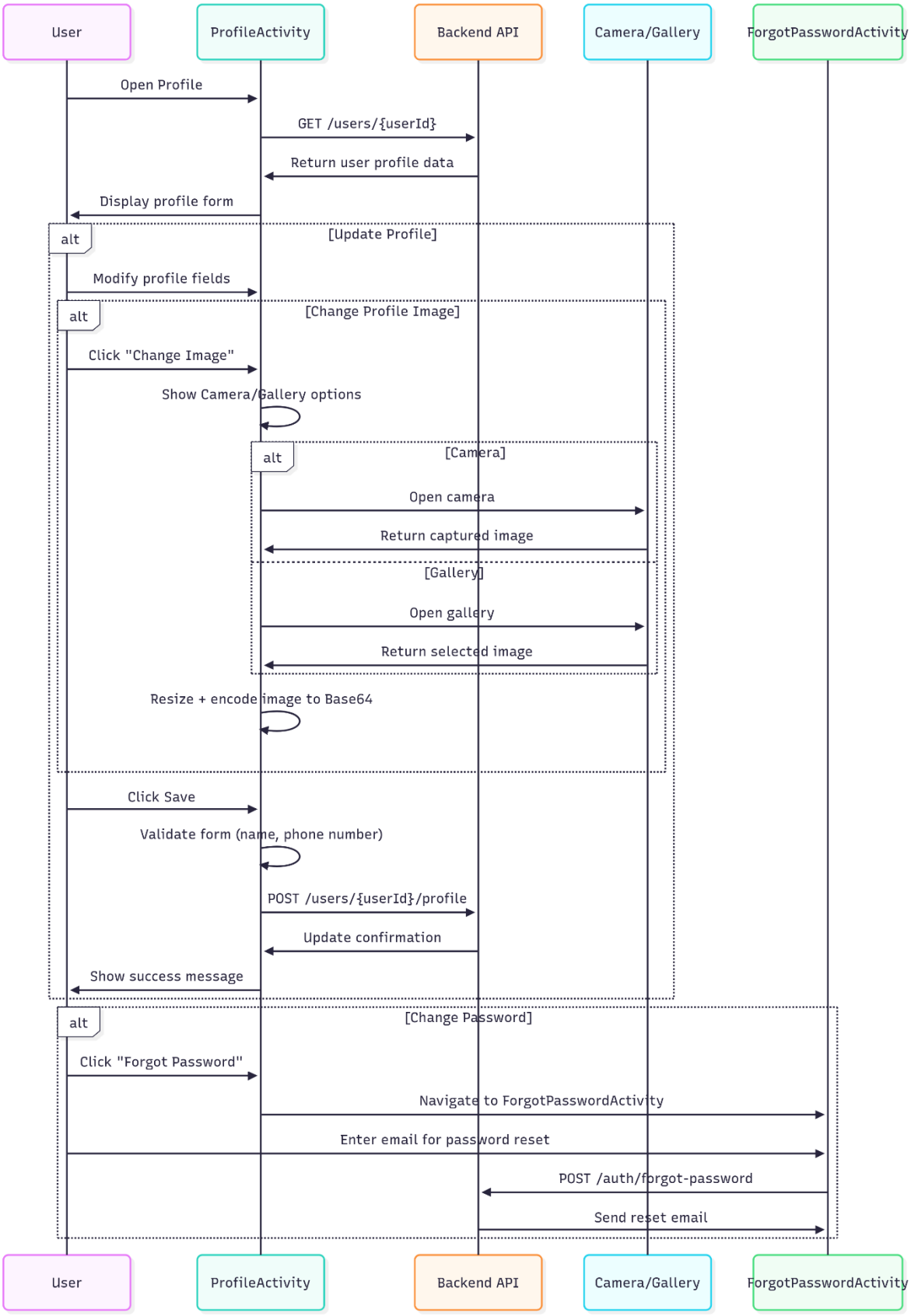


Fig 7.9 Sequence Diagram Profile_Sequence

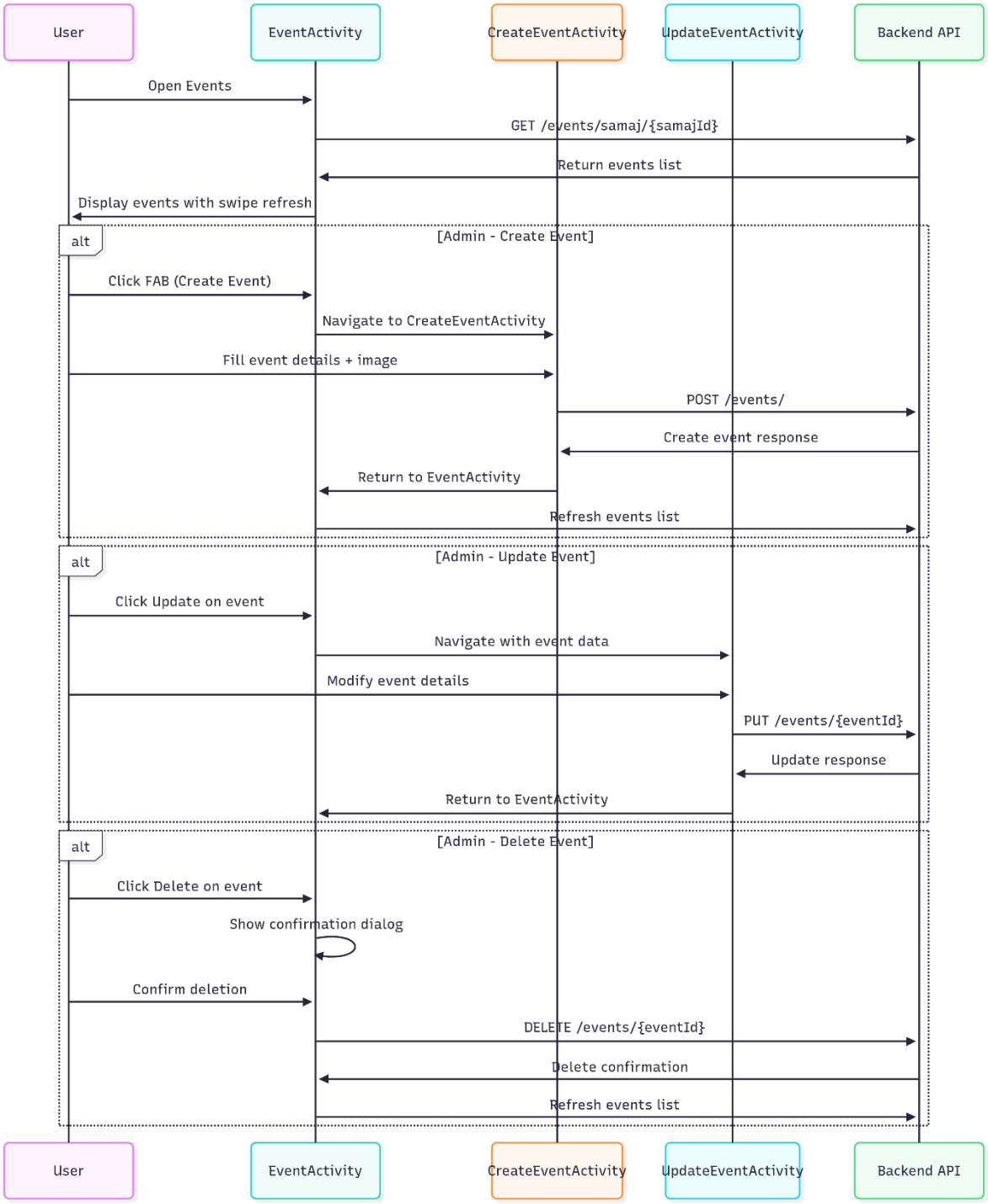


Fig 7.10 Sequence Diagram Event

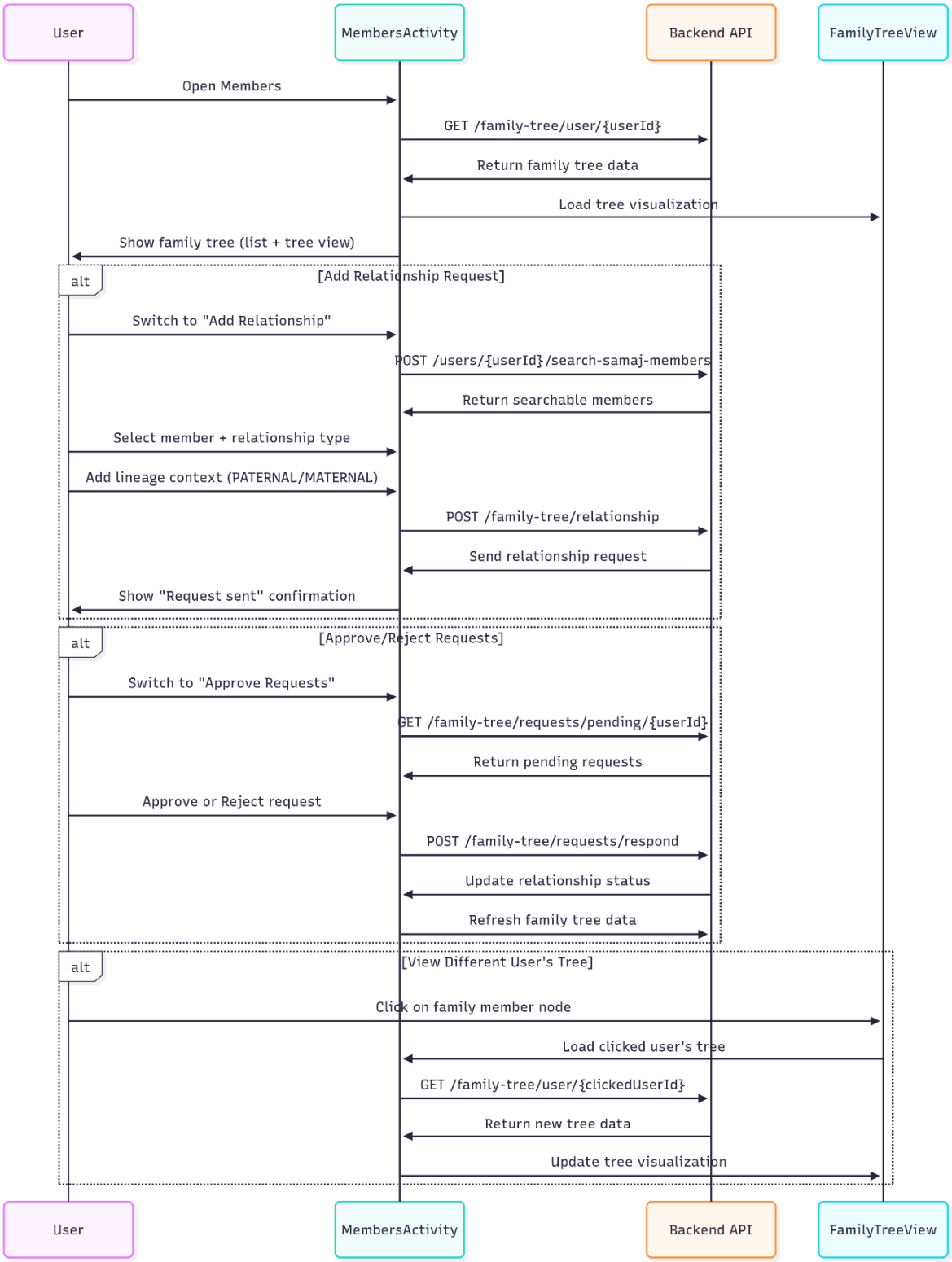


Fig 7.11 Sequence Diagram Family_Tree

7.2.5. Activity Diagram

The activity diagram maps the flow of actions for key user operations such as registration, event participation, and relationship management. It includes decision branches, parallel flows, and user/system actions, clearly showing how users engage with the platform step-by-step.

