

| Department of Computer Science |
| --- |



**CPSC 490 / 491 CAPSTONE PROJECT PROPOSAL**

**AUTHOR(S)**

| **Student Name** |  | **CSUF Email** |  | **Student ID** |
| --- | --- | --- | --- | --- |
| **Diego Vela** |  | dvela2@csu.fullerton.edu |  | 885098145 |

| Tentative Title: | **HarmoniQ** |
| --- | --- |
| CPSC 490: | **Fall 2024** |
|  | Semester, Year |
| CPSC 491 (Expected): | **Spring 2025** |
|  | Semester, Year |

| Proposal (CPSC 490) Faculty Advisor: |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Name |  | Email |  | Date Approved |
| Project (CPSC 491) Faculty Advisor: |  |  |  |  |  |
|  | Name |  | Email |  | Date Approved |

**Abstract**

HarmoniQ is a mobile application designed to teach music theory through an engaging, gamified learning experience. Drawing inspiration from platforms like Duolingo, the app incorporates progressive challenges, levels, quizzes, and interactive exercises to help users of varying skill levels master key music theory concepts. By offering lessons on topics such as music notation, rhythm, scales, and chord progressions, the app provides a fun and effective way to build and enhance musical literacy. Gamification elements, including achievements and streaks, foster user motivation and retention. The project aims to make music theory accessible and enjoyable, allowing users to develop both their theoretical knowledge and practical musical skills. HarmoniQ will be developed using Flutter, targeting both Android and iOS devices.

**Table of Contents**

[**1. Introduction 1**](#_hgazj5ms3asq)

[1.1. Background and Context 1](#_ni80wybk8c0i)

[1.2. Objectives and Significance 1](#_a2cp1l4ac7dj)

[1.3. Motivation for Proposal Choice 1](#_w2pouhp9rf7x)

[1.4. Capstone Summary 1](#_q611xvs0f3mq)

[**2. Product Scope and Charter 2**](#_s0grox78vx6m)

[2.1. Description of the Product 2](#_ffpuu9vj6cg5)

[2.2. Features and Functional Requirements 3](#_6knjngo36ryc)

[2.2.1. Core Product 3](#_sl9bazp0cgjl)

[2.2.2. Stretch Goals / Phase 2 3](#_3khsuckkbvcf)

[2.3. Non-functional Requirements 3](#_z0blhrh27ppc)

[2.4. Target and Operational Environment 3](#_a46owkdor4um)

[2.5. Project Deliverables 4](#_bf40i028belo)

[2.6. Project Assumptions and Limitations 4](#_sl4g9rio6mg5)

[2.7. Ethical Considerations 4](#_jtjei2k2xelo)

[**3. Related Works and Concerns 4**](#_1lm9ec6p0bz3)

[3.1. Literature Review 4](#_2gfb0s2qybju)

[3.2. Related Products and Systems 5](#_615il6jpduv9)

[3.3. Security Concerns and Approaches 5](#_7mh4o824ojc3)

[3.4. Relevant Standards 5](#_tu6uasaatexy)

[**4. Stakeholders, Users, and Benefits 5**](#_j31j44bpilt6)

[4.1. Stakeholder Identification 5](#_hath6ow9uj00)

[4.2. User Identification 6](#_osagtobow3sm)

[4.3. User Needs, Benefits, and Roles 6](#_b3fzjsbrpp7t)

[4.4. Internal User Roles 6](#_ppe9g1sktejn)

[**5. Technical Architecture & High-Level Design 6**](#_8da2qxe0935a)

[5.1. System Architecture 6](#_mam3jqhb5rpm)

[5.2. High-Level Design 7](#_anfswz3x77ny)

[5.3. Data Stores 7](#_1a39u7nqab0j)

[5.4. Frameworks, APIs, Components, Services, and Tools Used 7](#_duhzlquu7d43)

[**6. Project Management 8**](#_3qvsm9qs8cd0)

[6.1. Team 8](#_t8au8qy933m1)

[6.2. Methodology 8](#_4vgd8z6636fg)

[6.3. Project Timeline 8](#_jj3y5ma1mf3d)

[6.4. Subprojects 8](#_7kv5a110wet6)

