

Software Design Document

Introduction

This document provides an updated detailed design description for the *Momento* application, incorporating new features and enhancements since the initial design. It serves as a comprehensive guide for developers and potential stakeholders, detailing the system architecture, component interactions, and design decisions.

2. System Overview

Momento is a visual journaling app for iOS devices that enables users to document daily moments through photos, track their moods, and reflect on their experiences. The app offers features such as daily photo capture with vintage filters, mood check-ins, daily inspirational quotes, monthly recaps, and secure data storage via Firebase.

3. Design Considerations

3.1 Assumptions and Dependencies

- Users possess iOS devices with camera functionality.
- An active internet connection is available for data synchronization.
- The app complies with Apple's App Store guidelines and data privacy regulations.

3.2 Constraints

- Compatibility with iOS 14 or later versions.
- Adherence to Firebase's usage policies for data storage.

4. Architectural Design

4.1 System Architecture

The architecture follows a Model-View-ViewModel (MVVM) pattern:

- Model: Manages data and business logic.
- View: Handles the user interface and user interactions.
- ViewModel: Acts as an intermediary between the Model and the View, facilitating communication and data binding.

4.2 Component Diagram

A high-level component diagram illustrates the interactions between core components:

- User Interface (UI) Layer: Comprises SwiftUI views for various screens (Home, Capture, Mood Check-in, Recap, Archive).
- ViewModel Layer: Contains view models corresponding to each UI component, managing state and business logic.
- Model Layer: Includes data models representing photos, moods, and quotes.
- Services: Encompasses Firebase service for data storage and retrieval, and a Quote service for fetching daily quotes.

5. Detailed Design

5.1 User Interface Design

- Home Screen: Displays navigation options to Capture Photo, Mood Check-in, View Recap, and Archive.
- Capture Photo Screen: Provides camera functionality with options to apply vintage filters.
- Mood Check-in Screen: Presents a selection of mood icons for user input.
- Recap Screen: Showcases a visual summary of the user's monthly activities.
- Archive Screen: Allows browsing of past entries organized by date.

5.2 Data Management

- Data Models:
 - PhotoEntry: Represents a photo with attributes like image data, date, and applied filter.
 - MoodEntry: Captures the user's mood with attributes such as mood type and associated date.
 - Quote: Contains the daily inspirational quote text and author.
- Data Storage:
 - Utilizes Firebase Firestore for storing user-generated content and mood entries.
 - Implements Firebase Authentication for secure user access.

5.3 Daily Quote Feature Integration

- Quote Service:
 - Fetches daily quotes from a preloaded database or an external API.
 - Caches the quote locally to ensure availability during offline usage.
- ViewModel Updates:
 - Incorporates logic to retrieve and display the daily quote on the Home and Capture Photo screens.

5.4 Monthly Recap Generation

- Recap Algorithm:
 - Aggregates photos and mood entries for the month.

- Generates a visual summary, highlighting trends and notable moments.
- UI Presentation:
 - Displays the recap in a scrollable, visually appealing format.
 - Allows users to share their monthly recap via social media or save it to their device.

6. Interface Design

6.1 External Interfaces

- Camera Interface: Integrates with the device's native camera for photo capture.
- Share Sheet: Utilizes iOS's sharing functionality to enable users to share content.

6.2 Internal Interfaces

- ViewModel to Model Communication: Employs Combine framework for reactive data binding between ViewModels and Models.
- Service Integration: ViewModels interact with services (e.g., FirebaseService, QuoteService) to perform data operations.

7. Design Rationale

- MVVM Pattern: Chosen for its separation of concerns, facilitating easier testing and maintenance.
- SwiftUI: Selected for its declarative syntax and seamless integration with iOS, enhancing development efficiency.
- Firebase: Adopted for its real-time capabilities and scalability, ensuring secure and efficient data management.

8. Security Considerations

- Data Encryption: Ensures all user data is encrypted during transmission and storage.
- Authentication: Implements Firebase Authentication to manage user access securely.
- Privacy Compliance: Adheres to relevant data protection regulations, providing users with control over their data.

9. Conclusion

This updated design description outlines the enhancements and new features integrated into *Momento*. The design decisions prioritize user experience, performance, and security, ensuring that *Momento* remains a reliable and engaging platform for visual journaling.