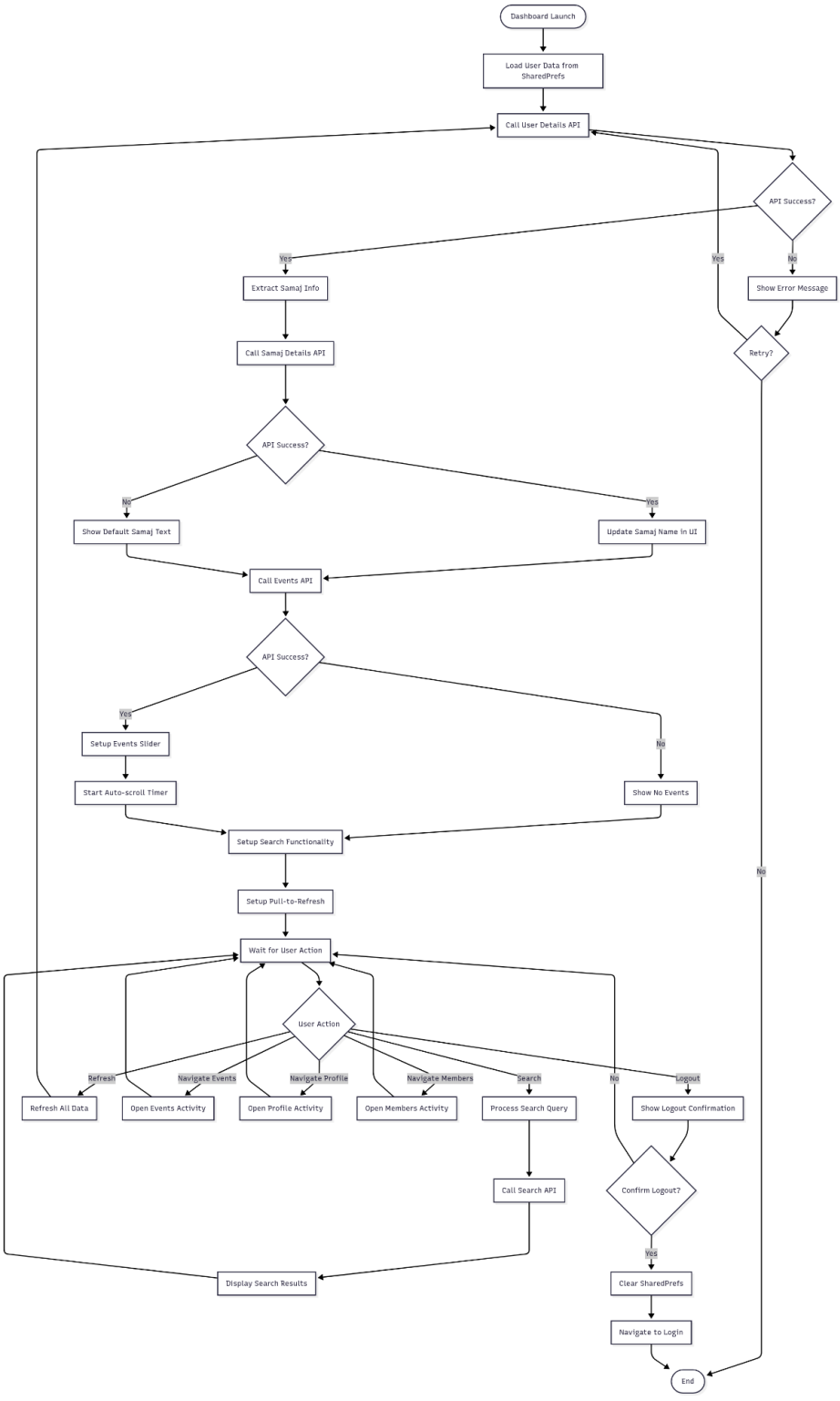


Fig 7.12 Activity Diagram - Dashboard

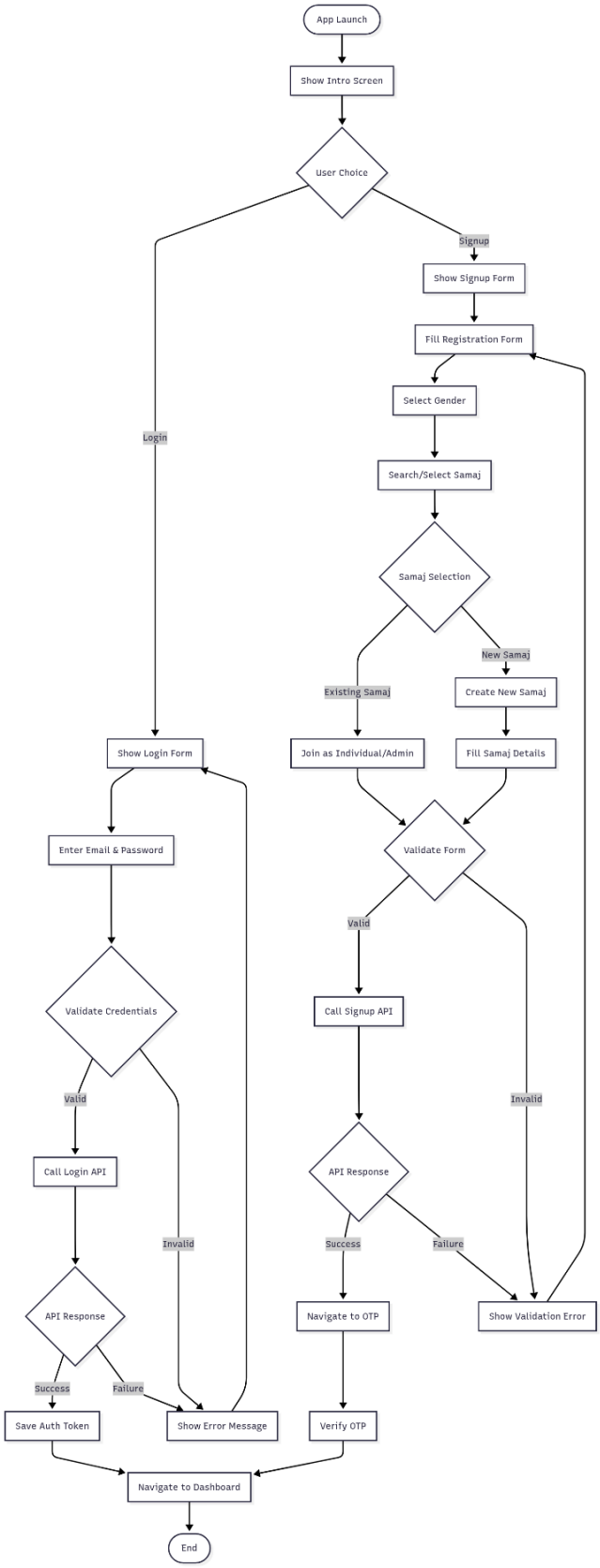


Fig 7.13 Activity Diagram - Authentication

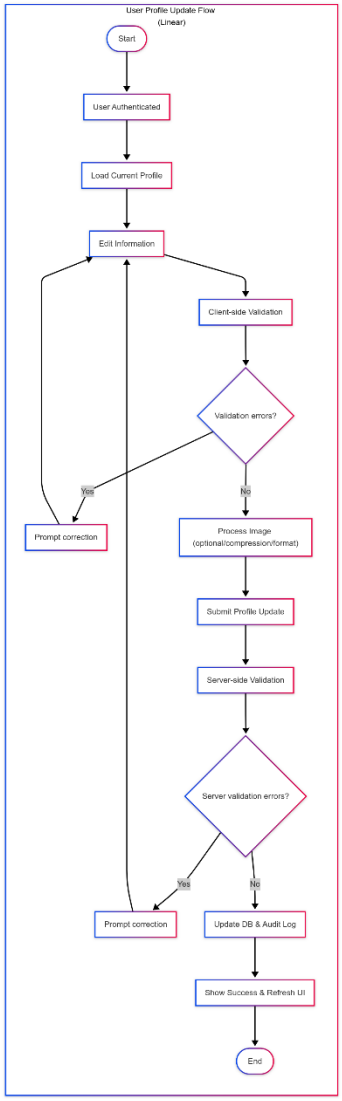


Fig 7.14 Activity Diagram - Profile

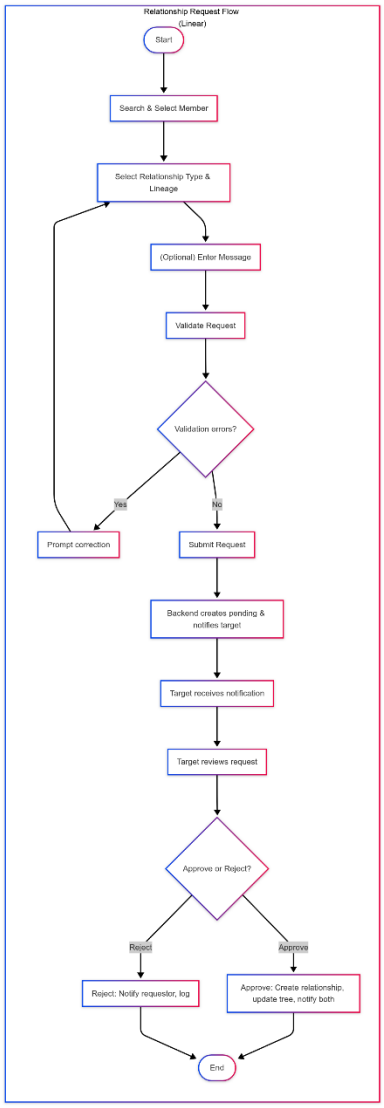


Fig 7.15 Activity Diagram - Relationship

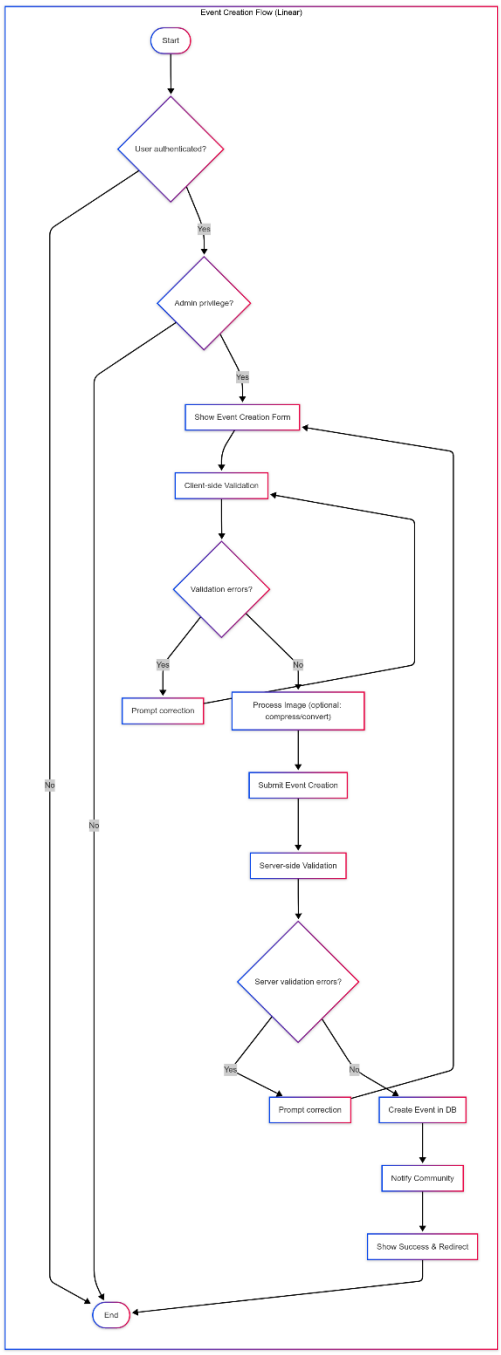


Fig 7.16 Activity Diagram - Event

7.2.6. DFD Diagram

The DFD represents the movement of data between different components like users, relationship modules, events, and the database. It highlights key processes such as "Manage User," "Handle Relationships," and "Process Events," showing how data enters, is processed, and stored in the system.

