# CEN4010 Principles of Software Engineering

### Spring 2021

# Milestone 1 Project Proposal and High-level Description

## Group 2: Team Rocket

## Project name: *Apollo Melodies*

### Team Members

Ivan Bruno-Gaston (Product Owner) - ibrunog@fau.edu

Craig Peroni (Scrum Master) - cperoni2020@fau.edu

Sharmada Iyer (Developer) - siyer2018@fau.edu

Belal Mahmood (Developer) - bmahmood2018@fau.edu

Isabel Tait (Developer) - itait2017@fau.edu

### Date: 2/16/2021

|  |  |
| --- | --- |
| Revision Number | Date |
| 1.0 | 2/16/2021 |

1. Executive Summary

**Introduction**

Listening to music is a hobby to many people. There are many music platforms that allow users to enjoy music uploaded by several artists. However, most of these platforms rarely use music as a fun and competitive interaction. The COVID-19 pandemic has restricted outdoor fun activities. The internet consists of games that are fun and competitive but only very few incorporate music. The purpose of this project is to use music in the form of a game to unleash a competitive environment between users.

2. Competitive Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Single Player | Music Player | feature3 | feature4 | feature5 |
| Apollo Melodies | ✅ | ✅ |  |  |  |
| Competitor 1  (Kahoot??) | ✅ |  |  |  |  |
| Competitor 2  (Quizlet) | ✅ |  |  |  |  |
| Etc |  |  |  |  |  |

Summarize advantages or competitive relationships to what is already available.

3. Data Definitions

Music quiz - A series of multiple choice questions related to the series of songs that are played on the webpage.

Score - the total amount of right answers a user has achieved on the quiz which is stored in their profile database.

Point-The number of questions answered correctly.

4. Overview, Scenarios, Use Cases

Scenario 1

John went to go tested for COVID-19 last Friday. He ended up testing positive so he began his quarantine. At first he accomplished all the things he didn’t do then he began to get really bored. He has a strong interest in music and decided to check out Apollo Melodies to test his knowledge on music. He scrolled through and saw the genre he was comfortable with and began to play a game where they test your knowledge about a song lyric or short clip of. He enjoys the product.

Scenario 2

Kate and Jill were having a competition about who knew the most songs. They go to Kate’s house and go to Apollo Melodies to play. They listen to quick snippets of songs and answer the questions that follow. They keep track and see who wins.

5. Initial High-Level Functional Requirements

1.

1.1

2.

2.1

6. List of Non-Functional Requirements

*6.1 Security*

Since the application is not expected to contain any Personally Identifiable Information (P.I.I.) and will not facilitate any financial transactions, security risk to the site is considered to be low. Therefore a moderate level of password encryption will be used to encrypt password data. Passwords must contain at least 9 characters with upper and lower characters. A number must also be included in the password for it to be accepted.

Validation of passwords based on password rules will be accomplished with JavaScript on the front end of the app. The encryption algorithm will be executed in PHP on the server-side of the application.

*6.2 Usability*

The target customer for this application will be a user with moderate computing skills. The user should not need advanced computing skills in order to use the product. GUI design should be simple and screen hints should be available to guide users through the process

*6.3 Reliability*

User volume on this application is not expected to be high and, thus server up-time is considered to be moderately critical. The expected downtime (other than hardware failures) will likely be due only to pushing new implementations of the product from the development environment to the production environment. Dev-to-Prod pushes are expected to bring the site offline no more than 5 minutes. Dev-to Prod pushes will be off-cycle throughout the duration of the project. However, these pushes will take place after hours.

6.4 Maintainability

Admin users will be created for this site. Admin users will have the ability to make basic configuration changes to the site, and manage regular users on the site. An admin user should be able to make basic configuration changes to the site through the front end GUI without the need to directly program to an API.

6.5 Performance

* 6.5.1 - **Page Loading Time** - Page loading times should take no longer 90 seconds on a computer or mobile device with a current browser and internet speed of 4G or more
* 6.5.2 - **Concurrent usage** - The site should be able to support at least 1000 concurrent users operating on the site
* 6.5.3 - **UI Responsiveness** - The display should adapt to the size of the viewing port within 10 seconds of full page load.

7. High-Level System Architecture

*7.1 Overview*

In this section of the proposal,we will detail the technology stack used to implement the software solution. All tools from the hardware to the client-facing GUI will be detailed. The supporting technology will be configured as an ecosystem where this or multiple applications (or application components) can share real time access to the underlying databases

*7.2 Hardware*

The hardware which will host the application is a LAMP server provisioned by Florida Atlantic University for our group to use. This Linux server implements Apache, MySQL, and PHP Application development environments (hence L.A.M.P.). No other hardware will be used to implement the application.

*7.3 Database Utility*

To build the application databases, we will use the MySQL implementation on the LAMP server. One database will be developed and tables will be created for member data, activities, and administrative needs. This database will be relationally implemented. The unique key and relational key for all database tables will be the unique user ID.

*7.4 Server-Side Scripting*

In order for the application info interface between the client-facing GUI and the database, PHP will be used to perform the server-side scripting. This includes validating and encrypting the user login, creating new users, and adding/modifying content at the user level. Other features which emerge during the progression of the project will also use PHP.

*7.5 Client-Side Scripting*

Any client-side scripting will be done using JavaScript and CSS. This includes animation, front-end form validation, and any other user experience elements.

*7.6 “Eye-Level” Graphical User Interface (GUI)*

The front end GUI for the application will be Coded in HTML5 and CSS by implementing a mobile-responsive Bootstrap template acquired from <https://startbootstrap.com/>. The actual final template chosen will be decided later in the project. Proper credit for the template used will be listed on the site.

*7.7 Quiz Module*

Google forms will be used to build out each song quiz. Quizzes will be administered within our site using Google APIs to bridge interactions between the two applications. APIs will be implemented using JavaScript.

8. Group 2: Team Rocket

Ivan Bruno-Gaston (Product Owner) - [ibrunog@fau.edu](mailto:ibrunog@fau.edu)

Craig Peroni (Scrum Master) - [cperoni2020@fau.edu](mailto:cperoni2020@fau.edu)

Sharmada Iyer (Developer) - [siyer2018@fau.edu](mailto:siyer2018@fau.edu)

Belal Mahmood (Developer) - [bmahmood2018@fau.edu](mailto:bmahmood2018@fau.edu)

Isabel Tait (Developer) - [itait2017@fau.edu](mailto:itait2017@fau.edu)

About Us

<https://lamp.cse.fau.edu/~cen4010_s21_g02/>

9. Checklist

|  |  |
| --- | --- |
| Method of Communication Decided | DONE |
| Team Found a Timeslot to Meet | DONE |
| Front and Back End Team Leads Chosen |  |
| Github Master Chosen | DONE |
| Team Ready/Able to use front/back-end Frameworks |  |
| Skills of Team Members Defined and Known by All | DONE |
| Team Lead ensured final M1 Reviewed by All | On Track |