

**Batch: D-2**      **Roll No.: 16010123324**  
                                 **16010123325**  
                                 **16010123331**

**Experiment No.: 3**

**Grade: AA / AB / BB / BC / CC / CD / DD**

**Signature of the Staff In-charge with date**

**TITLE: Project Plan document for Mini Project**

**AIM:** To learn and understand the way of developing the software by classical methods of software engg. planning and monitoring of the project using tools and prepare a document for the same by using the concept of software engineering

---

**Expected OUTCOME of Experiment:**

Analyse the software requirements and Model the defined problem with the help of UML diagram

---

**Books/ Journals/ Websites referred:**

1. Roger Pressman, Software Engineering: A practitioners Approach, McGraw Hill, 2010 ,6<sup>th</sup> edition
  2. Ian Sommerville , Software Engineering , Addison Wesley, 2011, 9<sup>th</sup> edition
  - 3 [http://en.wikipedia.org/wiki/Software\\_requirements\\_specification](http://en.wikipedia.org/wiki/Software_requirements_specification)
-

**Software Project Management Plan**

**for**

***Clink***

***Shreya Menon, Shreyans Tatiya, Siddhant Raut***

***21-08-2024***

| <b>Version</b> | <b>Release Date</b> | <b>Responsible Party</b> | <b>Major Changes</b>                 |
|----------------|---------------------|--------------------------|--------------------------------------|
| 0.1            |                     |                          | Initial Document Release for Comment |

**Table of Contents**

|   |           |
|---|-----------|
| <b>1. Introduction</b>                                | <b>4</b>  |
| 1.1 Project Overview                                  | 4         |
| 1.2 Project Deliverables                              | 4         |
| 1.3 Evolution of the SPMP                             | 5         |
| 1.4 Reference Materials                               | 5         |
| 1.4.1 Clink Software Requirements Specification (SRS) | 6         |
| 1.4.2 Standards and Guidelines                        | 6         |
| 1.4.3 Software Engineering References                 | 6         |
| 1.4.4 Competitive Analysis and Market Research        | 7         |
| 1.4.5 Technical Documentation and APIs                | 7         |
| 1.4.6 Regulatory and Compliance References            | 7         |
| 1.5 Definitions and Acronyms                          | 8         |
| 1.5.1 Definitions                                     | 8         |
| 1.5.2 Acronyms  | 8         |
| <b>2. Project Organization</b>                        | <b>9</b>  |
| 2.1 Process Model                                     | 9         |
| 2.2 Organizational Structure                          | 12        |
| 2.3 Organizational Interfaces                         | 13        |
| 2.4 Project Responsibilities                          | 14        |
| <b>3. Managerial Process</b>                          | <b>15</b> |
| 3.1 Management Objectives and Priorities              | 15        |
| 3.2 Assumptions, Dependencies, and Constraints        | 15        |
| 3.3 Risk Management                                   | 16        |
| 3.4 Monitoring and Controlling Mechanisms             | 16        |
| 3.5 Staffing Approach                                 | 17        |
| <b>4. Technical Process</b>                           | <b>18</b> |
| 4.1 Methods, Tools, and Techniques                    | 18        |
| 4.2 Software Documentation                            | 19        |
| 4.2.1 Software Requirements Specification (SRS)       | 19        |
| 4.2.2 Software Design Description (SDD)               | 19        |
| 4.2.3 Software Test Plan                              | 19        |
| 4.3 User Documentation                                | 19        |
| 4.4 Project Support Functions                         | 20        |
| <b>5. Work Packages, Schedule, and Budget</b>         | <b>20</b> |
| 5.1 Work Packages                                     | 20        |
| 5.2 Dependencies                                      | 21        |
| 5.3 Resource Requirements                             | 21        |
| 5.4 Budget and Resource Allocation                    | 22        |
| 5.5 Schedule  | 22        |
| <b>6. Additional Components</b>                       | <b>23</b> |
| 6.1 Index   | 23        |
| 6.2 Appendices  | 24        |
| A. Current Top 10 Risk Chart                          | 25        |
| B. Current Project Work Breakdown Structure           | 25        |
| C. Current Detailed Project Schedule                  | 26        |

## 1. Introduction

Clink is a mobile-first social platform designed to revolutionize spontaneous, interest-based social planning for everyday users. The project aims to simplify real-world coordination by providing an intuitive app for creating, discovering, and joining casual hangouts, moving beyond rigid calendar systems and fragmented messaging tools. This document outlines the development approach, major features, and management plan for ensuring Clink's successful delivery and adoption.

### 1.1 Project Overview

Clink is a modern, mobile-first platform built to make spontaneous, interest-based social plans simple and accessible. The main objectives are to enable users to effortlessly create, discover, and join real-world hangouts or events without the hassle of traditional RSVPs or fragmented messaging apps. Key features include personalized feeds, integrated group chat, intuitive plan management, and role-based admin tools, all designed for high usability and spontaneity.