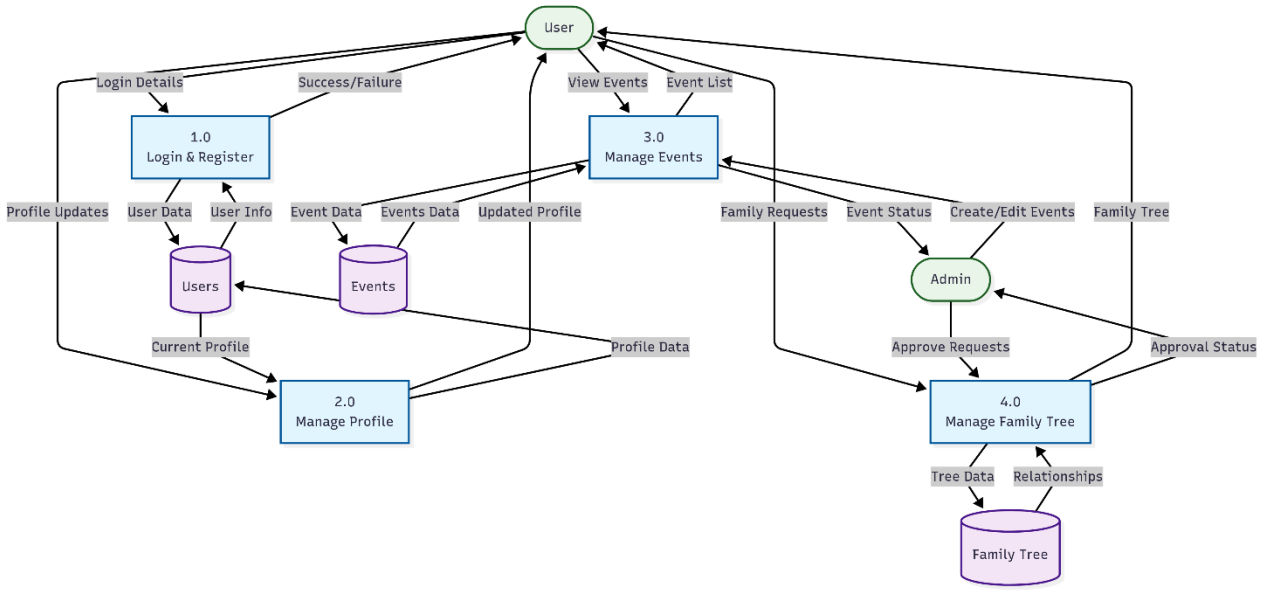


Fig 7.17 Data-Flow-Diagram level-0 Authentication,Profile,Event,Relation

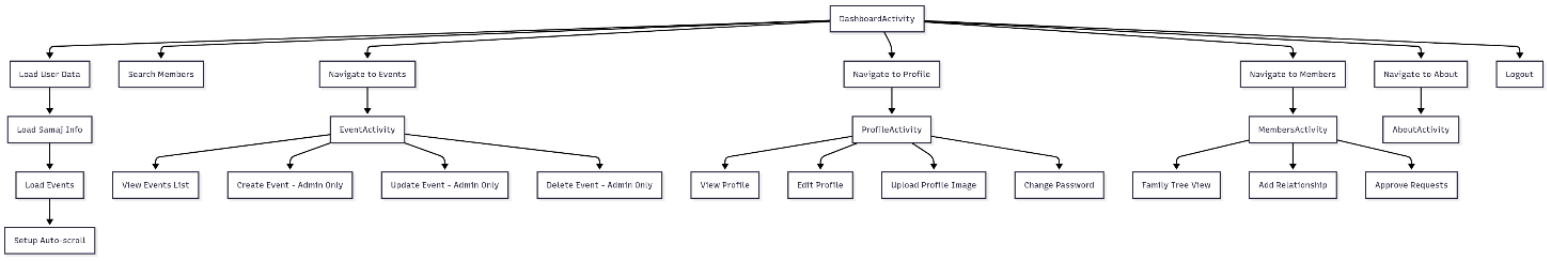


Fig 7.18 Data-Flow-Diagram level-0 Dashboard

7.3. Data Dictionary

Table: users

Column Name	Data Type	Default	Constraints
id	bigint	auto-generated	Primary Key
name	character varying	-	NOT NULL
email	character varying	-	NOT NULL, UNIQUE
password	character varying	-	NOT NULL
phone_number	character varying	-	-
gender	character varying	'OTHER'	NOT NULL
address	character varying	-	-
created_at	timestamp	-	-
updated_at	timestamp	-	-
email_verified	boolean	-	-
otp_code	character varying	-	-
otp_expiry	timestamp	-	-
is_admin	boolean	false	-
profile_img	bytea	-	-
samaj_id	bigint	-	FK → samajs(id)

Table 7.1: Data Dictionary-User

Table: user_relationships

Column Name	Data Type	Default	Constraints
id	bigint	auto-generated	Primary Key
user_id	bigint	-	FK → users(id)
related_user_id	bigint	-	FK → users(id)
created_by	bigint	-	FK → users(id)
relationship_type	character varying	-	CHECK → Valid relationship types (FATHER, MOTHER, etc.)
relationship_side	character varying	-	CHECK → 'PATERNAL', 'MATERNAL', etc.
generation_level	integer	-	NOT NULL
is_active	boolean	-	-
created_at	timestamp	-	-
updated_at	timestamp	-	-

Table 7.2: Data Dictionary- User_Relationships

Table: samajs

Column Name	Data Type	Default	Constraints
id	bigint	auto-generated	Primary Key
name	character varying	-	NOT NULL, UNIQUE
description	text	-	-
rules	text	-	-
established_date	date	-	-
created_at	timestamp	CURRENT_TIMESTAMP	-
updated_at	timestamp	CURRENT_TIMESTAMP	-

Table 7.3: Data Dictionary- samajs

Table: events

Column Name	Data Type	Default	Constraints
id	bigint	auto-generated	Primary Key
event_title	character varying	-	NOT NULL
event_description	character varying	-	-
event_date	timestamp	-	-
created_by	bigint	-	FK → users(id)
samaj_id	bigint	-	FK → samajs(id)
like_count	integer	0	-
dislike_count	integer	0	-
location	character varying	-	-
img_data	bytea	-	-
created_at	timestamp	CURRENT_TIMESTAMP	-
updated_at	timestamp	CURRENT_TIMESTAMP	-

Table 7.4: Data Dictionary- events

Table: event_reactions

Column Name	Data Type	Default	Constraints
id	bigint	auto-generated	Primary Key
user_id	bigint	-	NOT NULL, FK → users(id)
event_id	bigint	-	NOT NULL, FK → events(id)
reaction_type	character varying	-	CHECK: 'LIKE' or 'DISLIKE'
created_at	timestamp	CURRENT_TIMESTAMP	-
updated_at	timestamp	CURRENT_TIMESTAMP	-

Table 7.5: Data Dictionary- event_reactions

Table: relationship_requests

Column Name	Data Type	Default	Constraints
id	bigint	auto-generated	Primary Key
requester_user_id	bigint	-	FK → users(id)
target_user_id	bigint	-	FK → users(id)
relationship_type	character varying	-	NOT NULL
relationship_side	character varying	-	NOT NULL
request_message	text	-	-
generation_level	integer	0	-
status	character varying	'PENDING'	CHECK: 'PENDING', 'APPROVED', 'REJECTED', 'CANCELLED'
created_at	timestamp	now ()	-
updated_at	timestamp	now ()	-
responded_at	timestamp	-	-