[6.5. Risks and Feasibility](#_z6xxu4a8mvnm) 9

[6.7. Release Plan 9](#_cxmwkds2qfgz)

[**7. Evaluation and Assessment 9**](#_1s6na6g8730r)

[7.1. Success Criteria 9](#_h9lq5lvpofo6)

[7.2. Testing Plan 9](#_hmmlb8t8f17q)

[7.3. User Evaluation 9](#_pxk7zr1nymt3)

[7.4. Measurement of Benefits 10](#_e73wle5rq66v)

[**8. Glossary 10**](#_g8y86vsbbhz)

[**9. Appendix 10**](#_kp3xvj9zmxoh)

[9.1. Appendix 1: Product Summary 10](#_15ssmr7ghxym)

[9.2. Appendix 2: ECS Expo Abstract 10](#_q9x1r8cyvnx8)

[9.2.1. Project Background 10](#_ca4kkjv069n4)

[9.2.2. Problem Statement 10](#_ldelguppm9mi)

[9.2.3. Goals and Objectives 11](#_g2mkudacvokm)

[9.2.4. Design Requirement/Specifications 11](#_d9bfjmju3fdk)

[9.2.5. Methodology 11](#_43hhpluks6gd)

[9.2.6. Results or Outcomes 11](#_5f8cv1lln4a1)

[9.3. Draft ECS Poster 11](#_ufkea1p89d3)

[**10. References 11**](#_go3t29r5aeew)

**List of Figures and Tables**

*2.2.1 User Stories…………………………………………………………………………… 4*

*2.2.2 Stretch Goal User Stories…………………………………………………………… 5*

*2.3 Non-Functional Requirements User Stories………………………………………… 6*

*5.2.1: UX Flow Diagram……………………………………………………………………10*

*5.2.2: Logo Prototype………………………………………………………………………10*

*5.2.3: Design Prototype: Main Page……………………………………………………..10*

## Introduction

HarmoniQ is a Duolingo-type mobile application that focuses on taking users of varying levels of music theory knowledge and making them proficient at reading music, building chords, and understanding progressions by using gamification to encourage retention and interest. This application will be built using Flutter to target Android and iPhone users.

## Background and Context

Current traditional methods of learning music theory are slow and lack the feeling of achievement as students enrich their knowledge. Many good musicians have not had formal training and musical literacy is an underrepresented skill in the musical world.

## Objectives and Significance

*This project aims to develop a mobile application that teaches music theory in a fun, engaging, and accessible way. {Many good musicians have not had formal training and musical literacy is an underrepresented skill in the musical world.} This application provides an opportunity to increase musical literacy from the comfort of a mobile device.*

## Motivation for Proposal Choice

*I chose this project because I’ve noticed the increasing power of gamification for education and my knowledge of music theory. Bringing my background and coupling it with my career choice seemed like a fun and interesting project that could benefit other people of similar backgrounds.*

## Capstone Summary

*The HarmoniQ project will demonstrate a broad range of computer science concepts and skills, addressing an unmet need in the educational landscape. By developing a mobile application that teaches music theory through gamification, I will employ a variety of computer science disciplines such as mobile app development, algorithms, software engineering, UX design, data management, security practices, and testing and deployment. all while creating a solution to an existing gap in musical literacy.*

*Breadth of Computer Science Topics*

*To develop this app, I will apply knowledge from several upper-division computer science courses including:*

* *Algorithms(CPSC 335)*
* *Mobile Dev Programming(CPSC 411)*
* *Web Back-End Engineering(CPSC 449)*
* *Computer Communications(CPSC 471)*
* *Software Engineering(CPSC 362)*
* *Intro to Game Design(CPSC 386)*

*Project Magnitude*

*This project exceeds the scale of a typical upper-division final project due to its combination of multiple domains, such as mobile app development, backend services, and game design. It also includes unique challenges, such as real-time progress tracking, achievement systems, and multimedia content integration. Furthermore, the project’s focus on cross-platform deployment requires a thorough understanding of app optimization, scalability, and user interface consistency across different devices.*