### 1.2 Project Deliverables

1. Clink Mobile Application (Android/iOS): Delivery by 15th November 2025, Quantity: 1, Delivery Location: Production cloud servers.
2. Host/Admin Dashboard (Web): Delivery by 20th November 2025, Quantity: 1, Delivery Location: Production cloud servers.
3. Real-Time Notification & Messaging Module: Delivery by 25th November 2025, Quantity: 1, Delivery Location: Production cloud servers.
4. Documentation & User Manuals: Delivery by 5th December 2025, Quantity: 10 hard copies and digital copies, Delivery Location: Client's office and web portal.
5. Training Sessions: Delivery: Ongoing throughout deployment, Quantity: 5 sessions, Delivery Location: Client's premises and online.
6. Technical Support: Delivery: Ongoing post-launch, Quantity: Continuous support for 1 year, Delivery Location: Online and onsite as needed.

### 1.3 Evolution of the SPMP

#### ***Completion and Dissemination Plan:***

1. **Development Phase:** The project begins with a detailed analysis, followed by iterative design, coding, and feature testing. Each milestone undergoes review and internal validation before integration.
2. **Deployment and Testing:** After successful internal tests, the system is deployed to production servers and tested in live environments for reliability and security.
3. **Documentation:** User manuals, technical docs, and training materials are created during development and updated as new features are added.
4. **Training Sessions:** Training for end-users and hosts is provided online and in-person, ensuring effective onboarding.
5. **Technical Support:** Dedicated support resources are available after deployment to resolve queries and maintain system functionality throughout the first year.

#### ***Change Control:***

1. **Scheduled Updates:** Planned updates, improvements, or feature additions will follow a structured change control process. This process includes requirements analysis, design, coding, testing, and deployment. Users will be notified in advance of scheduled updates.
2. **Unscheduled Updates:** Critical security patches and urgent bug fixes will be addressed immediately, following a rapid response change control procedure. Users will be informed of any unplanned downtime or changes.

### 1.4 Reference Materials

Provide a complete list of all documents and other sources of information referenced in the plan. Include for each the title, report number, date, author, and publishing organization.

#### 1.4.1 Clink Software Requirements Specification (SRS)

- Title: Software Requirements Specification for Clink
- Version: 1.0
- Date: July 24, 2025
- Authors: RTM & et al
- Publishing Organization: K. J. Somaiya College of Engineering (KJSCE)
- Document Reference: SE\_-EXP-1-SRS.pdf

#### 1.4.2 Standards and Guidelines

##### IEEE Standard for Software Project Management Plans

- Title: IEEE Std 1058-1998 - IEEE Standard for Software Project Management Plans
- Date: 1998
- Publishing Organization: Institute of Electrical and Electronics Engineers (IEEE)

##### Agile Software Development Standards

- Title: Agile Manifesto and Principles
- Date: 2001
- Authors: Kent Beck, Martin Fowler, et al.
- Publishing Organization: Agile Alliance
- URL: <https://agilemanifesto.org/>

#### 1.4.3 Software Engineering References

##### Software Engineering: A Practitioner's Approach

- Title: Software Engineering: A Practitioner's Approach
- Edition: 6th Edition
- Date: 2010
- Author: Roger S. Pressman
- Publishing Organization: McGraw-Hill

##### Software Engineering

- Title: Software Engineering
- Edition: 9th Edition
- Date: 2011
- Author: Ian Somerville
- Publishing Organization: Addison Wesley
-

#### 1.4.4 Competitive Analysis and Market Research

##### Partiful Platform

- Title: Partiful - Event Planning Platform
- URL: <https://partiful.com/>
- Access Date: September 2025
- Description: Reference platform for event planning and social coordination features

##### Luma Platform

- Title: Luma - Event Discovery and Management
- URL: <https://lu.ma/>
- Access Date: September 2025
- Description: Competitive analysis reference for event discovery and community building features

#### 1.4.5 Technical Documentation and APIs

##### React Native Documentation

- Title: React Native - Learn once, write anywhere
- Publishing Organization: Meta (Facebook)
- URL: <https://reactnative.dev/>
- Description: Mobile development framework documentation

##### Firebase Documentation

- Title: Firebase - Google's Mobile Platform
- Publishing Organization: Google LLC
- URL: <https://firebase.google.com/docs>
- Description: Backend-as-a-Service platform documentation

##### Node.js Documentation

- Title: Node.js - JavaScript Runtime
- Publishing Organization: Node.js Foundation
- URL: <https://nodejs.org/en/docs/>
- Description: Backend development runtime documentation

#### 1.4.6 Regulatory and Compliance References

##### GDPR Compliance Guidelines

- Title: General Data Protection Regulation (GDPR)
- Date: May 25, 2018
- Publishing Organization: European Union

- Description: Data protection and privacy regulation compliance requirements

#### Software Requirements Specification Guidelines

- Title: Software Requirements Specification
- Source: Wikipedia
- URL: [http://en.wikipedia.org/wiki/Software\\_requirements\\_specification](http://en.wikipedia.org/wiki/Software_requirements_specification)
- Access Date: September 2025
- Description: General reference for SRS documentation standards
- 

### 1.5 Definitions and Acronyms

#### 1.5.1 Definitions

**Agile Development:** An iterative software development methodology emphasizing customer collaboration, adaptive planning, and rapid delivery of working software increments.