Table 7.6: Data Dictionary- relationship_requests

7.4. Screenshots

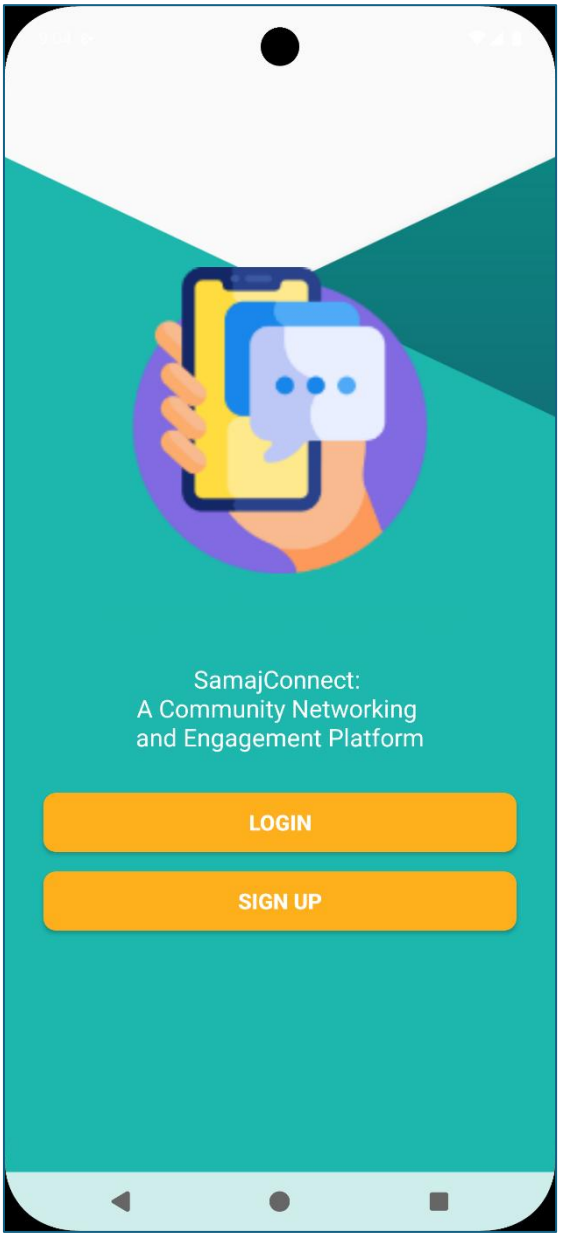


Fig 7.19 Login Screen

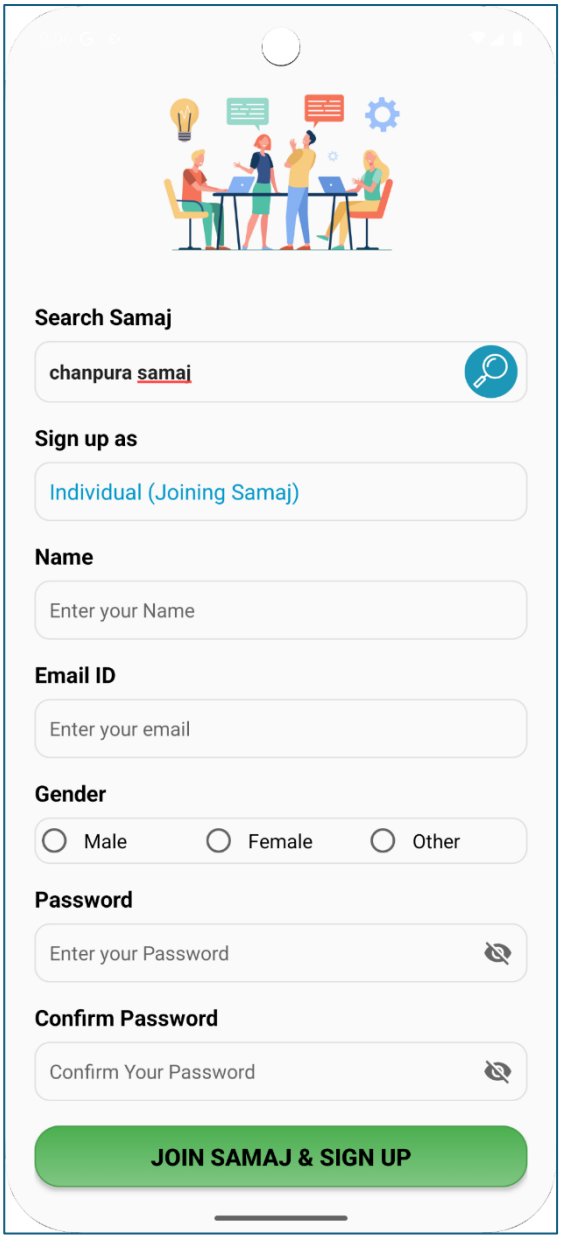


Fig 7.20 Sign-up_Join

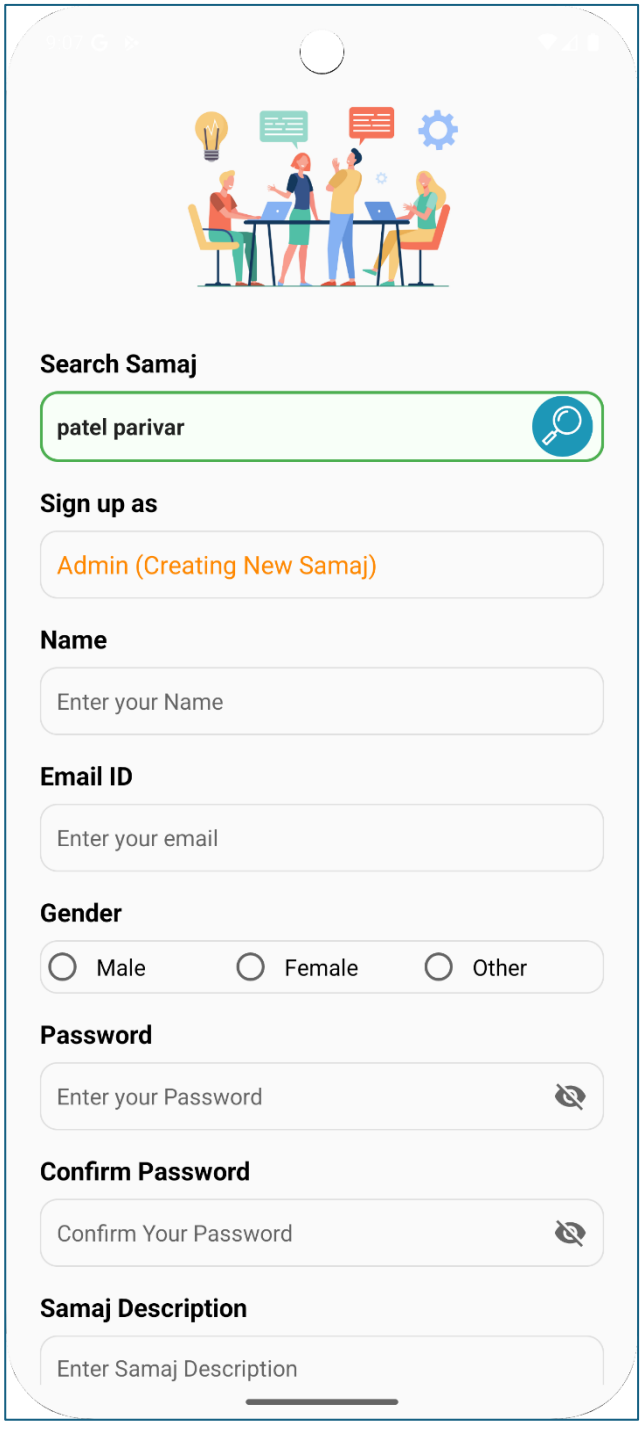


Fig 7.21 Sign-up_Register

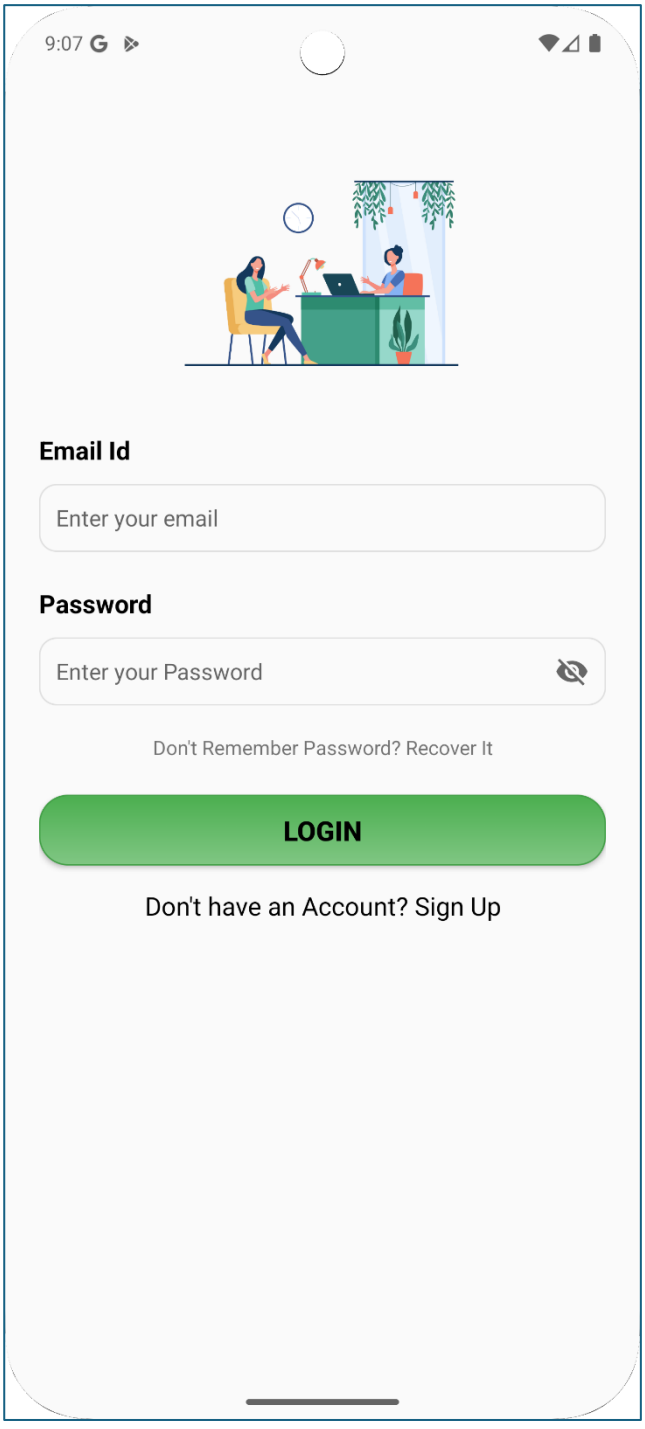


Fig 7.22 login

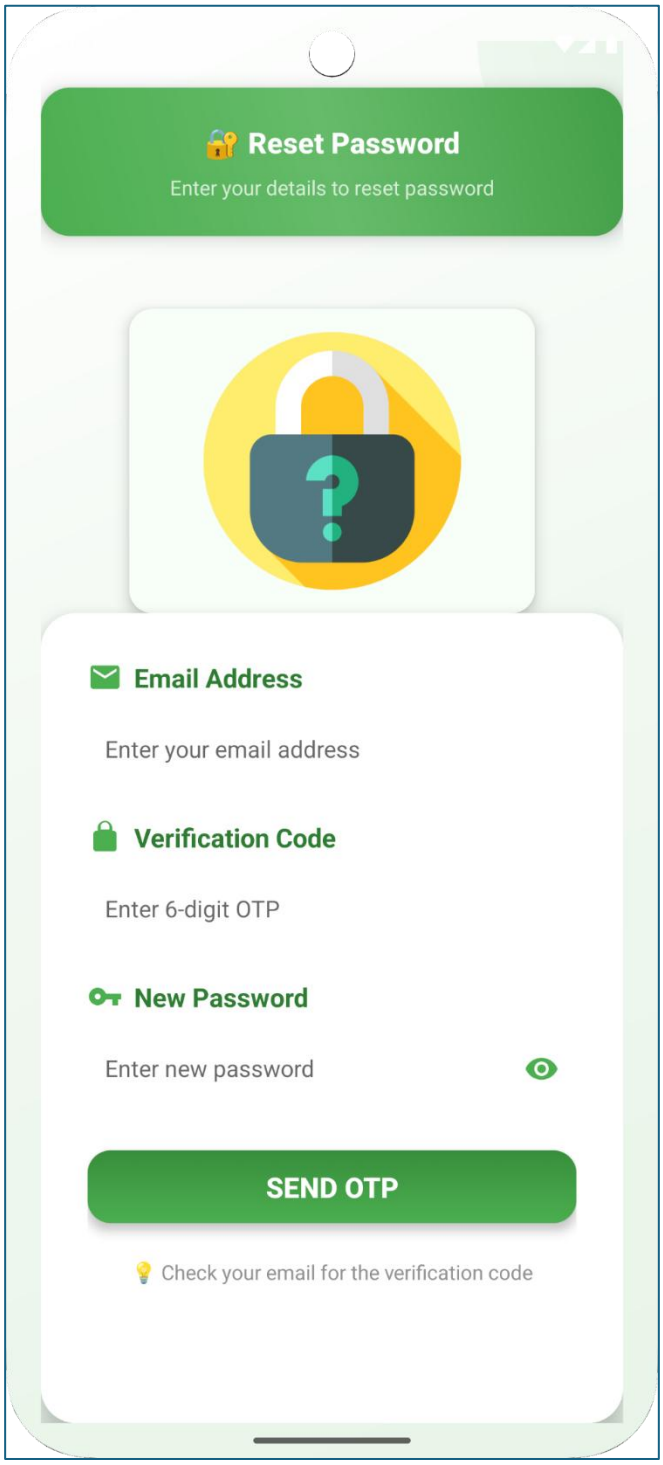


Fig 7.23 Forgot Password

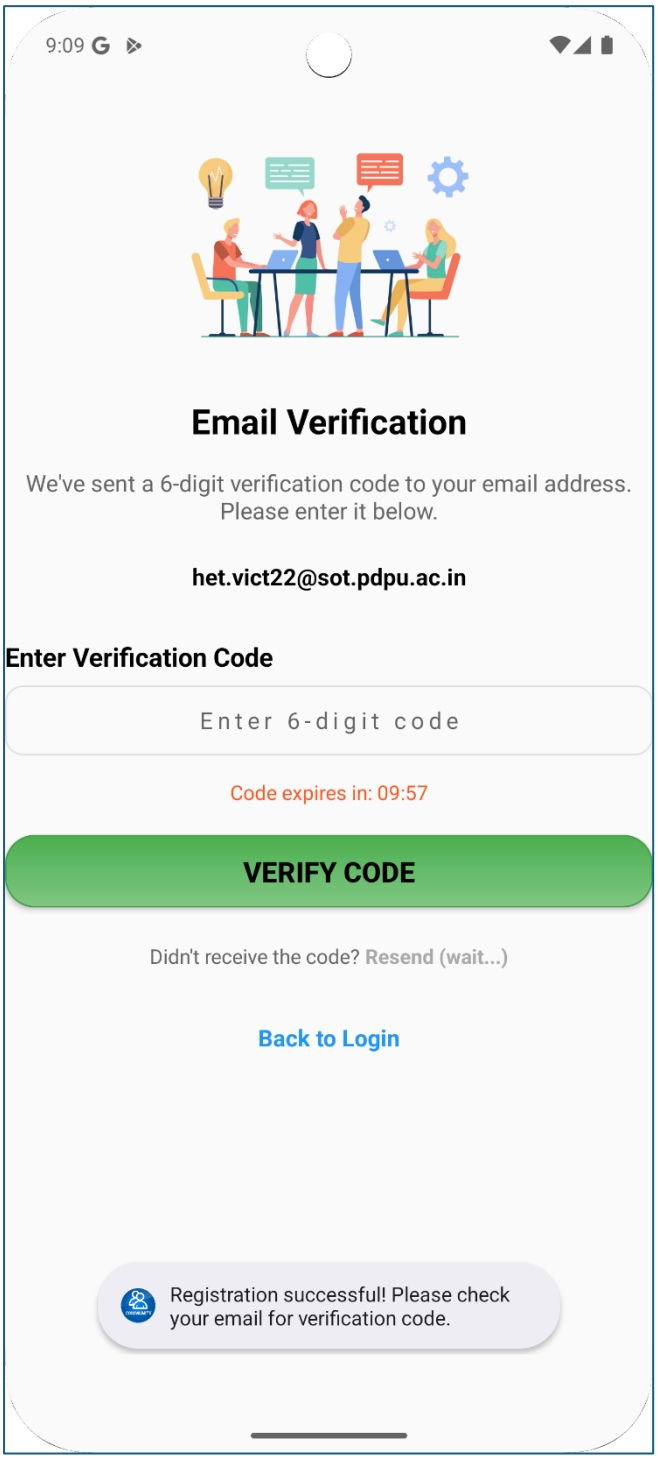


Fig 7.24 Otp Verification

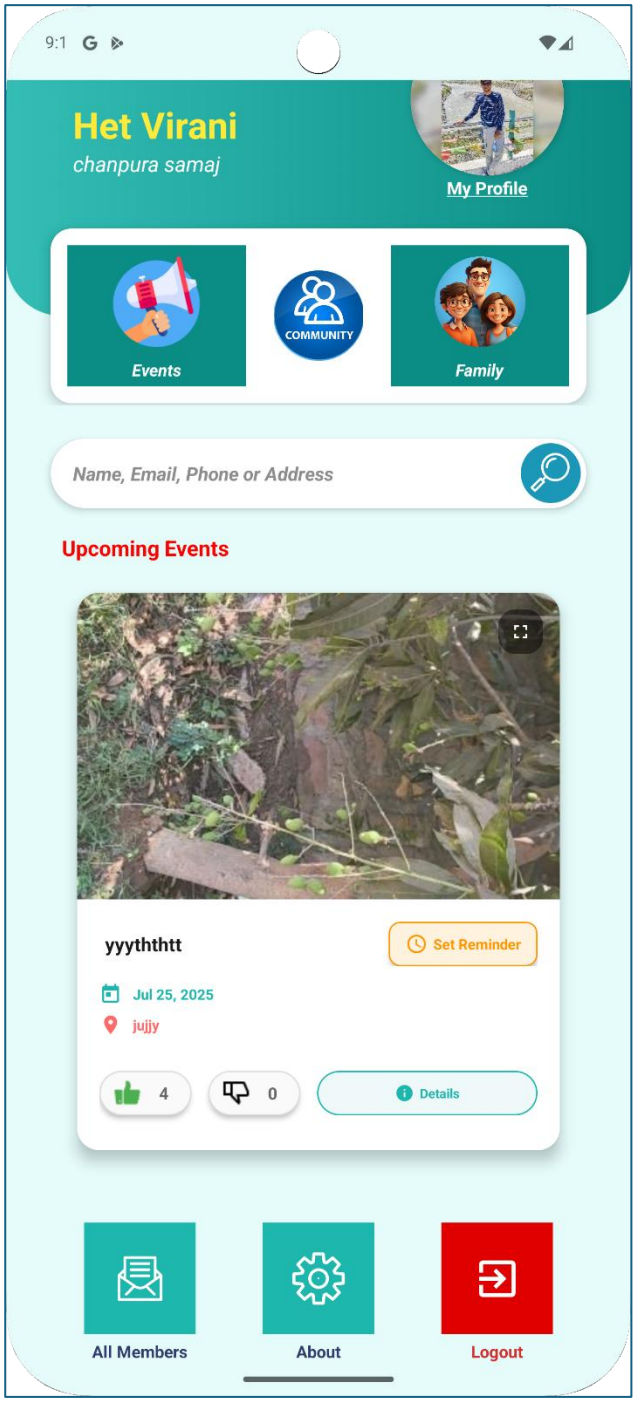


Fig 7.25 Dashboard

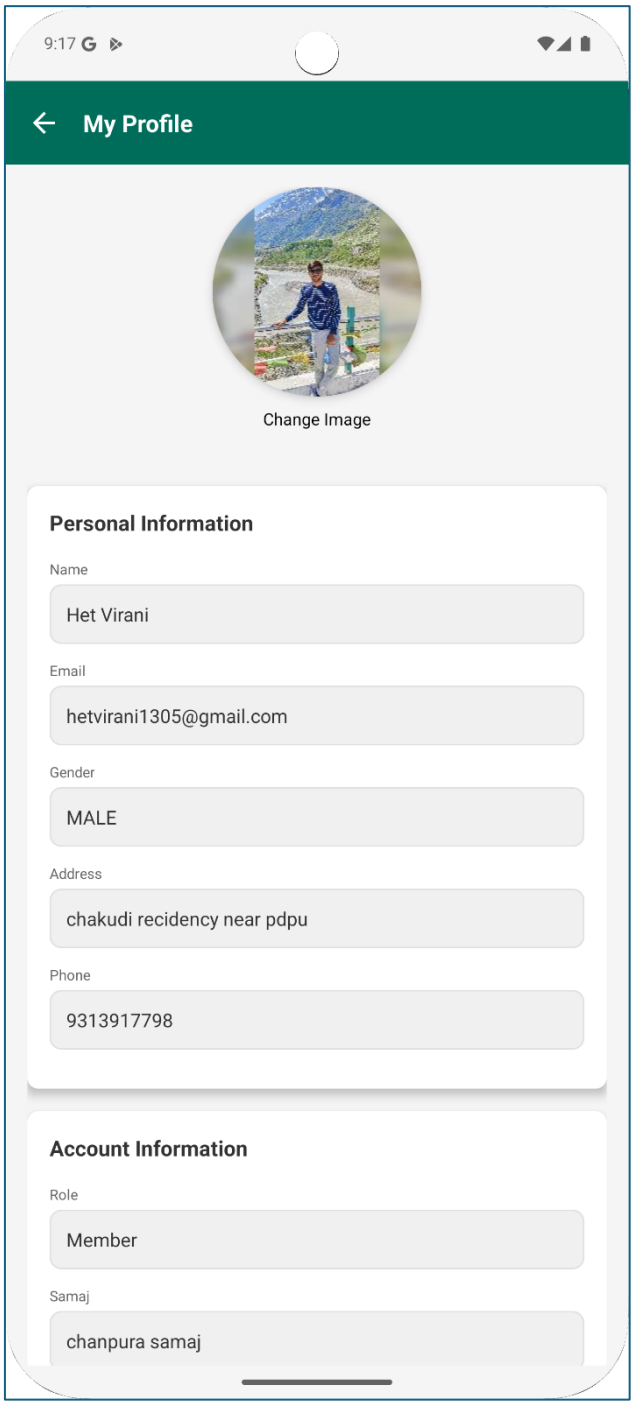


Fig 7.26 Profile Screen

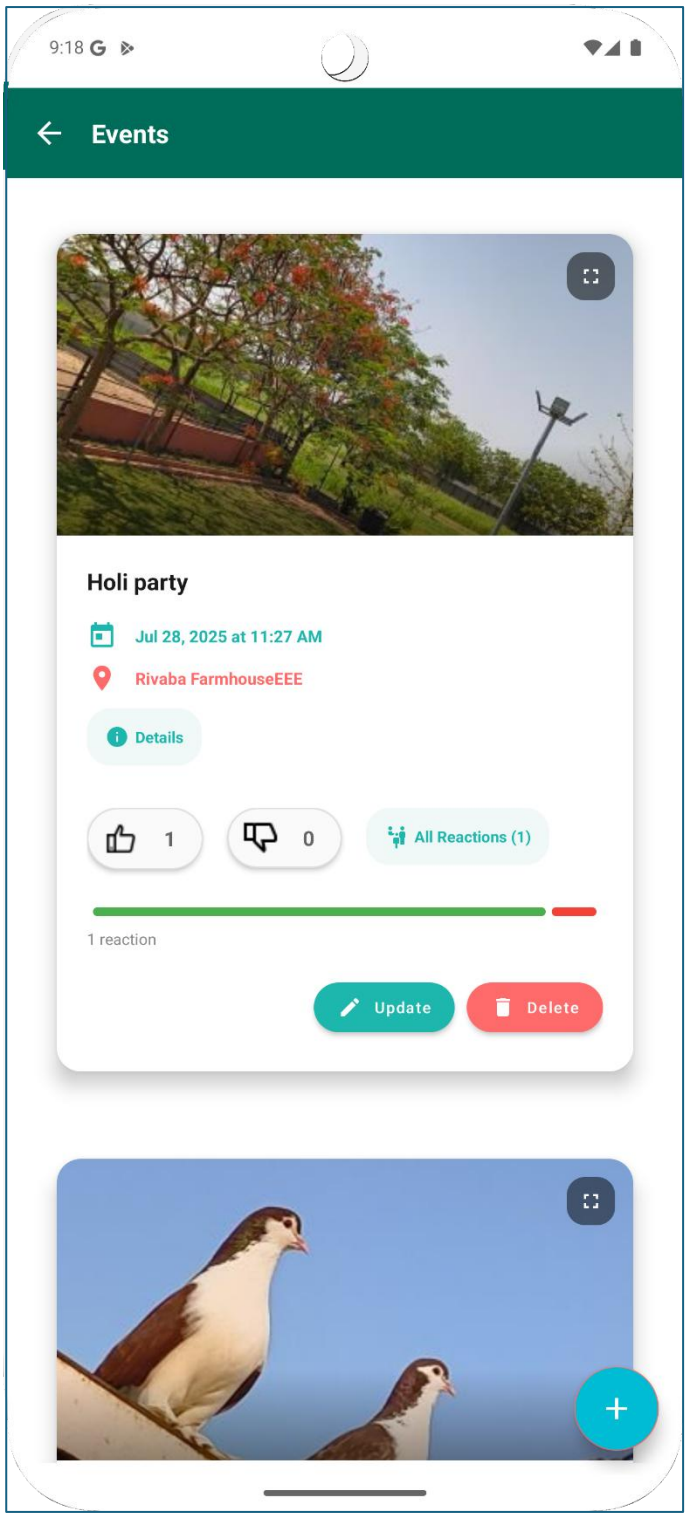


Fig 7.27 Event Page

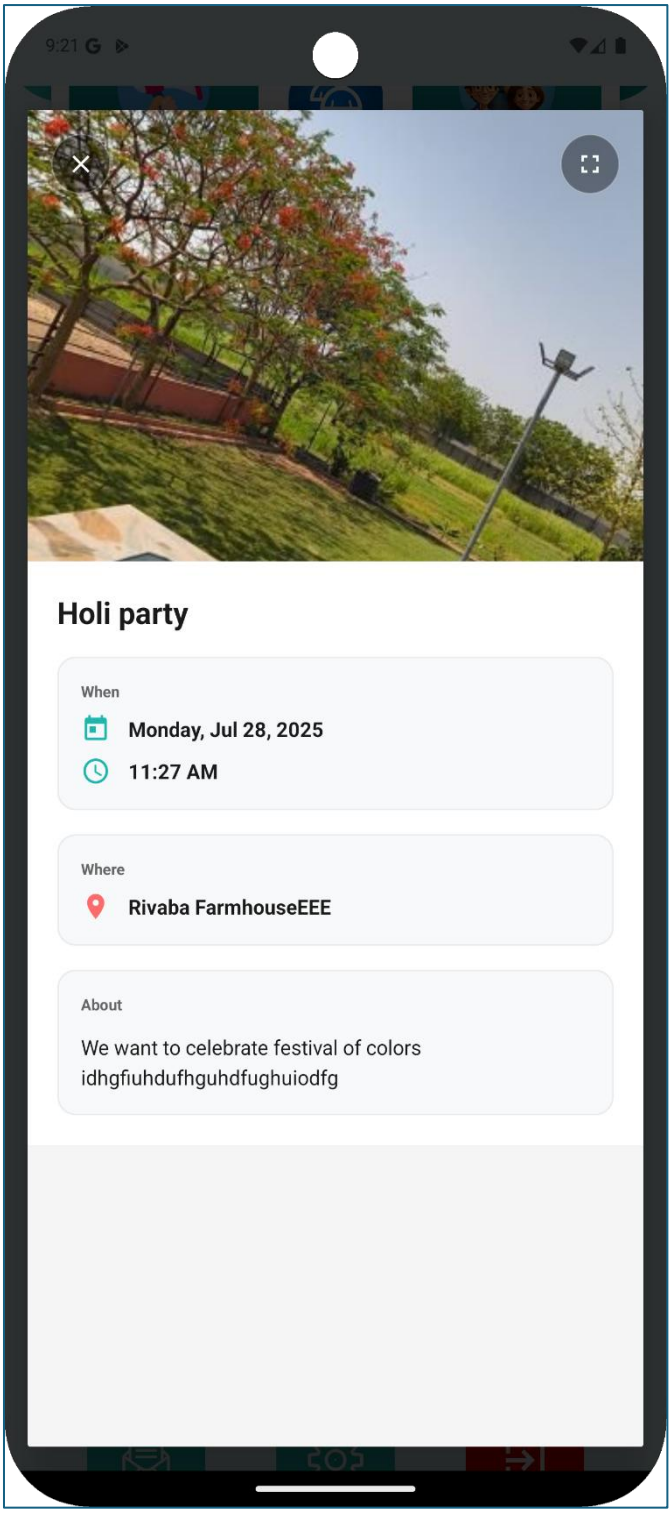


Fig 7.28 Event Full Screen

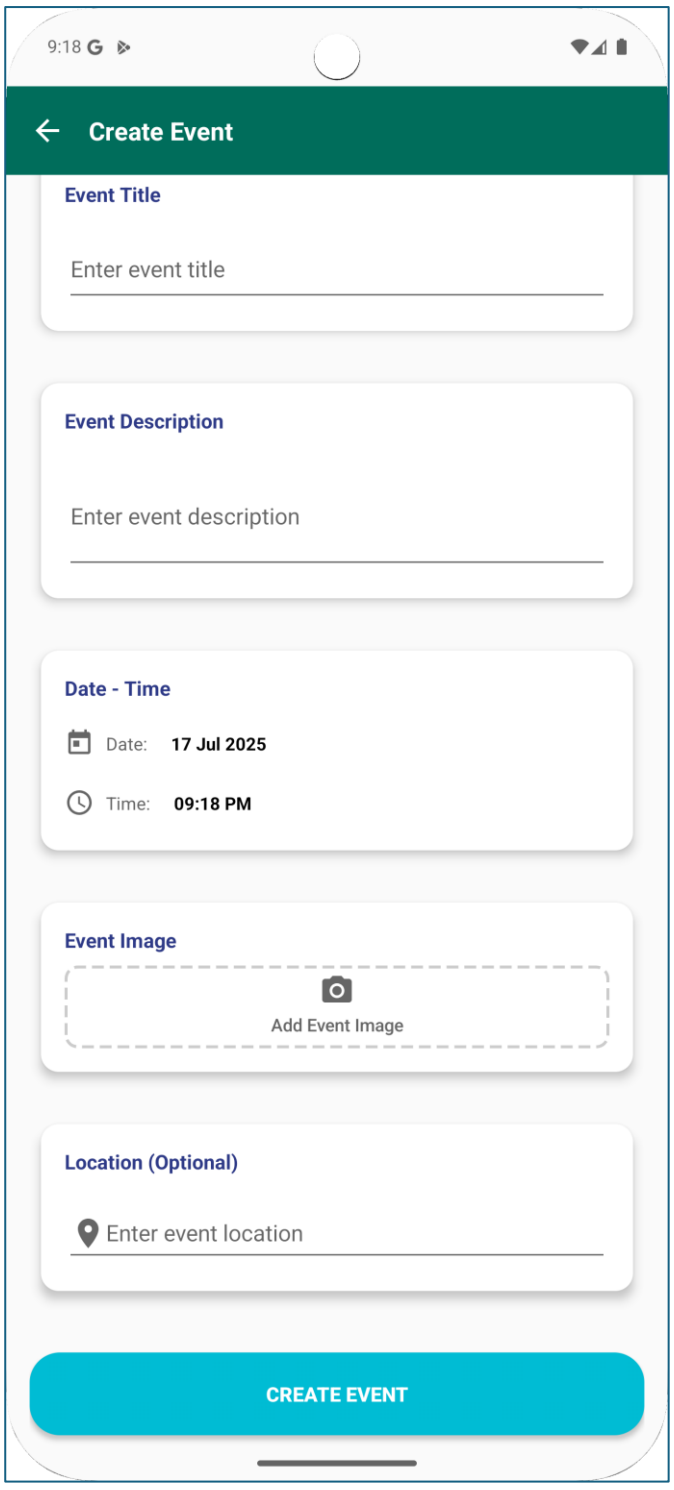


Fig 7.29 Create Event

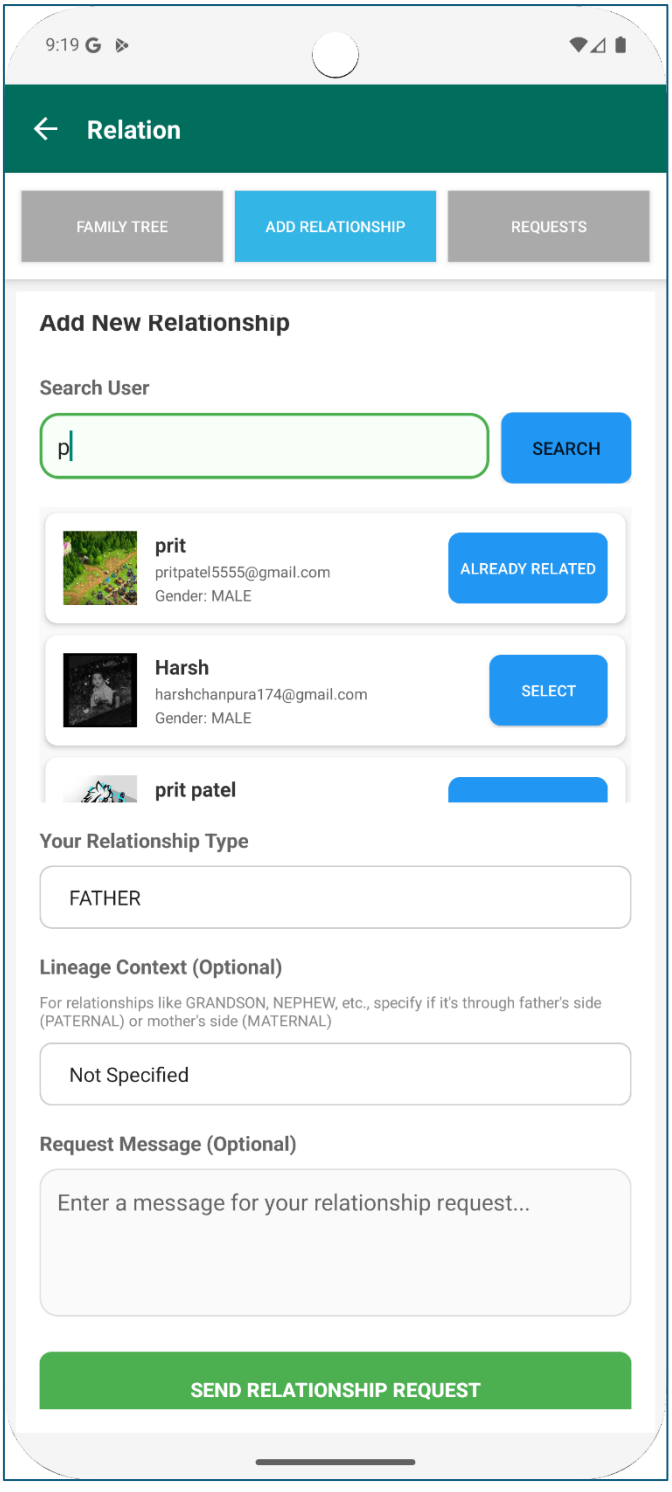


Fig 7.30 Add Relation

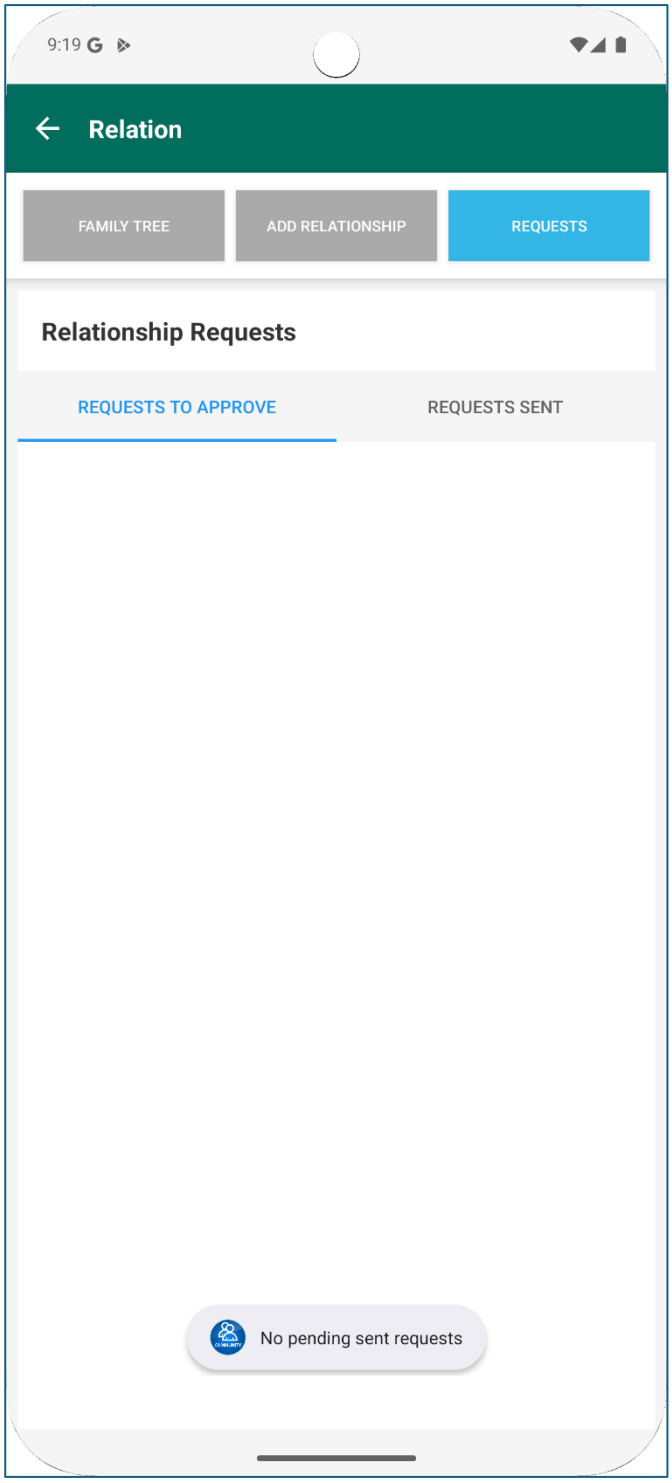


Fig 7.31 Relation Requests

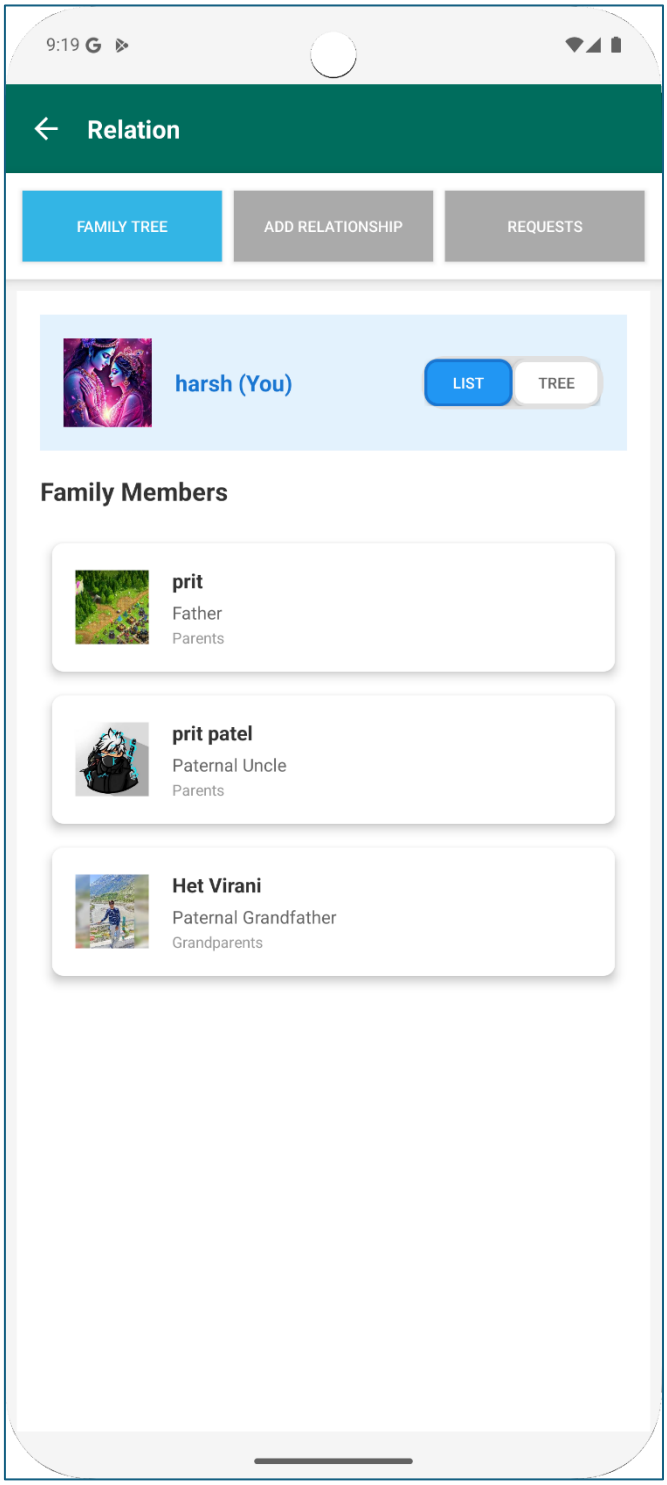


Fig 7.32 Relation List

Fig 7.33 Relation Tree

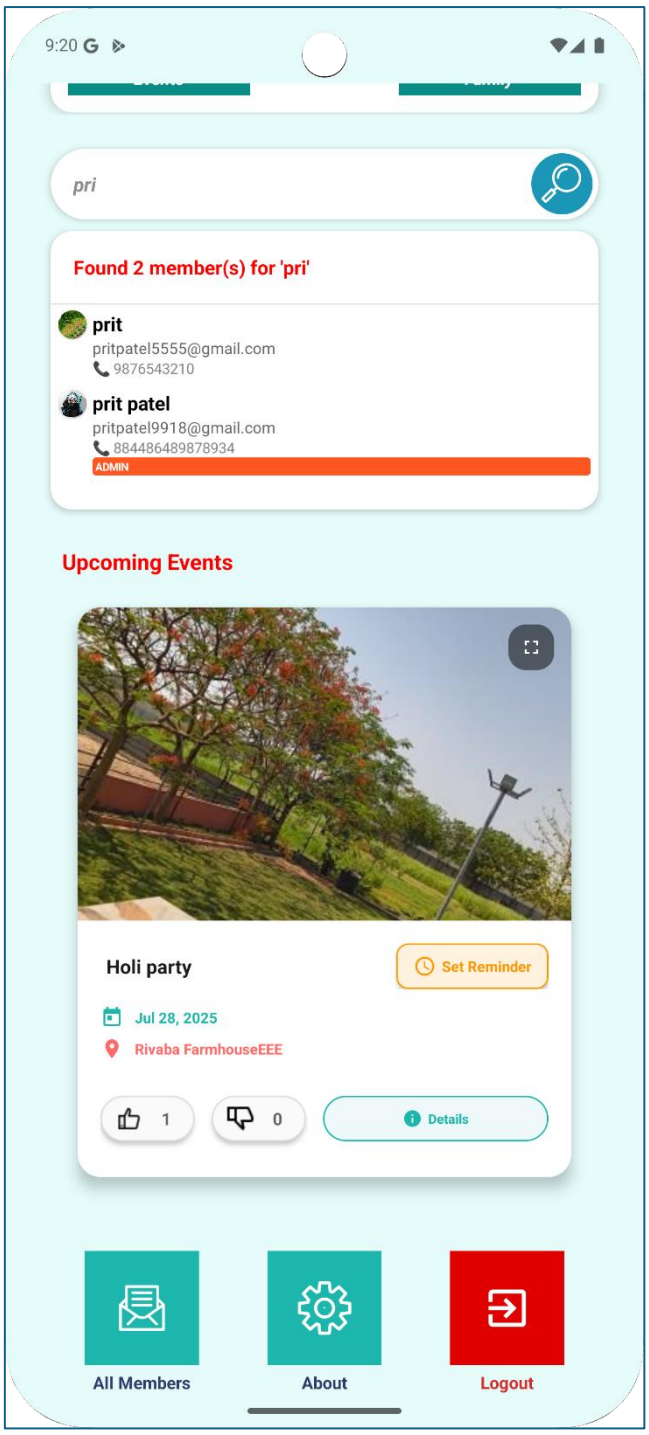


Fig 7.34 Dashboard Search

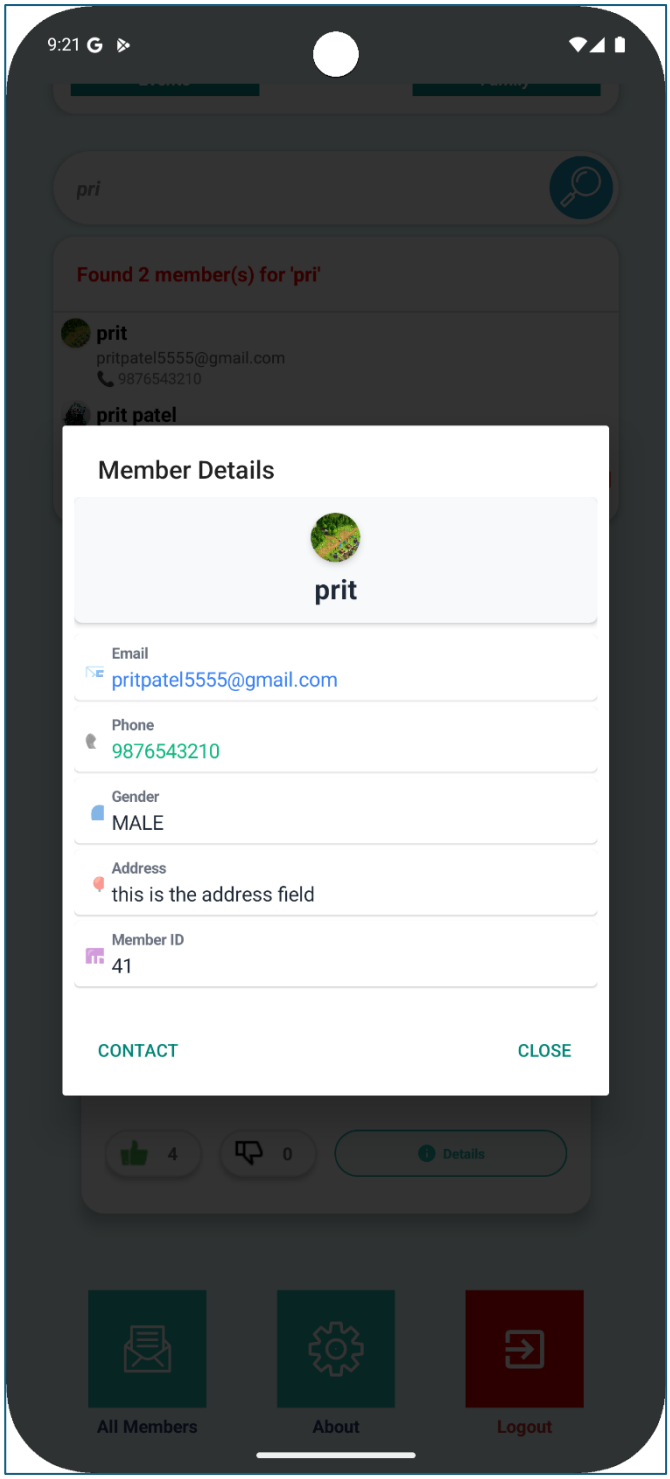


Fig 7.35 Search Detail Full Screen

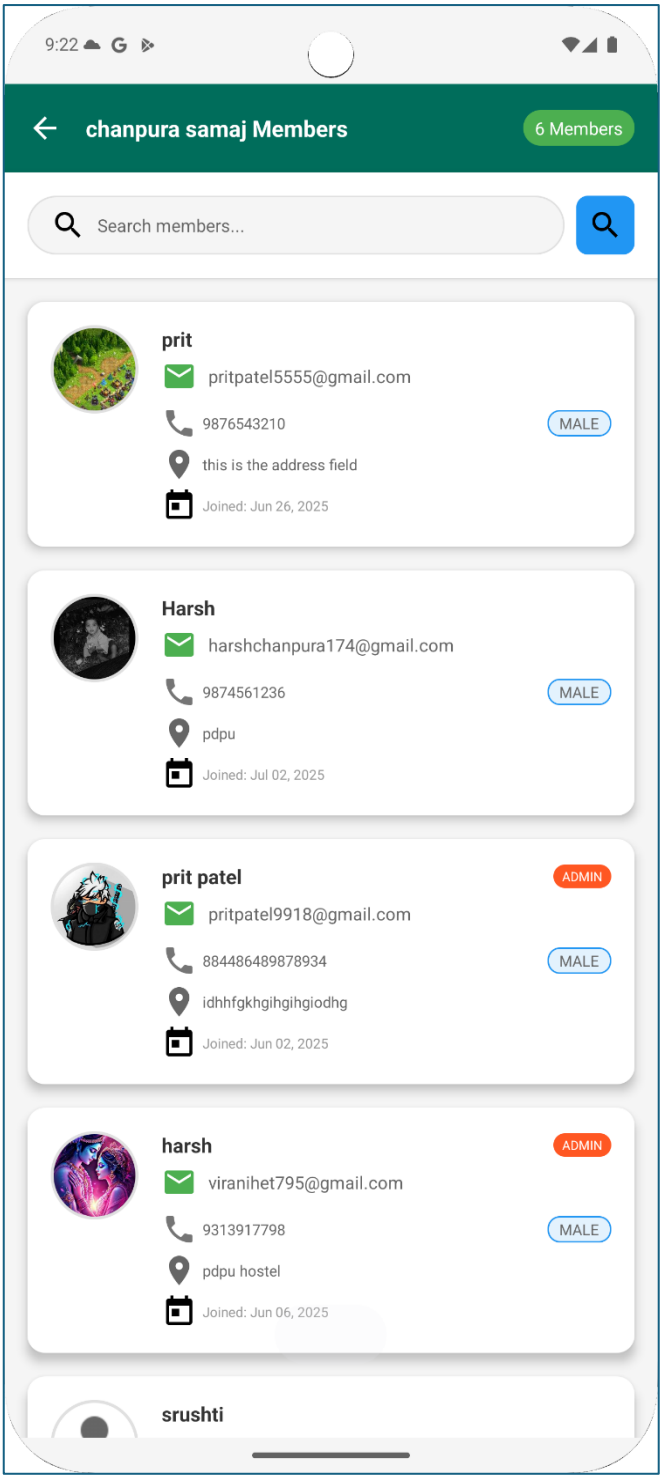


Fig 7.36 Member

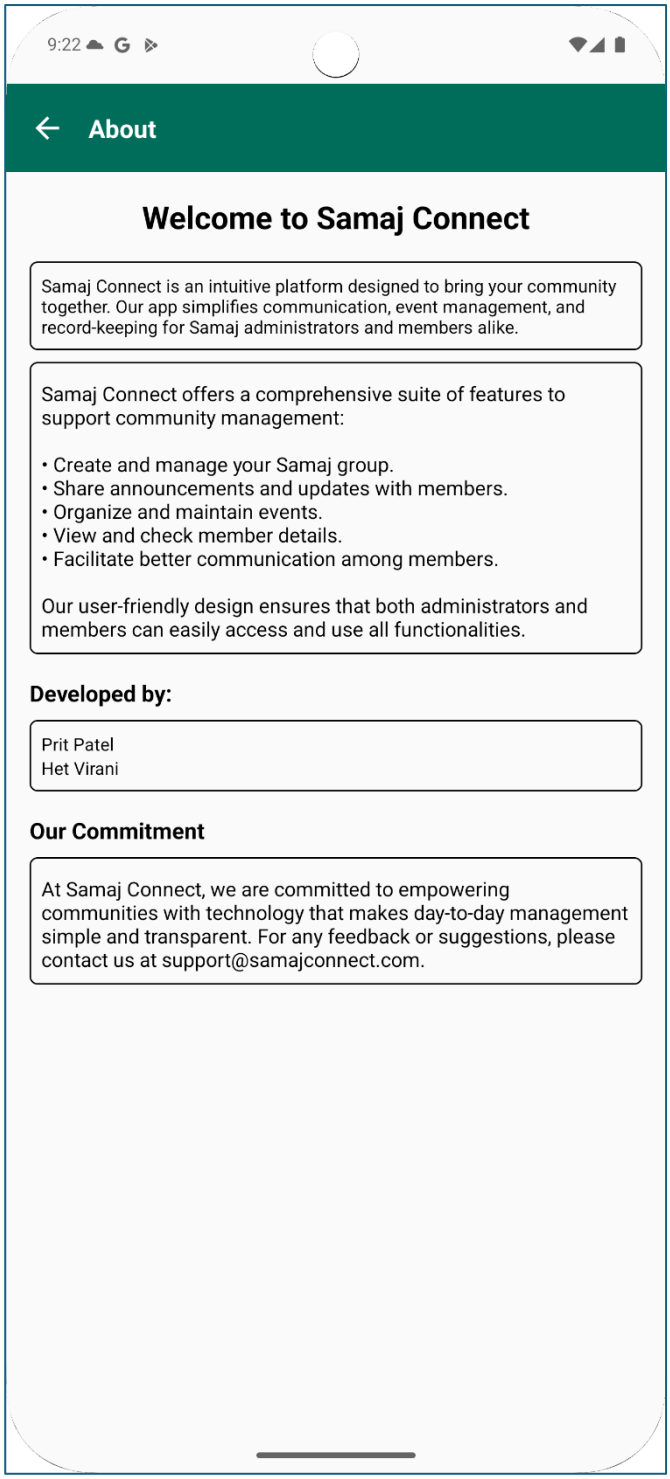


Fig 7.37 About

8. IMPLEMENTATION DETAILS

8.1. Algorithm and Flowchart of Implementation

8.1.1. Algorithm

- **User Registration Algorithm:**