*Unsolved problem or significant improvement to existing solutions*

*Currently, while music theory applications like MusicTheory.net and Teoria offer useful resources for self-learners, none provide an interactive, engaging, and novel experience to keep users engaged over time. The HarmoniQ app aims to fill this gap by offering an experience that motivates users to return regularly through daily challenges, streaks, and achievements, thereby addressing a significant need in the music education field.*

*I conducted a review of existing music theory websites such as MusicTheory.net, applications, and similar educational content. The results were that no application has effectively combined comprehensive musical literacy with a gamified approach to the same extent as HarmoniQ. The app’s unique blend of fun and learning makes it stand out by offering a user experience that both teaches and entertains.*

## Product Scope and Charter

## Description of the Product

*It's like Duolingo for music theory. This application is designed for music enthusiasts of all levels and curious individuals to enhance their musical literacy. It encourages user participation through achievements and streaks to create a game-like environment. You can sharpen your hearing, reading, and music creation skills through small daily interactions that reward you as you become a musical wizard!*

## Features and Functional Requirements

### Core Product

The product will offer the following features:

*FE-1: Users can learn how the basics of music theory which include notation, rhythm and time, major scales, keys, intervals, and chords.*

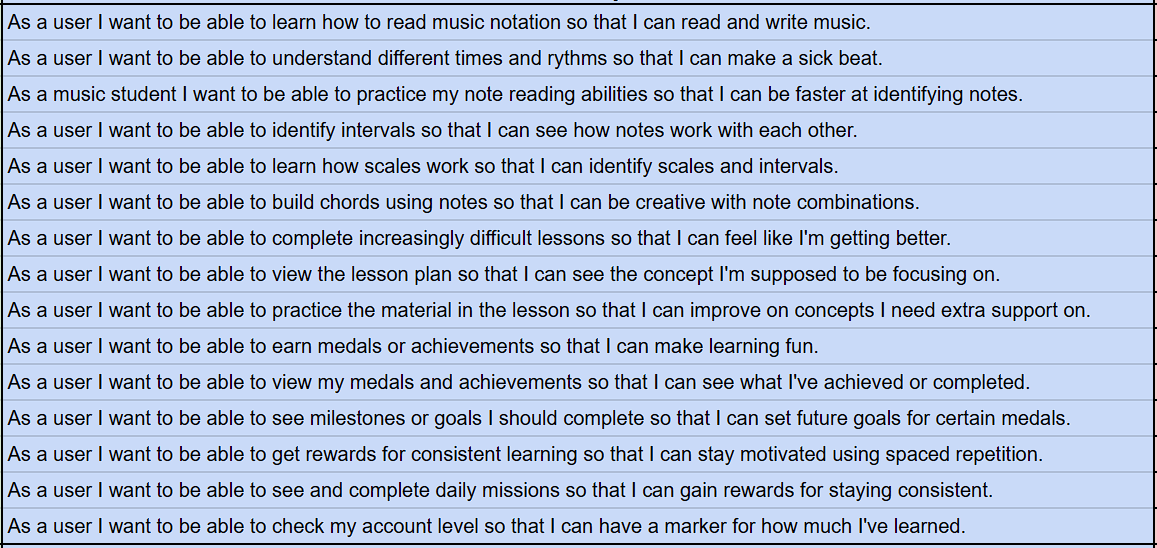
*FE-2: Users can progress through units in the course by completing lessons.*

*FE-3: Users will earn achievements and levels for milestones and progress.*

*FE-4: Users will get experience for consistent learning and activity using streaks and daily missions.*

*FE-5: Users can learn how to build basic harmonies and chord progressions.*

*Figure 2.2.1 User Stories*

**

### Stretch Goals / Phase 2

If the core product is completed in time to permit extra development, I plan to add:

*Minor Scales and Keys: The application will expand the concepts introduced by going into minor scales and keys and how they interact with their major counterparts.*

*Elephant Customization Rewards: As the user completes lessons and gets achievements, they will earn customization options for their mascot, Stuffy.*