**Clink:** Both the name of the social planning platform and an individual spontaneous social plan created by users within the system.

**Host:** A user who creates and manages a social plan, with the ability to set visibility, enable chat, and update plan details.

**Interest Matching:** Algorithm that ranks and filters Plans based on overlap between user-selected interests and Plan tags.

**Mobile-First Design:** A design approach that prioritizes the mobile user experience before scaling up to larger screens.

**Real-Time Updates:** Instantaneous synchronization of plan changes, participant lists, and chat messages across all connected devices.

**Scrum:** An Agile framework for managing product development with defined roles, events, and artifacts.

**Sprint:** A time-boxed iteration (typically 2 weeks) in Agile development where a potentially shippable product increment is created.

#### 1.5.2 Acronyms

**AES:** Advanced Encryption Standard; symmetric-key encryption algorithm used to secure data at rest and in transit.

**Department of Computer Engineering**

**API:** Application Programming Interface; a set of HTTP/HTTPS endpoints and protocols that enable communication between client applications and backend services.

**AWS:** Amazon Web Services; cloud computing platform providing infrastructure services.

**CDN:** Content Delivery Network; distributed server system for delivering web content efficiently.

**CI/CD:** Continuous Integration/Continuous Deployment; software development practice for automated testing and deployment.

**FCM:** Firebase Cloud Messaging; Google's cross-platform messaging solution for push notifications.

**HTTPS:** HyperText Transfer Protocol Secure; encrypted version of HTTP for secure data transmission.

**JWT:** JSON Web Token; compact, URL-safe token format used to represent authenticated user identity and permissions.

**OAuth:** Open standard for token-based, secure authorization, used for third-party logins.

**QA:** Quality Assurance; systematic process for ensuring software quality standards.

**REST:** Representational State Transfer; architectural style for designing web services.

**WebSocket:** Network protocol providing full-duplex communication channels over a single TCP connection for real-time updates.

## **2. Project Organization**

### **2.1 Process Model**

**Life Cycle Model:** The Clink Social Planning Platform Project will follow the Agile Software Development Life Cycle with continuous integration and deployment practices suitable for mobile-first applications.

**Roles:**

- **Product Owner:** Represents stakeholders, defines user stories, and sets feature priorities for spontaneous social planning needs

**Department of Computer Engineering**

**Department of Computer Engineering**

- Scrum Master: Ensures Agile practices are followed and facilitates sprint ceremonies
- Development Team: Mobile developers (React Native), backend developers (Node.js), UI/UX designers, QA engineers, DevOps engineers
- Stakeholders: End users (hosts and attendees), community moderators, college administrators, social group coordinators

**Activities:**

1. Project Initiation:
  - Define Clink's vision for spontaneous social coordination
  - Establish user personas (hosts, attendees, admins)
  - Create initial product backlog with core features (plan creation, feed discovery, chat integration)
  - Set up development and cloud infrastructure
2. Product Development:
  - Sprint planning (2-week cycles focused on feature increments)
  - Daily stand-up meetings for rapid issue resolution
  - Implement core functionalities: real-time feeds, location services, notification systems
  - Continuous testing on Android and iOS platforms
  - Regular demos to stakeholder groups for feedback
3. Product Release:
  - Beta testing with select user groups
  - User acceptance testing for mobile experience
  - Deployment to production cloud servers (AWS/Firebase)
  - App store submission and approval process
  - User onboarding and community guidelines training
4. Project Termination:
  - Evaluate user engagement metrics and platform adoption
  - Gather feedback for future feature enhancements
  - Archive technical documentation and user guides
  - Transition to maintenance and support phase

**Entry Criteria and Exit Criteria:**

- Project Initiation:
  - Entry: Project charter approved, technical stack selected, stakeholder interviews completed
  - Exit: Defined MVP scope, development team assembled, infrastructure provisioned

**Department of Computer Engineering**

- **Product Development:**
  - Entry: Approved product backlog with prioritized user stories
  - Exit: Functional mobile app with core features, real-time backend operational, security testing passed
- **Product Release:**
  - Entry: Completed development, successful beta testing, app store compliance verified
  - Exit: Live platform deployment, user documentation available, monitoring systems active
- **Project Termination:**
  - Entry: Successful product launch with active user base
  - Exit: Project retrospective completed, handover to operations team, knowledge transfer documented

This Agile approach enables rapid iteration and user feedback incorporation essential for social platform development, ensuring Clink meets the dynamic needs of spontaneous social coordination while maintaining high performance and security standards.

