

4.1 SDP Plan Introduction

The **Momento** app aims to simplify the process of visual journaling by offering users a unified, aesthetically pleasing platform to capture daily moments, moods, and memories without the need for written entries. Development activities will include gathering requirements, designing the system, implementing key features, integrating relevant APIs, conducting testing, and deploying the application. Key milestones include completing the requirements analysis by **Week 5**, implementing the core modules by **Week 12**, and conducting final testing and deployment by **Week 15**.

4.1.1 Project Deliverables

- **Requirements Specification Document** (Due Week 5): Detailed description of the application's features and functionalities.
- **Module Implementations** (Due Week 10):
 - **Daily Photo Capture with Vintage Disposable Filter**: Upload photo when the user desires.
 - **Add Photos from Camera Roll with iOS Suggestions**: Users can upload photos they want to document in their journal.
 - **Daily Mood Check-ins**: Users can check in with a photo representing their mood in that current moment.
 - **Monthly Recap of Photos**: Every month users will have a recap compilation for them, allowing them to see how their month has been. These recaps will be saved in an archive section of the app.
- **Testing Report** (Due Week 12): Results of unit and integration testing.
- **Final Deployment** (Due Week 15): Fully functional application deployed and demonstrated.

4.2 Project Resources

4.2.1 Hardware Resources

- **Development Machines**:
 - **Processor**: Apple M1 or Intel i5 (or higher)
 - **RAM**: 16 GB minimum
 - **Storage**: 256 GB SSD minimum
- **Testing Devices**: Various iOS devices including iPhones and iPads, covering multiple screen sizes.

4.2.2 Software Resources

- **Design Tools:**
 - **Figma:** Used for designing the UI/UX of the app, creating wireframes, prototypes, and high-fidelity mockups. Collaboration features in Figma will enable team feedback and revisions.
- **Development Tools:**
 - **Xcode:** The official IDE for iOS development.
 - **Swift:** Programming language for native iOS development.
 - **Firebase:** Backend for cloud storage and real-time syncing of user photos and mood data.
- **APIs:**
 - **Core Image Framework:** For implementing photo filters.
 - **iOS Journaling Suggestions API:** To recommend relevant photos based on location and events from the camera roll.

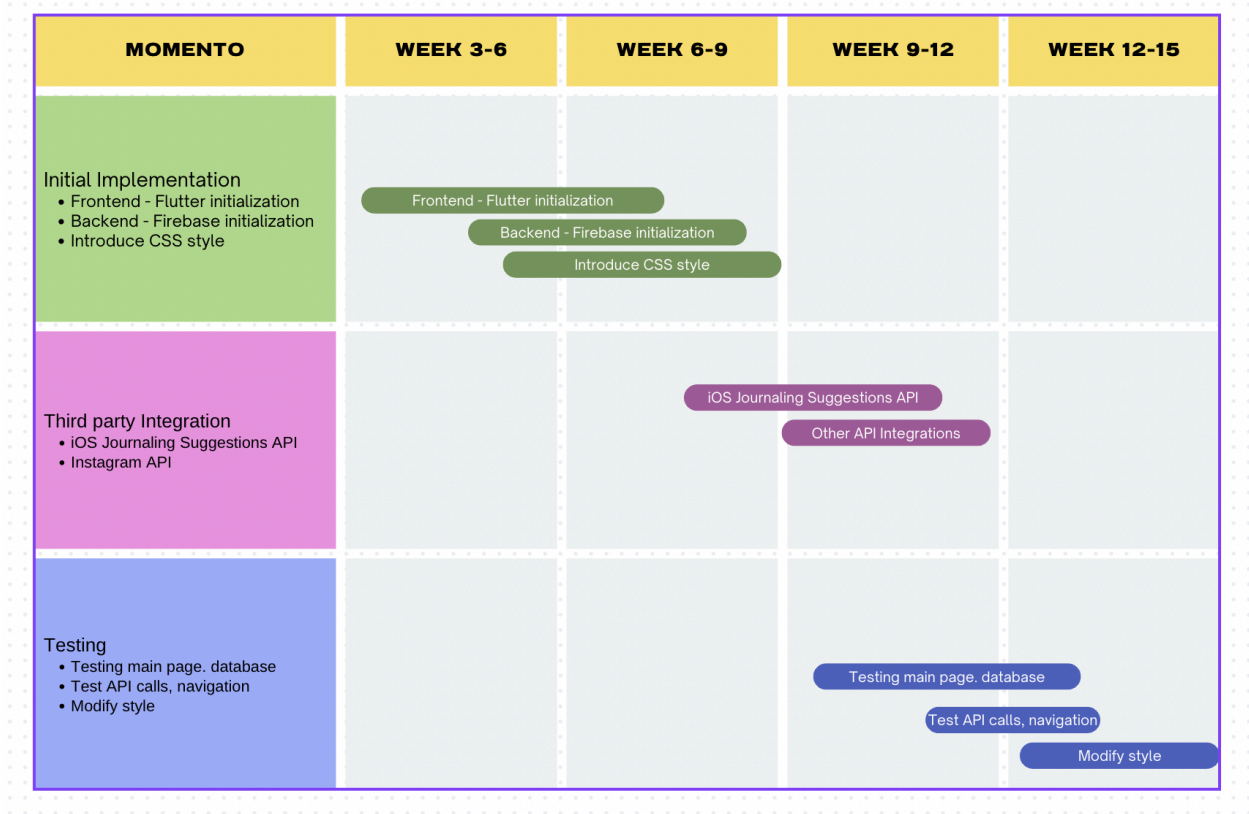
4.3 Project Organization/Human Resources

The project is structured around four major functions corresponding to the application modules. Each function will be managed and developed sequentially to ensure quality and integration.

Functions and Responsibilities

- **Daily photo capture:** Allow users to take a photo directly within the app, timestamp the photo, and store it in the user's journal for the day. Integrate the device's camera API to capture photos, automatically save with metadata (date, time, location), and store it in a database or local storage.
- **Photo upload:** Enable users to upload photos from their device's gallery to their journal. Use the device's file picker API to select and upload photos, compress them if necessary, and store them with relevant metadata like upload date.
- **Mood check-ins:** Create a user interface that prompts users to take or upload a photo reflecting their mood. The photo is then stored with a timestamp and any optional mood tags (e.g., emojis or mood categories) for later reflection.
- **Monthly recap:** Generate a monthly summary by displaying key photos from the month, accompanied by mood trends, and allow users to review or share their memories. Query the stored photos, retrieve key photos based on user activity, and present the summary in a grid or carousel format. You could also visualize mood trends with charts or color-coded displays.
-

4.4 Schedule



4.4.2 Task/Resource Table

Task	Resources Required	Person
Requirements Analysis	Development Machine, IDE	All members
Design Document	Development Machine, IDE	All members
Frontend Implementation in Flutter	Development Machine, Flutter, IDE	All members
Backend Implementation: Firebase for cloud storage and photo management.	Development Machine, Firebase, IDE	All members

iOS Journaling Suggestions API Integration	Development Machine, iOS Journaling Suggestions API Integration	All members
Instagram API	Development Machine, Instagram API, IDE	All members
Integration and Testing	All Modules, Testing Devices	All members
Final Deployment	Server, Development Machine	Dev Raiyani