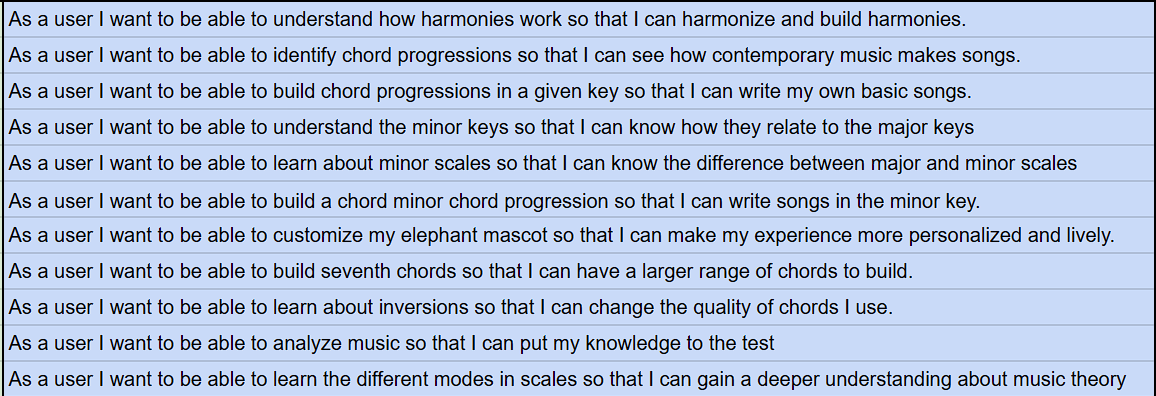
*Seventh Chords and Inversions: The application will cover seventh chords and inversions.*

*Musical Analysis: The application will feature royalty-free music that allows the user to apply music theory knowledge and expand on it using covered concepts in identification and harmony.*

*Modes: The application will begin to cover advanced music theory concepts such as modes, which include Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aeolian, and Locrian.*

*Basic CounterPoint: The application will add counterpoints and allow the user to experiment with modal melodies.*

*Figure 2.2.2 Stretch Goal User Stories*

**

## Non-functional Requirements

***Performance Requirements***

*Response: Smooth transitions between lessons and activities.*

*Scalability: The app should handle 1000 simultaneous users without crashing.*

***Usability Requirements***

*Ease of Use: The app should be intuitive, with clear navigation and easy-to-understand instructions. A first-time user should be able to navigate through the core functionality without external help. This will be tested by giving the app to friends and family who are new to it and 90% of them can experience the full application.*

*Device Compatibility: The app should work smoothly on the two most recent Apple and Samsung Devices mobile devices and adapt to standard screen sizes for phones and tablets.*

***Security Requirements***

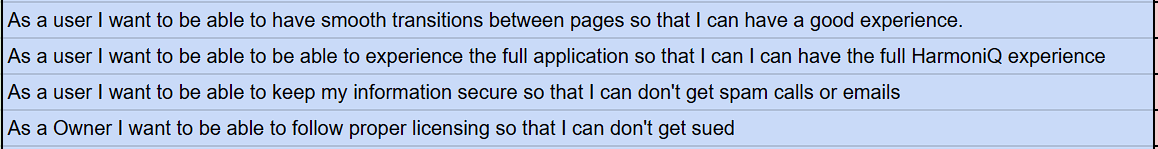
*Authentication: Use an industry-standard authentication system. (firebase)*

***Legal and Regulatory Requirements***

*Licensing: Ensure that the app's third-party libraries, music, or educational content are properly licensed for commercial use.*

*Compliance: The app must comply with relevant app store policies (e.g., Apple App Store, Google Play Store) for privacy, content, and security.*

*Figure 2.3 Non-Functional Requirements User Stories*

**

## Target and Operational Environment

*The target environments will be Android and iOS mobile systems(phones and tablets) running Android 8.0+ and iOS 12+ respectively.*

## Project Deliverables

*GitHub Repository:* [*https://github.com/Diegov5498/HarmoniQ*](https://github.com/Diegov5498/HarmoniQ)

*Deployed in Apple App Store and Google Play (Google and Apple)*

## Project Assumptions and Limitations

***Project Assumptions***

*Availability of Development Tools: It is assumed that all necessary development tools will be available throughout the project. This is a risk*

***Project Limitations***

*Content Scope: The app will cover only basic to intermediate music theory concepts initially. More advanced topics, such as composition and advanced harmony, might be considered for future versions.*

## Ethical Considerations

***Educational Integrity***

*Accuracy of Content: The accuracy of the information presented is critical since the application is educational. The music theory lessons must be correct, up-to-date, and not misleading. Providing wrong or incomplete information could negatively affect learners.*