**Internal Management Structure:**

The internal management structure of the Clink Social Planning Platform Project comprises key roles and responsibilities within the development team, tailored for mobile-first application delivery and real-time social coordination features.

1. **Project Manager:**
  - Responsibilities: Overall project planning, coordination, and control of mobile app development cycles and cloud infrastructure deployment
2. **Product Owner:**
  - Responsibilities: Represents social platform stakeholders, sets feature priorities for spontaneous planning needs, manages the product backlog for user experience optimization
3. **Scrum Master:**
  - Responsibilities: Ensures Agile practices are followed, facilitates daily stand-up meetings, removes impediments for rapid mobile development cycles
4. **Development Team:**
  - Responsibilities: React Native developers, Node.js backend engineers, UI/UX designers specialized in social interfaces, mobile QA testers, DevOps engineers

**Department of Computer Engineering**

- Roles: Scrum Team Members responsible for real-time feature development and cross-platform compatibility

**Relationship with the Organization:**

The Clink Social Planning Platform Project operates as an innovative solution for spontaneous social coordination within educational and community environments. The project maintains regular communication with college administrators, student organizations, and potential user communities to ensure alignment with social engagement goals and campus policies.

**Lines of Authority:**

The Project Manager oversees the entire project lifecycle, while the Scrum Master and Product Owner collaborate closely with the Development Team to ensure efficient development and delivery of Clink's mobile-first social planning features.

**2.2 Organizational Structure**

- Project Manager: Responsible for planning, tracking sprint cycles, and ensuring deliverables for Clink's mobile-first social planning platform.
- Mobile Developers: Build cross-platform mobile UI using React Native with real-time feed interfaces and social interaction components.
- Backend Developers: Implement Node.js REST APIs, Socket.IO real-time chat, Firebase integration, and interest-based feed algorithms.
- Database Administrator: Maintain Firebase Firestore or MongoDB Atlas schemas, ensure data integrity for user profiles, plans, and chat messages.
- QA/Testers: Create test cases for social features, perform cross-platform mobile testing, and validate real-time synchronization functionality.
- DevOps Engineer: Handle deployment on cloud platforms (Firebase/AWS), manage push notification services (FCM/APNs), and maintain production monitoring systems.

### 2.3 Organizational Interfaces

| ORGANISATION                               | LIAISON        | CONTACT INFO  |
|--|----------------|---|
| Customer: College Administration           | [Liaison Name] | Phone: xxx-xxx-xxxx<br>Email: admin@kjsomaia.edu    |
| Mobile App Developer: Frontend Team        | [Liaison Name] | Phone: xxx-xxx-xxxx<br>Email: mobile_dev@clink.com  |
| Backend Developer: Server Infrastructure   | [Liaison Name] | Phone: xxx-xxx-xxxx<br>Email: backend_dev@clink.com |
| Database Administrator: Cloud Storage      | [Liaison Name] | Phone: xxx-xxx-xxxx<br>Email: db_admin@clink.com    |
| IT Support: Infrastructure Management      | [Liaison Name] | Phone: xxx-xxx-xxxx<br>Email: it_support@clink.com  |
| Quality Assurance Team: Testing & Security | [Liaison Name] | Phone: xxx-xxx-xxxx<br>Email: qa_team@clink.com     |

**Table F-1. Project Interfaces**

## 2.4 Project Responsibilities

| Function/Activity              | Description  | Responsible Individual |
|--------------------------------|--|------------------------|
| 1) Project Initiation          | Define Clink's scope, social planning objectives, and mobile-first goals               | Project Manager        |
| 2) Requirements Gathering      | Collect user stories for spontaneous social planning and document system requirements  | Business Analyst       |
| 3) System Design               | Plan mobile architecture, real-time infrastructure, and social coordination components | System Architect       |
| 4) Mobile Development          | Develop React Native app, implement feed algorithms, and integrate chat features       | Development Team       |
| 5) Backend Development         | Build Node.js APIs, configure Firebase/WebSocket for real-time updates                 | Backend Team           |
| 6) Testing & Quality Assurance | Perform cross-platform testing, security validation, and user experience testing       | QA Team                |
| 7) Deployment & DevOps         | Manage cloud infrastructure, app store submissions, and production monitoring          | DevOps Engineer        |
| 8) Maintenance and Support     | Provide ongoing system maintenance, user support, and feature updates                  | Support Team           |

**Table F-2. Project Responsibilities**

### 3. Managerial Process

This section of the SPMP specifies the management process for this project.