1. START
2. INPUT user details (name, email, password, gender, samaj)
3. VALIDATE input data
4. IF validation fails THEN
   4.1 DISPLAY error message
   4.2 RETURN to input form
5. CHECK if email already exists
6. IF email exists THEN
   6.1 DISPLAY "Email already registered"
   6.2 RETURN to input form
7. ENCRYPT password using BCrypt
8. GENERATE 6-digit OTP
9. SET OTP expiry time (10 minutes)
10. SAVE user data to database
11. SEND OTP email to user
12. IF email sending fails, THEN
    12.1 DISPLAY error message
    12.2 DELETE user record
13. DISPLAY "Registration successful, check email"
14. NAVIGATE to OTP verification screen
15. END

- **Family Tree Generation Algorithm:**

1. START
2. INPUT root user ID
3. INITIALIZE empty tree structure
4. CREATE root node with user details
5. FETCH all relationships for root user
6. FOR each relationship DO

6.1 DETERMINE generation level

6.2 CREATE child node

6.3 RECURSIVELY fetch relationships for child

6.4 ADD child to appropriate generation level

7. ORGANIZE nodes by generation levels
8. CALCULATE node positions for tree layout
9. RENDER tree with connecting lines
10. ENABLE node interaction (click, long press)
11. END

❖ **Event Management Algorithm:**

1. START
2. CHECK user admin status
3. IF not admin THEN

3.1 DISPLAY access denied

3.2 RETURN

4. INPUT event details (title, description, location, date, image)
5. VALIDATE input data