***Intellectual Property and Copyright***

*Use of Music Content: The intellectual property rights of music creators should be respected if used. If any copyrighted music or educational materials are used, the appropriate licenses will be obtained.*

## Related Works and Concerns

## Literature Review

*MusicTheory.net*

*teoria.com/*

## Related Products and Systems

*Duolingo*

## Security Concerns and Approaches

*Data Security: All data concerns and security are being handled by Google’s Firebase*

*Compliance with Regulations: Users have the right to access, modify, or delete their data.*

## Relevant Standards

*General Data Protection Regulation (GDPR) for European users and the California Consumer Privacy Act (CCPA) in the U.S.*

## Stakeholders, Users, and Benefits

## Stakeholder Identification

*Bruce McKenzie(Advisor): Concerned with complexity, attainability, and completing the product.*

*Diego Vela(Owner): Concerned with quality, security, effectiveness, efficiency, and completion of the product.*

## User Identification

*Self-Taught Musicians: Users who learned music without any formal training.*

*Casual Users: Anybody with an interest in the application or learning music theory.*

*Musicians with Traditional Training: Users who have had limited or extensive formal training in music.*

## User Roles – their needs and the benefits gained

*Regular User: Users will expand their knowledge of musical theory which includes but is not limited to reading music sheets, building chords, and understanding chord progressions.*

## Internal User Roles

*None*

## Technical Architecture & High-Level Design

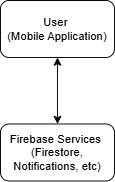
## System Architecture

*Client-Side (Flutter Mobile App): The user-facing application where learners interact with lessons, exercises, and their progress.*

*Backend (Firebase): Provides backend functionality, including data storage and tracking progress.*

*Interaction: The user will authenticate via Google or Apple accounts. Users engage with the lessons and exercises and the app collects data on user activity. Progress data will be stored locally to allow offline access. When an internet connection is available, the app automatically syncs data with the Firebase backend, storing user data. The Firebase Cloud Functions can trigger actions based on activity, such as updating progress or notifying users about streaks.*

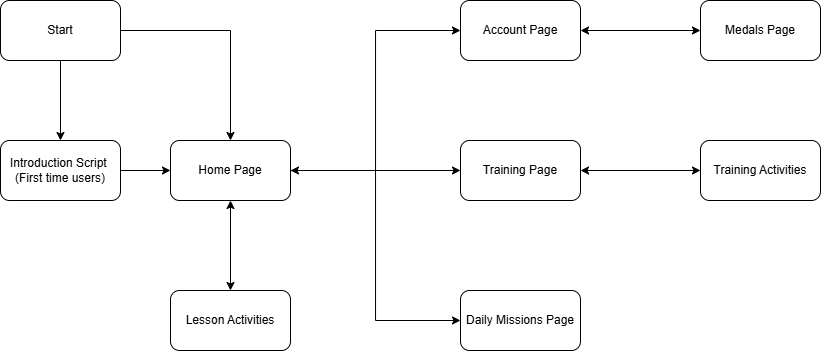
*Figure 5.1.1 System Arch Diagram*

**

*This system architecture ensures the app is scalable, user-friendly, and secure while providing offline capabilities and real-time synchronization. By leveraging Firebase’s services, the app will handle many users efficiently, meeting functional and non-functional requirements.*