#### 3.1 Management Objectives and Priorities

The management philosophy for Clink emphasizes rapid iteration, user-centered design, and real-time feature delivery to support spontaneous social coordination. Goals include achieving high mobile app performance, seamless cross-platform functionality, and robust community moderation capabilities. Priority is given to delivering core social planning features while maintaining flexibility for user feedback integration

| Project Dimension     | Fixed | Constrained | Flexible |
|-----------------------|-------|-------------|----------|
| Cost                  |       | X           |          |
| Schedule              | X     |             |          |
| Scope (functionality) |       |             | X        |

**Table F-3: Flexibility Matrix**

#### 3.2 Assumptions, Dependencies, and Constraints

Assumptions:

- Target users have Android 8.0+ or iOS 13+ devices with GPS and internet connectivity
- Users are familiar with social media interface patterns and location-based services
- Host users will responsibly manage plan information and attendee coordination

Dependencies:

- Firebase or equivalent cloud platform for real-time database and authentication
- Google Maps/Mapbox API availability for geolocation services
- App Store and Google Play approval processes for mobile deployment
- Push notification services (APNs/FCM) for user engagement

Constraints:

- Must maintain 99.9% uptime for core social planning features
- Response times  $\leq 2$  seconds for major user actions
- GDPR/CCPA compliance for user data handling
- Cross-platform compatibility requirement for mobile-first approach

**Department of Computer Engineering**

Priority Statement: Schedule adherence takes highest priority to ensure timely launch, followed by core functionality delivery, with cost optimization as a secondary consideration.

### 3.3 Risk Management

Risk Identification Process:

- Weekly risk assessment meetings during sprint planning
- Stakeholder feedback monitoring for user adoption concerns
- Technical debt evaluation during code reviews

Key Risk Factors:

- Technological Risks: Real-time synchronization complexity, cross-platform compatibility issues
- User Adoption Risks: Competition from established social platforms, user behavior patterns
- Scalability Risks: Backend performance under high concurrent user loads
- Security Risks: User privacy concerns, data protection compliance

Mitigation Strategies:

- Prototype testing for real-time features before full implementation
- Phased rollout strategy starting with college communities
- Automated testing and monitoring systems for performance tracking
- Regular security audits and penetration testing

### 3.4 Monitoring and Controlling Mechanisms

Reporting Mechanisms:

- Sprint burndown charts for development progress tracking
- User engagement metrics dashboard for platform adoption
- Real-time system performance monitoring for uptime and response times

Review and Audit:

- Weekly sprint reviews with stakeholder demonstrations
- Monthly technical architecture reviews
- Quarterly user feedback analysis and feature prioritization

Table F-4: Communication and Reporting Plan

**Department of Computer Engineering**

| <b>Information Communicated</b> | <b>From</b>      | <b>To</b>                   | <b>Time Period</b> |
|---------------------------------|------------------|-----------------------------|--------------------|
| <b>Sprint Status Report</b>     | Development Team | Project Manager             | Weekly             |
| <b>Platform Metrics Report</b>  | Project Manager  | Product Owner, Stakeholders | Weekly             |
| <b>Technical Review</b>         | Development Team | System Architect            | Bi-weekly          |
| <b>User Feedback Analysis</b>   | Product Owner    | Development Team            | Monthly            |
| <b>Security Audit Report</b>    | Security Team    | Project Manager             | Monthly            |

**Table F-4: Communication and Reporting Plan**

### 3.5 Staffing Approach

**Required Skills:**

- Mobile development expertise (React Native, iOS/Android native)
- Backend development (Node.js, Express.js, real-time systems)
- Cloud infrastructure management (Firebase, AWS, MongoDB Atlas)
- UI/UX design specializing in social interfaces
- Security and data privacy compliance knowledge

**Recruitment Strategy:**

- Internal team augmentation with mobile specialists
- Contract UI/UX designers with social platform experience
- Cloud infrastructure consultants for scalability planning

**Training Requirements:**

- React Native development workshops for frontend team
- Firebase/WebSocket real-time programming training
- GDPR compliance and security best practices sessions

#### **4. Technical Process**

This section specifies the technical methods, tools, and techniques to be used on the project. It also includes identification of the work products and reviews to be held and the plans for the support group activities in user documentation, training, software quality assurance, and configuration management.

##### **4.1 Methods, Tools, and Techniques**

**Development Stack:**

- Mobile Framework: React Native for cross-platform compatibility
- Backend: Node.js with Express.js for API development
- Database: Firebase Firestore or MongoDB Atlas for scalable data storage
- Real-time Communication: Socket.IO or Firebase Realtime Database
- Authentication: JWT tokens with Firebase Auth integration
- Styling: Tailwind CSS for responsive design

**Development Methods:**

- Agile/Scrum methodology with 2-week sprints
- Test-driven development for critical social features
- Continuous integration and deployment practices
- Code review process for all commits

**Standards and Conventions:**

- RESTful API design standards for backend services
- Mobile UI/UX guidelines following platform-specific design systems
- Security protocols adhering to HTTPS/TLS standards

## 4.2 Software Documentation

Specify the work products to be built for this project and the types of peer reviews to be held for those products. It may be useful to include a table that is adapted from the organization's standard collection of work products and reviews. Identify any relevant style guide, naming conventions and documentation formats. In either this documentation plan or the project schedule provide a summary of the schedule and resource requirements for the documentation effort.

