

# Socially Anxious Hub

## Project Plan



Student: Virag Szabo (4727444)

Date: Summer of 2025

Subject: Threading in C#

School: NHL Stenden



# Table of contents

## Contents

Table of contents .....	2
1 Project Overview.....	3
2 Scope.....	3
3 Objectives .....	3
4 People.....	4
4.1 Stakeholders.....	4
4.2 Members .....	4
5 Timeline .....	5
6 Task and Activities .....	6
6.1 Design .....	6
6.2 Development .....	6
6.3 Testing.....	6
6.4 Presentation .....	6
7 Risk Management & Mitigation .....	7
8 Definition of Success .....	8
8.1 Key Performance Indicators (KPIs).....	8
8.2 Success Criteria.....	8



# 1 Project Overview

The **Socially Anxious Hub** project aims to develop a comprehensive C# .NET MAUI application that helps users to manage their social life through music and personal memories. It will allow users to create and manage playlists using **Spotify API** and a **digital memory board** to cherish personal moments. The app will be designed with a focus on user privacy and a calm, reassuring user experience.

## 2 Scope

The project scope includes designing the user interface and experience (UI/UX), developing backend services for API integration and local data storage, building cross-platform frontend components using **.NET MAUI**, implementing a comprehensive testing strategy, and delivering a final presentation summarizing project objectives and outcomes. The project will focus on the **Android and iOS platforms**.

## 3 Objectives

- **Implement Core Features:** Successfully build and validate the Spotify integration and the memory board.
- **Ensure User-Centric Design:** Develop a clean, intuitive and easily accessible user interface that is easy to navigate and use.
- **Prioritize Security:** Securely handle user authentication and store sensitive data locally using appropriate encryption methods.
- **Establish a Robust Foundation:** Prepare for future features by creating a well-structured and scalable application architecture (e.g., MVVM).
- **Perform Comprehensive Testing:** Conduct thorough unit, integration, performance, user acceptance testing to ensure the app is reliable and bug-free.
- **Present Findings:** Deliver a presentation to stakeholders detailing the project's journey, key features, and future potential.



## 4 People

### 4.1 Stakeholders

Name	Email	Phone	Place
<b>Rob Loves</b>	rob.loves@nhlstenden.com	+31610480182	Emmen

As a stakeholder, he's responsible for providing input, feedback, and support for the project. He's also involved in reviewing progress, ensuring the project aligns with organizational goals, and addressing any concerns that arise during the project lifecycle.

### 4.2 Members

Name	Email	Phone	Place
<b>Virag Szabo</b>	virag.szabo@student.nhlstenden.com	+31641685452	Zwolle

As the project manager, I'm responsible for overall planning, coordination, and execution. As the UI/UX designer, I'm responsible for creating mock-ups. As the backend developer, I'm responsible for implementing authentication, Spotify API integration, and local data persistence. As the frontend developer, I'm responsible for implementing the cross-platform UI using .NET MAUI. As the tester, I'm responsible for developing and executing a comprehensive test plan. As the presenter, I'm responsible for preparing and delivering a presentation at the end.



## 5 Timeline

Phase	Date	Description
<b>System Architecture</b>	August 1, 2025	Define the overall application architecture. Write Start Document.
<b>User Interface Mock-ups</b>	August 5, 2025	Create and finalize mockups for the app's interfaces.
<b>Development</b>	August 10 – 30, 2025	Build the project, add core features (Spotify integration, memory board), and implement threading for performance.
<b>Testing</b>	August 20 – 30, 2025	Test features and fix bugs.
<b>Submit</b>	August 31, 2025	Hand in the project with all necessary documents and folders.
<b>Presentation</b>	August 31, 2025	Submit the video presentation.



## 6 Task and Activities

### 6.1 Design

**UI/UX Design:** Create UI mock-ups using a tool like Canva, Figma or Adobe XD for the playlist and memory board pages, ensuring consistency and a calming user experience.

**Data Models:** Define data models for Song and MemoryItem, including properties for storage and display.

### 6.2 Development

**Environment Setup:** Configure the .NET MAUI development environment.

**Backend Development:** Implement the SpotifyService for authentication (PKCE flow) and API calls. Implement a DatabaseService using SQLite for storing memory board data and playlists.

**Frontend Development:** Develop the PlaylistPage and MemoryBoardPage using XAML and C# code-behind, leveraging the MVVM pattern with ObservableCollection to ensure a responsive UI.

### 6.3 Testing

**Unit Testing:** Develop and run unit tests for core methods in SpotifyService and DatabaseService.

**Integration Testing:** Verify the end-to-end functionality, from MainPage authentication to adding a song to a playlist and saving a memory.

**User Acceptance Testing (UAT):** Conduct testing with a small group of users to gather feedback on usability and identify any bugs.

**Performance Testing:** Verify if the UI is responsive, does not take excessive amounts of time to use the application, and does not take a lot of time to make a API call, also, the battery consumption is professionally enough.

### 6.4 Presentation

**Preparation:** Prepare a presentation summarizing project objectives, features, achievements, and challenges.

**Delivery:** Practice and deliver the presentation to stakeholders.



## 7 Risk Management & Mitigation

#	Risks	Mitigation Strategies
<b>Technical</b>	Implementing a secure authentication flow (PKCE) and handling different types of API responses.	Implement the Spotify PKCE authentication flow early in development to validate the approach.
<b>Development</b>	Limited time to complete all features.	Use an iterative approach to prioritize core features and deliver a working product on time.
<b>Vulnerabilities</b>	Risk of exposing API keys or user data if security measures are not implemented correctly.	Adhere to best practices like using SecureStorage and avoiding hardcoding secrets.
<b>Scope Management</b>	Temptation to add extra features that go beyond the project scope.	Stick to the core features and push extra ideas to a "future work" section of the project.
<b>Third-party Dependencies</b>	Reliance on Spotify's API, which could change or have outages.	While not in scope, mention the possibility of adding YouTube Music API as a fallback.
<b>User-Centric Design</b>	Continuously gather user feedback to improve application usability and engagement.	Continuously gather user feedback to improve application usability and engagement.



## 8 Definition of Success

### 8.1 Key Performance Indicators (KPIs)

- **User Engagement:** Number of users who successfully create a playlist or memory board item.
- **Data Integrity:** Accuracy of data fetched from Spotify and correctly displayed in the app.
- **Performance Metrics:** Fast load times for API requests and responsive UI (e.g., no freezing during searches).
- **User Satisfaction:** Positive feedback from test users on the app's functionality and ease of use.

### 8.2 Success Criteria

- **Cote Milestones:** Successful completion of all project milestones within the defined timelines.
- **Feature Completeness:** The app successfully authenticates with Spotify, fetches and displays songs, and allows users to create, view, and save memories.
- **Secure Implementation:** The app's security measures are validated through testing.
- **Stakeholder Satisfaction:** The project meets the expectations of the lecturer and presents a clear demonstration of the concepts covered in the course.