Project Overview

- **Course**: CMPUT 301 Winter 2025
- **Project Duration**: January 27 - March 31, 2025 (9 weeks)
- **Team Size**: 6 members
- **Project Description**: Android application allowing users to post, track, and share their moods with features including mood events, filtering, following other users, and geolocation.

Sprint Structure

- **Sprint Duration**: 2 weeks per sprint
- **Total Sprints**: 4 sprints + 1 week buffer/final submission
- **Weekly Stand-ups**: 45-60 minutes, Wednesday
- **Sprint Planning**: Beginning of each sprint
- **Sprint Review/Retrospective**: End of each sprint

Sprint Breakdown

Sprint 0 (January 27 - February 2): Project Setup & Planning

Goal: Set up the project environment and establish the foundations

Tasks:

- Set up Android project with appropriate architecture
- Configure GitHub repository with branching strategy
- Create UI/UX wireframes and mockups
- Design database schema (Firebase)
- Define project coding standards and documentation practices
- Set up CI/CD pipeline for automated testing
- Create initial product backlog and prioritize user stories

Deliverables:

- Project repository with initial commit
- Project architecture document
- UI mockups
- Database schema diagram
- Team roles assignment document

Sprint 1 (February 3 - February 16): Core Functionality

Goal: Implement core mood tracking functionality

User Stories:

- US 01.01.01: Add mood events with date/time, emotional state, optional trigger, and social situation
- US 01.02.01: Implement the 8 emotional states (anger, confusion, etc.)
- US 01.03.01: Create consistent emoticons and colors for emotional states
- US 01.04.01: View mood event details
- US 01.05.01: Edit mood event details
- US 01.06.01: Delete mood events
- US 01.07.01: Support for private/public mood events
- US 03.01.01: User profiles with unique usernames
- US 04.01.01: View mood history in reverse chronological order

Technical Tasks:

- Implement local database functionality
- Create data models for users and mood events
- Develop UI components for mood input and display
- Create adapter for mood history list view
- Implement user authentication system

Deliverables:

- Functional user registration and login
- Ability to create, view, edit, and delete mood events
- Basic mood history display
- Basic Unit tests for core functionality

Sprint 2 (February 17 - March 2): Enhanced Mood Events & Filtering

Goal: Complete mood event details and implement filtering functionality

User Stories:

- US 02.01.01: Add textual explanation (reason) for mood events
- US 02.02.01: Add photo capability to mood events
- US 02.03.01: Implement photo size restrictions
- US 02.04.01: Add social situation selection
- US 04.02.01: Filter mood history by recent week
- US 04.03.01: Filter mood history by emotional state
- US 04.04.01: Filter mood history by reason text
- US 07.01.01: Support offline mood event creation/editing

Technical Tasks:

- Implement image capture and storage functionality
- Create image compression algorithm to meet size requirements
- Develop filter UI and logic for mood history
- Implement local caching for offline functionality
- Create synchronization mechanism for offline changes

Deliverables:

- Complete mood event creation with all features (text, photos, social situation)
- Functional filtering system for mood history
- Offline capability with synchronization
- Integration tests for filtering and offline features

Sprint 3 (March 3 - March 16): Social Features & Following

Goal: Implement social features to enable sharing and following

User Stories:

- US 03.02.01: Search for other users
- US 03.03.01: View other users' profiles
- US 05.01.01: Request to follow another participant's mood events
- US 05.02.01: Grant following permission to other users
- US 05.02.02: View users who have requested to follow
- US 05.03.01: View recent mood events of followed users
- US 05.04.01: Filter following list by recent week
- US 05.05.01: Filter following list by emotional state
- US 05.06.01: Filter following list by reason text

Technical Tasks:

- Implement user search functionality
- Create following request system
- Develop permission management system
- Build UI for viewing followed users' mood events
- Implement filtering for followed mood events
- Add notifications for follow requests

Deliverables:

- User search and profile viewing functionality
- Follow request and permission system
- Feed of followed users' mood events with filtering
- Unit and integration tests for social features

Sprint 4 (March 17 - March 30): Geolocation & Wow Factor

Goal: Implement location features and add the wow factor

User Stories:

- US 06.01.01: Attach location to mood events
- US 06.02.01: View map of mood events from history
- US 06.03.01: View map of mood events from following list
- US 06.04.01: View map of recent mood events within 5km
- US 05.07.01: Comment on mood events
- US 05.07.02: View comments on mood events
- Wow factor: QR code generation to be scanned by another user to follow them

Technical Tasks:

- Integrate Google Maps API
- Implement location services and permissions
- Create map visualization for mood events
- Develop comment system for mood events
- Implement notifications for comments
- Final UI polishing and performance optimization

Deliverables:

- Functional geolocation features
- Interactive maps showing mood events
- Comment system for mood events
- Final application with all features implemented
- Comprehensive test suite

Final Week (March 31): Project Finalization

Goal: Finalize the project for submission

Tasks:

- Bug fixing and final testing
- Documentation completion
- Code review and refactoring
- Performance optimization
- Final presentation preparation
- Project submission

Deliverables:

- Completed Android application meeting all requirements
- Final documentation
- Presentation materials
- Final submission package

Testing Strategy

- **Unit Testing**: Test individual components throughout development
- **Integration Testing**: Test interaction between components after each sprint
- **UI Testing**: Manual and automated testing of user interface
- **User Acceptance Testing**: Have team members use the app as end users
- **Performance Testing**: Ensure the app performs well on different devices

Risk Management

1. **Technical Challenges**:
    - Geolocation and map integration complexity
    - Offline synchronization issues
    - Image compression requirements
2. **Time Constraints**:
    - Buffer time in final week for unexpected delays
    - Prioritize core functionality in early sprints
3. **Team Challenges**:
    - Regular communication to address misunderstandings
    - Pair programming for complex features
    - Knowledge sharing sessions for specialized topics