To ensure that the implementation of the software satisfies the requirements, the following documentation is required as a minimum:

### 4.2.1 Software Requirements Specification (SRS)

The SRS defines all essential requirements for Clink's spontaneous social planning functionality, including plan creation, feed discovery, real-time chat, and admin moderation capabilities. Each requirement is measurable and traceable to specific user scenarios and business needs.

### 4.2.2 Software Design Description (SDD)

The SDD outlines Clink's mobile-first architecture, including React Native frontend components, Node.js backend services, Firebase database schema, and real-time synchronization patterns. Interface specifications cover user interactions, API endpoints, and external service integrations.

### 4.2.3 Software Test Plan

The test plan encompasses unit testing for individual components, integration testing for real-time features, and user acceptance testing for social interaction scenarios. Testing includes cross-platform compatibility validation, performance benchmarking, and security vulnerability assessments.

## 4.3 User Documentation

Documentation Strategy:

- Interactive in-app onboarding tutorials for new users
- Context-sensitive help within key app sections
- Web-based FAQ and troubleshooting guides
- Video tutorials for host dashboard and plan management features
- Community guidelines and safety resources

**Delivery Methods:**

- Integrated help system within mobile application
- Online documentation portal accessible via web browser
- Email-based user guides for complex features
- In-app notification system for feature announcements

**4.4 Project Support Functions****Configuration Management:**

- Git-based version control with feature branch workflow
- Automated build and deployment pipelines for mobile apps
- Environment management for development, staging, and production

**Software Quality Assurance:**

- Automated testing for core user flows and real-time features
- Code quality metrics monitoring and technical debt management
- User experience testing with focus groups and beta user feedback
- Security testing including authentication and data protection validation

**Verification and Validation:**

- Requirements traceability matrix linking features to user needs
- User acceptance testing with representative social planning scenarios
- Performance validation against specified response time requirements

**5. Work Packages, Schedule, and Budget**

Specify the work packages, dependency relationships, resource requirements, allocation of budget and resources to work packages, and a project schedule. Much of the content may be in appendices that are living documents, updated as the work proceeds.

**5.1 Work Packages****WP-1: Project Foundation (Weeks 1-2)**

- Project setup, team onboarding, and infrastructure provisioning
- Requirements analysis and user story development
- Technical architecture design and technology stack finalization

**WP-2: Core Mobile Development (Weeks 3-8)**

**Department of Computer Engineering**

**Department of Computer Engineering**

- React Native app structure and navigation implementation
- User authentication and profile management features
- Plan creation and feed discovery functionality

**WP-3: Real-Time Features (Weeks 6-10)**

- WebSocket integration for live updates
- In-app chat system development
- Push notification implementation

**WP-4: Backend Services (Weeks 4-9)**

- Node.js API development and database integration
- Interest matching algorithms and feed generation
- Security implementation and data encryption

**WP-5: Admin Dashboard (Weeks 8-11)**

- Web-based moderation interface development
- Analytics and reporting functionality
- User management and content moderation tools

## 5.2 Dependencies

**Critical Path Dependencies:**

- WP-1 → WP-2 (Foundation must complete before mobile development)
- WP-2 → WP-3 (Core app required before real-time features)
- WP-4 → WP-3 (Backend APIs needed for real-time functionality)
- WP-2 → WP-5 (User system required before admin features)

**External Dependencies:**

- App store approval processes for mobile deployment
- Firebase service availability and API stability
- Third-party mapping service integration

## 5.3 Resource Requirements

**Personnel:** 3 team members (1 full-stack developer with mobile/backend expertise, 1 UI/UX designer with frontend capabilities, 1 project manager with QA responsibilities)

**Infrastructure:** Cloud hosting (AWS/Firebase), development environments, testing devices

**Software Tools:** Development IDEs, design software, project management tools

**Department of Computer Engineering**

Hardware: Mobile testing devices (Android/iOS), development workstations

#### 5.4 Budget and Resource Allocation

- Development Team: 70% of total budget
- Infrastructure and Tools: 20% of total budget
- Testing and QA: 5% of total budget
- Project Management: 5% of total budget

#### 5.5 Schedule

Phase 1 - Foundation (Weeks 1-3): Project setup, requirements analysis, architecture design

Phase 2 - Core Development (Weeks 4-9): Mobile app and backend implementation

Phase 3 - Integration (Weeks 8-11): Real-time features and admin dashboard

Phase 4 - Testing & Deployment (Weeks 10-12): Quality assurance, app store submission, production deployment

Key Milestones:

- Week 3: Technical architecture approval
- Week 6: MVP mobile app demo
- Week 9: Backend API integration complete
- Week 11: Beta testing launch
- Week 12: Production deployment and user onboarding

## 6. Additional Components

Certain additional components may be required and may be appended as additional sections or subsections to the SPMP. Additional items of importance for the Clink social planning platform project include mobile app security plans, real-time infrastructure management plans, cross-platform deployment plans, user privacy compliance plans, social feature moderation plans, and platform scaling strategies.