## High-Level Design

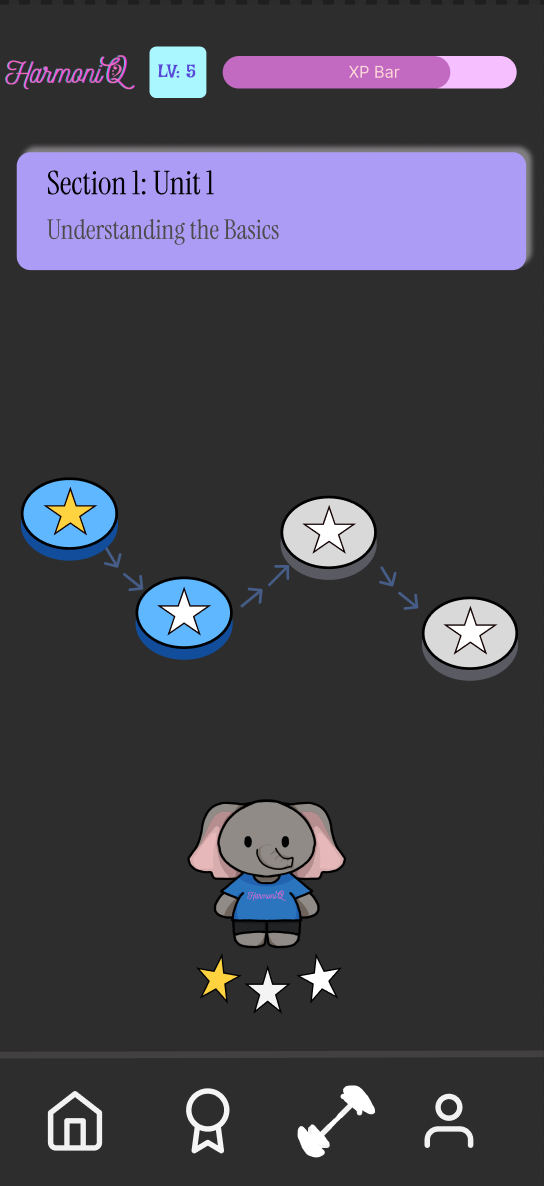
*Figure 5.2.1 UX Flow Diagram*

**

*Figure 5.2.2 Logo*

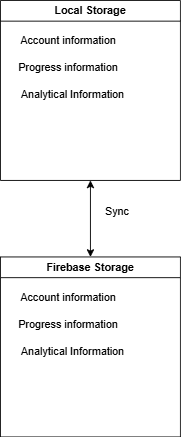
**

*Figure 5.2.3 Design Prototype: Main Page*

**

## Data Stores

*Stores user data of the accounts into the NoSQL database of Firebase and their progress is also stored locally for offline use.*

**

## Frameworks, APIs, Components, Services, and Tools Used

*Flutter SDK: The core framework for developing cross-platform mobile apps using Dart. Flutter allows you to build native apps for iOS and Android with a single codebase. It comes with a rich set of pre-built widgets for fast UI development.*

*Dart: The programming language used with Flutter. It supports both functional and object-oriented programming styles, and it is optimized for building user interfaces.*

*Firebase: For user authentication; users can log in, register, and manage their accounts. To store user data, such as progress tracking, quiz scores, and lessons. Firebase will handle the backend.*

*Widgets: Custom widgets for specific UI elements, such as quizzes, progress bars, or interactive exercises. To also manage dynamic layouts.*

*Android Studio and Visual Studio Code: IDEs commonly used with Flutter.*

*Git and GitHub/GitLab: For source control and collaboration to track changes, work with a team, and manage different versions of the app.*

*CI/CD (Continuous Integration/Continuous Deployment): GitHub Actions or CircleCI for automating builds and tests, ensuring code quality before deployment.*

*Figma: For designing the user interface, prototyping interactions, and collaborating with your team on the visual elements of the app*

## Project Management

## Team

*Diego Vela: Project Manager, Scrum Master, Developer, Stakeholder*

## Methodology

*The idea for HarmoniQ originated from my background in both music theory and software engineering. After conducting informal surveys with family members and friends, I found that many self-taught musicians have limited knowledge of fundamental music theory concepts, such as reading sheet music, understanding scales, and building chord progressions. Traditional music theory methods are often slow and lack engaging, interactive lessons that motivate learners. Drawing on my interest in music theory and the success of gamification in apps like Duolingo, I was inspired to create an app that blends music literacy with a game-like experience.*

*To validate the app concept, I plan to follow an iterative approach, beginning with feedback from the family and friends I initially surveyed. This group will help refine the early versions of the app, after which testing will expand to a larger, more diverse audience.*

*Feature prioritization was based on my understanding of the music literacy process. I started with core subjects essential for learning music theory, followed by gamified features designed to make the learning process enjoyable and engaging. Finally, I added a feature to bring all concepts together in a cohesive learning experience. These features will be continuously revised and refined based on user feedback in each iteration until the product is both effective and efficient.*