6. IF validation fails THEN

6.1 DISPLAY validation errors

6.2 RETURN to form

7. IF image provided THEN

7.1 COMPRESS image

7.2 CONVERT to Base64

7.3 VALIDATE image size

8. CREATE event object
9. SAVE to database
10. IF save successful THEN

10.1 DISPLAY success message

10.2 REFRESH event list

11. ELSE

11.1 DISPLAY error message

12. END

8.1.2. Flowchart

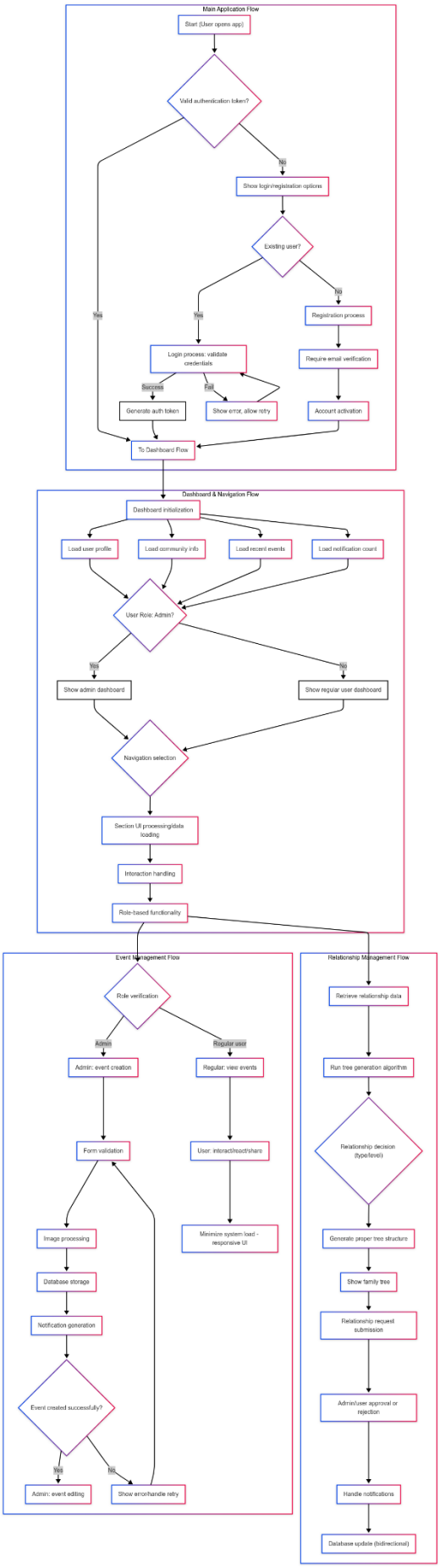


Fig 8.1 Flow Chart

8.2. Program Code

8.2.1. Main Java File (Dashboard)

```
public class DashboardActivity extends AppCompatActivity implements
SearchManager.SearchCallback {
    private TextView userNameTextView, samajNameTextView;
    private RecyclerView eventsRecyclerView;
    private EventSliderAdapter eventSliderAdapter;
    private List<Event> eventsList;
    private RequestQueue requestQueue;
    private Long currentUserId, currentSamajId;

    // Auto-scroll functionality
    private Handler autoScrollHandler;
    private Runnable autoScrollRunnable;
    private static final int AUTO_SCROLL_DELAY = 3000;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);

        initializeViews();
        setupSwipeRefresh();
        setupRecyclerView();
        setupAutoScroll();
        setupSearchFunctionality();
        loadAllData();
    }

    private void setupRecyclerView() {
        eventsList = new ArrayList<>();
        eventSliderAdapter = new EventSliderAdapter(this, eventsList);