### 6.1 Index

| Term                     | Reference Section | Description   |
|--------------------------|-------------------|---|
| <b>Admin Dashboard</b>   | 1.2, 5.1          | Web-based moderation interface with analytics & community management tools <sup>[1]</sup>                     |
| <b>API</b>               | 1.5, 4.1          | RESTful and WebSocket APIs enabling real-time communication between mobile clients and backend <sup>[1]</sup> |
| <b>Clink</b>             | 1.2, 2.1, 5.1     | Individual spontaneous social plan created and shared by users within the platform <sup>[1]</sup>             |
| <b>Feed Algorithm</b>    | 1.2, 4.1          | Interest-based recommendation system for personalized plan discovery <sup>[1]</sup>                           |
| <b>Firebase</b>          | 4.1               | Backend-as-a-Service platform for real-time database, authentication, and push notifications <sup>[1]</sup>   |
| <b>Host Module</b>       | 1.2, 5.1          | User interface for creating, managing, and updating social plans <sup>[1]</sup>                               |
| <b>Interest Matching</b> | 1.2, 4.1          | Algorithm that ranks and filters plans based on user preferences and location <sup>[1]</sup>                  |
| <b>JWT</b>               | 4.1               | JSON Web Tokens for secure user authentication and session management <sup>[1]</sup>                          |
| <b>Mobile-First</b>      | 1.5, 2.1          | Design approach prioritizing mobile user experience using React Native <sup>[1]</sup>                         |
| <b>Node.js</b>           | 4.1               | JavaScript runtime for backend API development and server-side logic <sup>[1]</sup>                           |

**Department of Computer Engineering**

|                               |               |   |
|-------------------------------|---------------|---|
| <b>Project Manager</b>        | 2.2, 2.4, 3.4 | Responsible for planning, sprint coordination, and mobile app delivery                          |
| <b>Push Notifications</b>     | 4.1           | Real-time alerts via FCM/APNs for plan updates and social interactions <sup>[1]</sup>           |
| <b>QA (Quality Assurance)</b> | 1.5, 4.4      | Cross-platform mobile testing and real-time feature validation                                  |
| <b>React Native</b>           | 4.1           | Cross-platform mobile development framework for iOS and Android apps <sup>[1]</sup>             |
| <b>Real-Time Chat</b>         | 1.2, 5.1      | In-app group messaging using <a href="#">Socket.IO</a> for plan coordination <sup>[1]</sup>     |
| <b>SDD</b>                    | 4.2.2         | Software Design Description for mobile architecture and backend services                        |
| <a href="#">Socket.IO</a>     | 4.1           | Library enabling real-time bidirectional communication for chat and live updates <sup>[1]</sup> |
| <b>SRS</b>                    | 1.5, 4.2.1    | Software Requirements Specification for Clink social planning platform <sup>[1]</sup>           |
| <b>WebSocket</b>              | 4.1           | Protocol for real-time communication between mobile clients and backend <sup>[1]</sup>          |
| <b>WBS</b>                    | Appendix B    | Work Breakdown Structure for mobile development and deployment phases                           |

## 6.2 Appendices

| Risk ID   | Risk Description                                 | Probability | Impact | Mitigation Strategy  |
|-----------|--|-------------|--------|--|
| <b>R1</b> | Real-time synchronization failure across devices | Medium      | High   | Implement robust WebSocket fallback mechanisms; extensive testing <sup>[1]</sup> |
| <b>R2</b> | Cross-platform mobile compatibility issues       | High        | Medium | Regular testing on multiple iOS/Android versions; automated CI/CD <sup>[1]</sup> |

**Department of Computer Engineering**

|            |  |        |        |  |
|------------|--|--------|--------|--|
| <b>R3</b>  | User adoption challenges in competitive social space | Medium | High   | Beta testing with college communities; user feedback integration <sup>[1]</sup>  |
| <b>R4</b>  | Firebase/cloud service outages                       | Low    | High   | Multi-cloud backup strategy; local caching implementation <sup>[1]</sup>         |
| <b>R5</b>  | Privacy compliance violations (GDPR/CCPA)            | Low    | High   | Legal review; privacy-by-design architecture; user consent flows <sup>[1]</sup>  |
| <b>R6</b>  | App store approval delays                            | Medium | Medium | Follow platform guidelines; prepare alternative distribution methods             |
| <b>R7</b>  | Mobile development team resource constraints         | Medium | Medium | Cross-training; agile sprint planning; external contractor backup                |
| <b>R8</b>  | Chat moderation and safety concerns                  | Medium | High   | Automated content filtering; robust reporting system; admin tools <sup>[1]</sup> |
| <b>R9</b>  | Location service accuracy issues                     | Low    | Medium | Multiple geolocation providers; user manual location input option <sup>[1]</sup> |
| <b>R10</b> | Scope creep with social features                     | Medium | High   | Change control process; stakeholder approval for new features                    |