*The project will follow an Agile methodology, enabling flexibility and iterative improvements. Features will be prioritized according to user needs and feedback, with regular revisions to the development roadmap. This ensures that the app will evolve to meet users’ expectations and address gaps in current music theory educational tools.*

## Project Timeline

*Work begins on December 8th. The first shippable version is due on December 24th.*

## Subprojects

*None*

## Risks and Feasibility

***Time Constraint***

*Risk: The time constraint of the project may be too tight for a solo developer.*

*Risk Management: The timeline will be clearly defined and spaced out reasonably to maintain progress. Additionally, work will begin in early December.*

***Knowledge Limitations***

*Risk: The current knowledge level of music theory may be short of what is required for a project of this scale.*

*Risk Management: Extra time will be focused on gathering requirements and researching similar applications, websites, and programs.*

## Release Plan

*The application will be delivered on the App Store and Google Play.*

## Evaluation and Assessment

## Success Criteria

*Success will be measured by including the five core features and deploying them into both the Apple App Store and Google Play Store within the time constraint.*

## Testing Plan

*Each feature will be tested independently by me, my family, and close friends. When the 5 core features are implemented, a larger test group that includes my friends and acquaintances will be used.*

User Evaluation

*A feedback feature will be added for a test release of the application for users to conveniently send feedback.*

## Measurement of Benefits

*Users will know the basics of music theory which include reading the clefs, identifying the notes, and understanding the general concept of keys within the first hour of use.*

## Glossary

*API: Application Programming Interface; is a set of protocols, tools, and definitions that allow different software applications to communicate with each other.*

*Duolingo: A popular mobile app that uses a gamified approach to foreign language education.*

*Agile: An approach to project planning that involves daily meetings, iterative development, and a team-based approach.*

*Waterfall: A traditional project management and software development methodology that follows a linear, sequential process. The project progresses through distinct, non-overlapping phases, where each must be completed before moving on to the next one.*

*UI: User Interface; how the application looks.*

*UX: User Experience; the experience of users when using the app.*

## Appendix

## Appendix 1: Product Summary

[*ProductSummary-HarmoniQ*](https://docs.google.com/spreadsheets/d/1_6-TWhniRoHW6XOZdPjNxD9WXZow4ibEcd-0Pv2MgCg/edit?usp=sharing)

## Appendix 2: ECS Expo Abstract

### Project Background

*The HarmoniQ mobile application aims to improve music theory literacy by offering an interactive, gamified learning experience. The concept was inspired by my background in music theory education and the success of gamified platforms like Duolingo. Many musicians, especially those without formal training, face difficulties with core concepts like reading sheet music, understanding scales, and constructing chords. Current educational methods lack engagement and interactivity, leading to slower learning and decreased retention. HarmoniQ seeks to fill this gap by offering a fun, dynamic, and effective approach to mastering music theory.*

### Problem Statement

*Self-taught musicians and casual learners often struggle to grasp fundamental music theory concepts due to the slow pace and lack of engagement in traditional learning methods. There is a lack of educational tools that combine interactive learning with motivation-driving features, such as gamification, to encourage long-term retention and consistent usage.*

### Goals and Objectives

*The primary goal of HarmoniQ is to create a mobile app that teaches users the fundamentals of music theory engagingly and enjoyably. The objectives include:*

* *Providing users with interactive lessons on key music theory topics such as notation, rhythm, scales, and chords.*
* *Motivating learners through gamification elements like achievements, daily streaks, and quizzes.*
* *Offering a structured learning path that encourages user engagement and improvement.*

### Design Requirement/Specifications

*HarmoniQ will be developed using Flutter, enabling cross-platform deployment on both Android and iOS devices. Key design features include:*

* *User Interface: Clean, intuitive UI designed to provide an enjoyable user experience for learners of all ages and skill levels.*
* *Offline Access: Users can continue their learning experience even without an internet connection, syncing data when they are online.*
* *Gamification: Features like daily streaks, achievements, and levels will motivate users to remain consistent and progress through lessons.*
* *Scalability: The app will be designed to support a large number of users, with a backend powered by Firebase to handle user authentication and data storage*

