# Developer Documentation - DearFuture App

## Introduction

DearFuture is a time capsule application that allows users to store digital messages, photos, or videos in capsules that remain locked until a specified future date. This document outlines the challenges faced during development and how the final implementation differs from the initial vision.

## Challenges and How the App Evolved

### 1. Learning and Code Organization

Problem: This was one of my first major projects, and I realized that I am still not very organized inside the code. Writing clean, structured, and efficient code was harder than expected.

Solution: Throughout development, I improved significantly in structuring my code, following better practices, and understanding how to break down functionalities into manageable components. While the current code is still not highly optimized or beautifully structured, I am satisfied with what I accomplished and recognize the progress I made.

### 2. Choosing the Right Platform

Initial Plan: The project was first developed as a WPF (Windows Presentation Foundation) application. The expectation was that WPF would offer a flexible UI and easy local data handling.

Problem: WPF turned out to be cumbersome, requiring a lot of UI customization and lacking native support for mobile platforms. This made it difficult to scale and integrate modern features like geolocation.

Solution: After realizing the limitations of WPF, I switched to .NET MAUI, which provided a streamlined and cross-platform development framework. This change simplified the implementation and allowed for better UI consistency across devices.

### 3. Complexity of Implementation

Initial Plan: The original vision included several key features:

- Create Time Capsules: Save text, images, or videos that unlock at a set date.

- Reminder Functionality: Notifications when capsules unlock.

- Personalization Options: Custom themes, titles, and descriptions.

- Sharing Options: QR code sharing and capsule collaboration.

- Secure Storage: Password protection for privacy.

- Interactive Timeline: A calendar-based unlock view.

What was actually implemented:

- Completed: Capsule creation and customization.

- Added: Location-based unlocking, archive system, and trash management.

- Not implemented: QR code sharing, public database, password protection, and reminders.

Problem: Some features, like QR sharing and a public database, required significant infrastructure and security considerations that were beyond the project's scope.

Solution: The focus shifted to improving core functionality with local SQLite storage, location-based unlocking, and an archive/trash system to keep the app streamlined and practical.

### 4. QR Code Sharing and Public Database

Initial Plan: Users would scan a QR code to receive shared capsules, requiring a public database for capsule storage.

Problem: Implementing a public database raised issues related to user authentication, data security, and complexity in managing shared access.

Solution: Sticking with local storage ensured privacy, removed authentication concerns, and simplified development. While QR sharing remains a possible future feature, it requires backend infrastructure that was outside the project's scope.

## Conclusion

The final app turned out differently from the initial vision, but in a way that made it more practical and achievable. Transitioning from WPF to .NET MAUI greatly improved development efficiency. The realization of how complex features like capsule management and QR sharing would be led to prioritizing a local database solution for a more streamlined experience.

I chose .NET MAUI in order to continue this project in the future, add more features, and easily release it across multiple platforms. Its cross-platform capabilities allow for better scalability and potential expansion without being limited to a single operating system.

Future improvements could include implementing cloud storage, account-based capsule sharing, notifications, and a more appealing UI design.

PS: This youtube video saved me a lot of time: <https://www.youtube.com/watch?v=DuNLR_NJv8U&t=6937s>

Dotnet issues: https://github.com/dotnet/android/issues/9133