**A. Current Top 10 Risk Chart**
**B. Current Project Work Breakdown Structure**
**1. Requirements Analysis & Planning (WP1)**

- 1.1 Mobile user research and persona development
- 1.2 Social planning feature requirements approval
- 1.3 Technical architecture planning for real-time systems

**2. System Design (WP2)**

- 2.1 Firebase database schema for users, plans, and chats
- 2.2 Mobile UI/UX wireframes and user flow design
- 2.3 API endpoint specification and real-time communication protocols

**3. Mobile Development Phases**

- Increment 1 (WP3): User authentication, profile setup, basic plan creation
- Increment 2 (WP4): Feed discovery, interest matching, plan joining
- Increment 3 (WP5): Real-time chat, push notifications, live updates

**Department of Computer Engineering**

**Department of Computer Engineering**

- Increment 4 (WP6): Admin dashboard, analytics, moderation tools

**4. Testing & Integration (WP7)**

- 4.1 Cross-platform mobile testing (iOS/Android)
- 4.2 Real-time feature integration testing
- 4.3 User acceptance testing with target social groups

**5. Deployment & Documentation (WP8)**

- 5.1 App store submission and cloud infrastructure deployment
- 5.2 User onboarding materials and help documentation
- 5.3 Technical documentation and handover materials

**C. Current Detailed Project Schedule**

| Week              | Activity                                      |
|-------------------|---|
| <b>Week 1</b>     | Requirements Analysis & Mobile Planning (WP1) |
| <b>Week 1-2</b>   | System Design & Architecture (WP2)            |
| <b>Week 2-4</b>   | Core Mobile Features Development (WP3)        |
| <b>Week 4-6</b>   | Social Discovery & Matching (WP4)             |
| <b>Week 6-8</b>   | Real-time Communication Features (WP5)        |
| <b>Week 8-10</b>  | Admin Tools & Analytics (WP6)                 |
| <b>Week 10-11</b> | Cross-platform Testing & Integration (WP7)    |
| <b>Week 11-12</b> | App Store Deployment & Documentation (WP8)    |

**Conclusion:**

This software engineering experiment successfully demonstrated the comprehensive development of a Software Project Management Plan (SPMP) for the Clink social planning platform, providing practical experience in translating user requirements into structured technical specifications and project management documentation. The exercise enhanced our understanding of modern software development methodologies, risk management strategies, and the critical importance of proper planning in delivering successful mobile-first applications that meet both technical and user experience requirement

### **Post Lab Descriptive Questions**

1. State various Scheduling principles and explain them in detail.

#### **1. Fairness**

- Meaning: Every process should get a fair share of CPU time.
- Explanation: No process should be starved (ignored indefinitely). Scheduling should prevent monopolization of the CPU by one process.
- Example: Round Robin scheduling ensures fairness by giving each process a fixed time slice in rotation.

#### **2. Efficiency**

- Meaning: Keep the CPU as busy as possible.
- Explanation: CPU utilization should be maximized so that no resources are wasted. Idle CPU = wasted computation power.
- Example: Multiprogramming ensures that while one process waits for I/O, another can use the CPU.

#### **3. Response Time**

- Meaning: The time taken from the submission of a request until the first response is produced.
- Explanation: Particularly important in interactive systems, where users expect quick feedback.
- Example: Shortest Job Next (SJN) minimizes response time by executing short processes first.

#### **4. Turnaround Time**

- Meaning: The total time taken from the submission of a job until its completion.

- Explanation: Users prefer their jobs to complete quickly. Scheduling should aim to minimize turnaround time.
- Example: Batch systems try to schedule jobs to minimize average turnaround.

#### 5. Throughput

- Meaning: The number of processes completed per unit time.
- Explanation: High throughput means the system can finish more processes in less time, which improves overall system productivity.
- Example: Running multiple smaller jobs first may increase throughput compared to one very large job.

#### 6. Predictability

- Meaning: Processes should finish in about the same amount of time each run.
- Explanation: Variations in response and completion times should be minimal to ensure reliability for users.
- Example: Priority scheduling without aging may cause unpredictability, so aging is added to stabilize it.

#### 7. Deadline Meeting (Timeliness)

- Meaning: Some tasks (especially in real-time systems) must finish before a deadline.
- Explanation: In hard real-time systems, missing a deadline can be catastrophic (e.g., medical equipment, autopilot).

#### 8. Balance

- Meaning: The system should balance different resource needs (CPU, I/O, memory).
- Explanation: Some processes are CPU-bound, others are I/O-bound; scheduling should ensure both types progress efficiently.