### Methodology

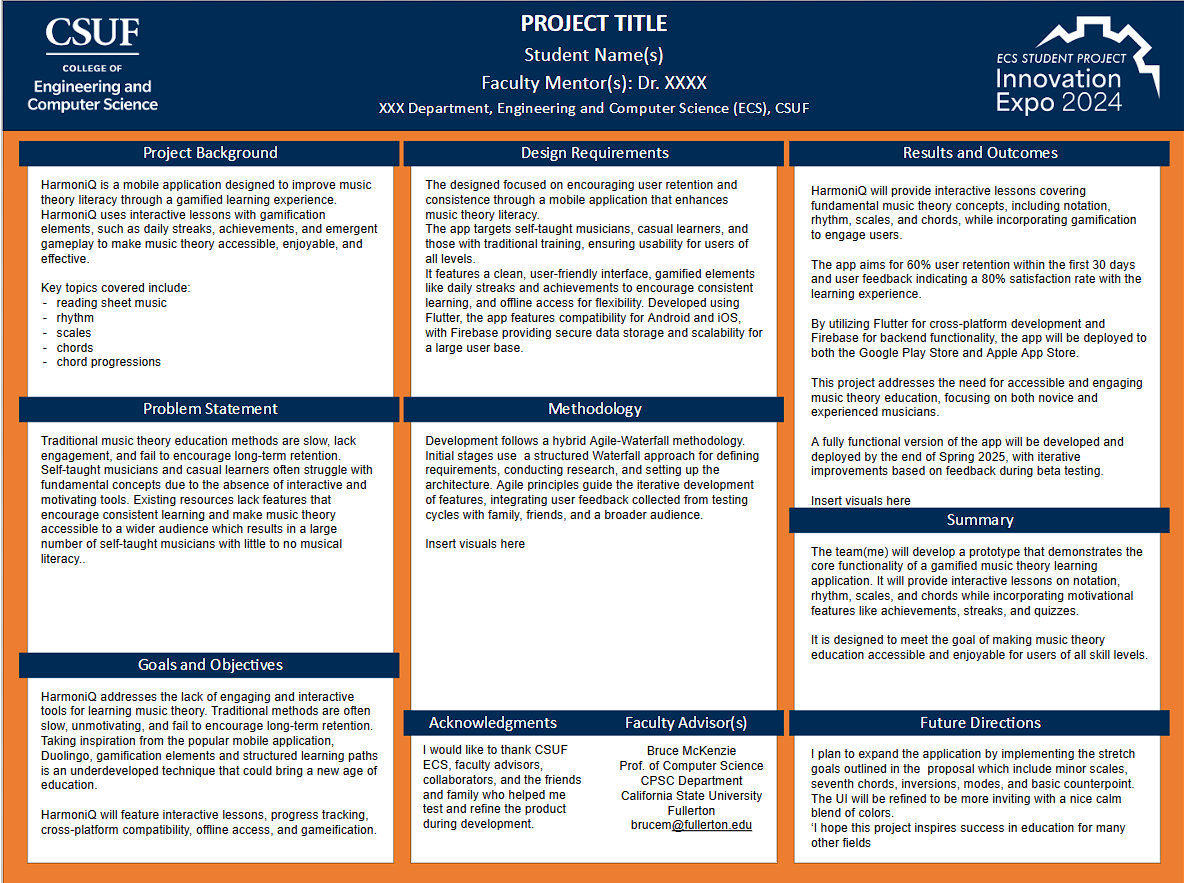
*The development of HarmoniQ will follow a hybrid methodology that combines both Agile and Waterfall approaches. This combination will allow for the flexibility and iterative improvements of Agile while providing a structured, step-by-step process in the early stages of the project using Waterfall.*

### Results or Outcomes

*The expected outcome of this project is a fully functional mobile application that makes learning music theory accessible, enjoyable, and effective for a wide range of users. The app will be deployed to both the Apple App Store and Google Play Store, offering users an interactive tool to improve their musical knowledge at their own pace. The success of the app will be measured by success in deployment.*

## Draft ECS Poster

[HarmoniQecs-expo-poster.pptx](https://docs.google.com/presentation/d/10bqZsLcIBoDtzWg1KDotHyW0uSvDkVHu/edit#slide=id.p1)



## References

None