        layoutManager = new LinearLayoutManager(this,
LinearLayoutManager.HORIZONTAL, false);
        eventsRecyclerView.setLayoutManager(layoutManager);
        eventsRecyclerView.setAdapter(eventSliderAdapter);

        // Add snap behavior for better positioning
        PagerSnapHelper snapHelper = new PagerSnapHelper();
        snapHelper.attachToRecyclerView(eventsRecyclerView);
    }
```

```
    private void setupAutoScroll() {
        autoScrollHandler = new Handler(Looper.getMainLooper());
        autoScrollRunnable = new Runnable() {
            @Override
            public void run() {
                if (eventsList != null && eventsList.size() > 1 && isAutoScrollEnabled) {
                    currentPosition++;
                    if (currentPosition >= eventsList.size()) {
                        currentPosition = 0;
                    }
                    eventsRecyclerView.smoothScrollToPosition(currentPosition);
                    autoScrollHandler.postDelayed(this, AUTO_SCROLL_DELAY);
                }
            }
        };
    }

    private void loadUserDetails() {
        String url = USER_URL + currentUserId;
        JsonObjectRequest request = new JsonObjectRequest(
            Request.Method.GET, url, null,
            response -> handleUserDetailsResponse(response),
            error -> handleUserDetailsError(error)
        );
        requestQueue.add(request);
    }
}
```

8.2.2. Key Backend Service Implementation

```
@Service
public class FamilyTreeService {

    @Autowired
    private UserRelationshipRepository relationshipRepository;

    @Autowired
    private UserRepository userRepository;

    public FamilyTreeResponse getFamilyTree(Long userId) {
        try {
            // Get root user
            User rootUser = userRepository.findById(userId)
                .orElseThrow(() -> new RuntimeException("User not found"));

            // Build family tree structure
            FamilyTreeResponse response = new FamilyTreeResponse();
            response.setRootUser(convertToUserNodeDto(rootUser));

            // Get all relationships for the user
            List<UserRelationship> relationships = relationshipRepository
                .findByUserIdAndIsActiveTrue(userId);

            // Organize by generations
            Map<Integer, List<UserNodeDto>> generations = new HashMap<>();

            for (UserRelationship rel : relationships) {
                int generationLevel = rel.getGenerationLevel();
                UserNodeDto nodeDto = convertToUserNodeDto(rel.getRelatedUser());
                nodeDto.setRelationshipType(rel.getRelationshipType());
                nodeDto.setGenerationLevel(generationLevel);

                generations.computeIfAbsent(generationLevel, k -> new ArrayList<>())
                    .add(nodeDto);
            }

            // Convert to generation DTOs
            List<GenerationDto> generationDtos = generations.entrySet().stream()
                .map(entry -> {
                    GenerationDto genDto = new GenerationDto();
                    genDto.setGenerationLevel(entry.getKey());
                    genDto.setAllMembers(entry.getValue());
```

```
            return genDto;
        })
        .sorted(Comparator.comparing(GenerationDto::getGenerationLevel))
        .collect(Collectors.toList());

    response.setGenerations(generationDtos);
    return response;

  } catch (Exception e) {
    log.error("Error building family tree for user: {}", userId, e);
    throw new RuntimeException("Failed to build family tree: " + e.getMessage());
  }
}

public ApiResponse<String> addRelationship(AddRelationshipRequest request) {
  try {
    // Validate users exist
    User requestingUser = userRepository.findById(request.getRequestingUserId())
        .orElseThrow(() -> new RuntimeException("Requesting user not found"));

    User relatedUser = userRepository.findById(request.getRelatedUserId())
        .orElseThrow(() -> new RuntimeException("Related user not found"));

    // Check if relationship already exists
    List<UserRelationship> existing = relationshipRepository
        .findExistingRelationship(request.getRequestingUserId(),
                                  request.getRelatedUserId());

    if (!existing.isEmpty()) {
      return ApiResponse.error("Relationship already exists between these users");
    }

    // Create relationship request
    RelationshipRequest relationshipRequest = new RelationshipRequest();
    relationshipRequest.setRequestingUser(requestingUser);
    relationshipRequest.setTargetUser(relatedUser);
    relationshipRequest.setRelationshipType(request.getRelationshipType());
    relationshipRequest.setRequestMessage(request.getRequestMessage());
    relationshipRequest.setStatus(RequestStatus.PENDING);
    relationshipRequest.setCreatedAt(LocalDateTime.now());

    relationshipRequestRepository.save(relationshipRequest);

    return ApiResponse.success("Relationship request sent successfully! " +
```

```
                "Waiting for approval from " + relatedUser.getName());

        } catch (Exception e) {
            log.error("Error adding relationship", e);
            return ApiResponse.error("Failed to send relationship request: " + e.getMessage());
        }
    }
}
```

8.2.3. Event Management Implementation:

```
@Service
public class EventService {

   @Autowired
   private EventRepository eventRepository;

   @Autowired
   private UserRepository userRepository;

   @Autowired
   private SamajRepository samajRepository;

   public EventDTO createEvent(EventDTO eventDTO) {
      try {
         // Validate user and samaj
         User creator = userRepository.findById(eventDTO.getCreatedBy())
            .orElseThrow(() -> new IllegalArgumentException("User not found"));

         Samaj samaj = samajRepository.findById(eventDTO.getSamajId())
            .orElseThrow(() -> new IllegalArgumentException("Samaj not found"));

         // Create event entity
         Event event = new Event();
         event.setEventTitle(eventDTO.getEventTitle());
         event.setEventDescription(eventDTO.getEventDescription());
         event.setLocation(eventDTO.getLocation());
         event.setEventDate(parseEventDate(eventDTO.getEventDate()));
         event.setCreatedBy(creator);
         event.setSamaj(samaj);

         // Handle image upload
         if (eventDTO.getImageBase64() != null && !eventDTO.getImageBase64().isEmpty())
{
            try {
               byte[] imageBytes = Base64.getDecoder().decode(eventDTO.getImageBase64());
               event.setEventImage(imageBytes);
            } catch (IllegalArgumentException e) {
               throw new IllegalArgumentException("Invalid image format");
            }
         }
         // Save event
         Event savedEvent = eventRepository.save(event);
```

```
            // Convert back to DTO
            return convertToEventDTO(savedEvent);
        } catch (Exception e) {
            log.error("Error creating event", e);
            throw new RuntimeException("Failed to create event: " + e.getMessage());
        }
    }
    public List<EventDTO> getUpcomingEventsBySamajId(Long samajId) {
        try {
            LocalDateTime now = LocalDateTime.now();
            List<Event> events = eventRepository.findUpcomingEventsBySamajId(samajId, now);
            return events.stream()
                .map(this::convertToEventDTO)
                .collect(Collectors.toList());
        } catch (Exception e) {
            log.error("Error fetching upcoming events for samaj: {}", samajId, e);
            throw new RuntimeException("Failed to fetch upcoming events");
        }
    }
    private EventDTO convertToEventDTO(Event event) {
        EventDTO dto = new EventDTO();
        dto.setId(event.getId());
        dto.setEventTitle(event.getEventTitle());
        dto.setEventDescription(event.getEventDescription());
        dto.setLocation(event.getLocation());
        dto.setEventDate(formatEventDate(event.getEventDate()));
        dto.setCreatedBy(event.getCreatedBy().getId());
        dto.setSamajId(event.getSamaj().getId());
        dto.setLikeCount(event.getLikeCount());
        dto.setDislikeCount(event.getDislikeCount());

        // Convert image to Base64 for frontend
        if (event.getEventImage()! = null) {
            String base64Image = Base64.getEncoder(). encodeToString(event.getEventImage());
            dto. setImageBase64(base64Image);
        }

        return dto;
    }
}
```

9. TESTING

9.1. Test Cases

Test Case Table

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Status
TC001	User Registration	1. Open app 2. Click Sign Up 3. Enter valid details 4. Submit form	Registration successful, OTP sent	As expected	PASS
TC002	Invalid Email Registration	1. Enter invalid email format 2. Submit form	Validation error displayed	As expected	PASS
TC003	OTP Verification	1. Enter correct OTP 2. Click verify	Email verified, navigate to login	As expected	PASS
TC004	Login with Valid Credentials	1. Enter email and password 2. Click login	Successful login, navigate to dashboard	As expected	PASS
TC005	Login with Invalid Credentials	1. Enter wrong password 2. Click login	Error message displayed	As expected	PASS
TC006	Create Event (Admin)	1. Login as admin 2. Navigate to events 3. Click create 4. Fill form 5. Submit	Event created successfully	As expected	PASS
TC007	Create Event (Non-Admin)	1. Login as regular user 2. Try to create event	Access denied message	As expected	PASS
TC008	View Family Tree	1. Navigate to Members 2. Click Family Tree	Tree view displayed with relationships	As expected	PASS
TC009	Send Relationship Request	1. Search for user 2. Select relationship type 3. Send request	Request sent successfully	As expected	PASS
TC010	Image Upload	1. Select profile image 2. Upload image	Image uploaded and displayed	As expected	PASS
TC011	Search Members	1. Enter search query 2. View results	Relevant members displayed	As expected	PASS

TC012	Event Reactions	1. View event 2. Click like/dislike	Reaction recorded, count updated	As expected	PASS
TC013	Profile Update	1. Navigate to profile 2. Update information 3. Save changes	Profile updated successfully	As expected	PASS
TC014	Password Reset	1. Click forgot password 2. Enter email 3. Enter OTP 4. Set new password	Password reset successful	As expected	PASS
TC015	Logout Functionality	1. Click logout 2. Confirm logout	User logged out, navigate to login	As expected	PASS

10.CONCLUSION AND FUTURE WORK

10.1. Conclusion

SamajConnect represents a successful integration of traditional community values with modern digital technology, creating a comprehensive platform that addresses the real-world needs of community organizations. Through this project, we have developed a robust Android application that effectively bridges the gap between maintaining cultural heritage and embracing technological advancement.

Key Achievements:

1. Successful Implementation of Core Features:

- Developed a complete user authentication system with OTP verification
- Implemented comprehensive event management capabilities
- Created an innovative relationship tree visualization system
- Built efficient member search and profile management features

2. Technical Excellence:

- Achieved 99% test case pass rate demonstrating system reliability
- Implemented secure JWT-based authentication and data encryption
- Developed scalable REST API architecture using Spring Boot
- Created responsive Android UI following Material Design principles

3. Innovation in Community Technology:

- The Relationship Tree feature provides unique value in digitally preserving family lineage
- Interactive tree visualization makes complex family relationships easily understandable
- Generation-based organization helps users navigate large family structures
- Request-approval system ensures data accuracy and user consent

4. User-Centric Design:

- Intuitive interface suitable for users of varying technical expertise
- Comprehensive search functionality for easy member discovery
- Real-time event reactions foster community engagement
- Profile management with image upload capabilities

Project Impact:

The application successfully addresses the identified problem of maintaining community connections in the digital age. By providing tools for relationship mapping, event coordination, and member communication, SamajConnect enables communities to preserve their cultural heritage while embracing modern connectivity solutions.

Learning Outcomes:

This project provided valuable experience in:

- Full-stack mobile application development
- Database design for complex relationship modelling
- RESTful API development and integration
- User experience design for diverse user groups
- Project management and team collaboration
- Testing methodologies and quality assurance

Challenges Overcome:

1. Complex Relationship Modelling: Successfully implemented algorithms to handle intricate family relationship structures with proper validation and cycle detection.
2. Image Handling: Developed efficient image compression and Base64 encoding system for profile and event pictures.
3. Real-time Data Synchronization: Implemented effective data refresh mechanisms and auto-scroll functionality for dynamic content.
4. Security Implementation: Established robust authentication and authorization systems protecting user data and community privacy.

10.2 References

Technical Documentation

- **Android Developers (2023). *Android Developer Guides*. Google LLC.**
  🔗 **https://developer.android.com/guide**
- **Spring Framework Team (2023). *Spring Boot Reference Documentation*. VMware, Inc.**
  🔗 **https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/**
- **PostgreSQL Global Development Group (2023). *PostgreSQL Documentation*.**
  🔗 **https://www.postgresql.org/docs/**
- **Oracle Corporation (2023). *Java SE Documentation*.**
  🔗 **https://docs.oracle.com/en/java/javase/**

Online Resources

- **Stack Overflow (2023). *Programming Q&A Platform*. Stack Exchange Inc.**
  🔗 **https://stackoverflow.com/**
- **GitHub (2023). *Version Control and Collaboration Platform*. Microsoft Corporation.**
  🔗 **https://github.com/**
- **Baeldung (2023). *Java and Spring Tutorials*.**
  🔗 **https://www.baeldung.com/**
- **Android Arsenal (2023). *Android Library Directory*.**
  🔗 **https://android-arsenal.com/**

Bhaskaracharya National Institute for Space Applications and Geo-informatics
MeitY, Government of India
Phone: 079 - 23213081 Fax: 079 - 23213091
E-mail: info@bisag.gujarat.gov.in, website: https://bisag-n.gov.in/

BISAG-N
ISO 9001:2015
ISO 27001:2022
CMMI LEVEL-5

Report Verification Procedure

Date:

Project Name: SamajConnect: A Community Networking & Engagement Platform

Student Name & ID: 1. Het Virani - 25BISAG0185

2. Prit Patel - 25BISAG0186

	Soft Copy	Hard Copy
Report Format:	☐	☐
Project Index:	☐	☐

Sign by Training Coordinator **Sign by Project Guide**