CEN 4010 Principles of Software Engineering

Milestone 1: Team Project Proposal and Description

Team Name: Core5

Project Name: CoreShare

Team Number: 25

September 25, 2021

**Team Members**

Mac Consolabe

mconsolabe2020@fau.edu

-

Victor Marinho Correa

jmarinhocorrea2021@fau.edu

-

Greydi Lora

Glora2019@fau.edu

-

Dominic Robbins

Drobbins2019@fau.edu

-

Michael Cevallos

Mcevallos2020@fau.edu

**History Revision Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Actions | Performed by |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Florida Atlantic University

Dr. Shihong Huang

shihong@fau.edu

Instruction:

This is a team project. Each team submits one document to Canvas

This document is the first milestone in your term-long project. You will propose a term project that will be developed and deployed over the course. Your team decides on the specific theme and features of your project, or you can work on the project that I have proposed for you (see Project Description on Canvas). If you choose your own project, please have instructor’s approval before proceeding.

Your term project proposal and description should include at least the following sections, based on which you can add more sections when you see fit.

Table of Contents

[1. Executive Summary 2](#_Toc83734478)

[2. Competitive analysis 3](#_Toc83734479)

[3. Data definition 4](#_Toc83734480)

[4. Overview, scenarios and use cases 6](#_Toc83734481)

[5. Initial list of high-level functional requirements 7](#_Toc83734482)

[6. List of non-functional requirements 8](#_Toc83734483)

[7. High-level system architecture 9](#_Toc83734484)

[8. Team 11](#_Toc83734485)

[9. Checklist 12](#_Toc83734486)

# Executive Summary

CoreShare is a social media web application that will be geared towards connecting people across the world that are far away from each other, to remind them, and help them reminisce on old memories, with the ability to create new ones. Throughout a time like we’re in right now, with COVID, there are a lot of different platforms that are aimed to connect people, mind you people probably won't even need a social media platform, they can just call each other. but with CoreShare the aim is to give them the opportunity to connect on a level that's deeper than just a phone call, or a simple text message, or even a tweet.

One of the biggest ways to connect with people that exists in this world is music.

Emotions and feelings sometimes cannot be described simply with words, being far away from your loved ones and not being able to talk & connect can sometimes make you lost for words for when you actually do get in touch with them, but sometimes just one song shared can completely transform this melancholy into happiness by reminiscing on old memories or even by creating a bridge of sympathy between yourself & another.

CoreShare’s goal is to develop that bridge, a bridge with many lanes, not simply to share sad feelings with others, but also to transfer more upbeat feelings with others as we’re stuck at home most of the day, yearning for something to do, experience, & feel, by connecting Music and People on a level that has never been done before. The purpose of this application is to allow you, me, & everyone else to have the ability to connect with our loved ones by just sharing a song, and it doesn’t stop there because the app is made for CoreConnection, not only by sympathy, but by curiosity, simply wanting to try something new, perhaps feel something new, and by new we mean new music.

We will give you the ability to not only connect directly with your friends and family but also to connect indirectly by providing you with the ability to learn more about your family and friends taste in music, even if it doesn’t involve a deeper connection. Everyone will be able to connect in such a way where, so long as someone is in your CoreCircle on the platform, they can tap into your experience & feelings by seeing what you are currently listening or have listened to in the past and in the process develop their own Unique experience & feelings toward new music.

All the experience, connection and sharing will be made inside the platform, without having to leave and grab a song somewhere else and come back. Everything will be within your grasp, your CoreCircle and music, all in the same place. Connection, love, harmony is the fuel for the application, but giving you a good user experience is also in our top priority.

# Competitive analysis

**Analyzing competitive products available today. Present competitors’ features vs. your planned ones. First, create a table with key features of competitors vs. yours. Only at very high level, 5-6 entries max. After the table, you must summarize what are the planned advantages or competitive relationship to what is already available.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Features/Apps** | **CoreShare** | **Twitter** | **Spotify** |
| **Communication** | **X** | **X** |  |
| **Music** | **X** |  | **X** |
| **In-app Music sharing** | **X** |  |  |

The ability to communicate already exists within one of our competitors’ products, such as Twitter, & the ability to give people music does as well, such as with Spotify, but having the ability to do both within a single platform is what CoreShare is about. With the In-App music sharing feature CoreShare’s advantages on these two platforms is its biggest strength. Bringing something that is already available is easy but bringing something new to the table is not and this is what we’re doing at CoreShare.

# Data definition

**This section serves as the “dictionary” of your document. It defines main terms, data structures and “items” or “entities” at high or logical (not implementation) level (e.g., name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements, actors, types of users etc.) specific for your application and not on general well know terms. These terms and their names must be used consistently from then on in all documents, user interface, in naming software components and database elements etc. In later milestones, you will add more implementation details for each item. You will later expand this section with more details.**

Framework: The essential supporting structure of the application. Frameworks consist of a mobile application using Bootstrap, Cordova, and Django.

UI: The user interface are access points where users interact with the designs.

User: Anyone who uses the application and has the ability to share music, make requests, and send messages.

Database elements: Database schema, schema objects, tables, records, indexes, fields, data types, keys

Database information: User ID, email, password, name, friend list

CoreCircle: Friends in your circle in which you both have accepted a friend request from.

Sign up: User will create an account using an email and password. Will have access to all features.

Login: Users can access their account using their email and password that is already registered into the system.

Search: Look up other users and send friend requests.

Post: Post a song and optionally include text.

Delete post: Deletes the user’s previous post.

Remove friend: Removes friend from your CoreCircle including theirs.

Send Friend Request: Requests another user to be added to their CoreCircle and vice versa.

Accept Friend Request: User is added to their friend’s CoreCircle. The user’s friend is added to their CoreCircle.

Deny Friend Request: User cannot be added to their CoreCircle.

Ignore Friend Request: Ignores the request and nothing is sent

Homepage: The main page where users can view their friend’s shared posts and music.

Friends: A list of friends in the user’s CoreCircle.

View Music: View what music the user’s friends are listening to including their history.

Messages: Access messages sent or received by other users in their CoreCircle.

Reply: Send a direct message to the user.

# Overview, scenarios and use cases

**This section describes the project overview (in much more details) and likelihood usage scenarios of your product from end users’ perspectives. Focus only on main use cases. Simple text format is OK and preferable – tell us a story about who and how is the application used. Focus on WHAT users do, their skill level, not on HOW the system is implemented. You can expand use cases provided in high level document in future milestones.**

The first main use case of the product will be the ability for users to have their own account, and within that account users will be able to add people such as family, friends, and even people that they meet on the platform. Once the user has an account and friends linked to his account the user will be able to unlock other features such as being able to create a CoreCircle.

The CoreCircle is a feature that allows the user to be able to view, listen, and communicate with a select group of people that the user had added in their CoreCircle that are considered the user’s closest friends.

This feature will allow the user to view, listen, and communicate with their closest friends in a private manner where everyone will be aware of who is in their CoreCircle and whose CoreCircles they’re in too.

The CoreCircle Request will be like any other friend request, except accepting it means, you’re letting me into your music world, such that if you send a CoreCircle request, the other person must accept it in order to add them to your CoreCircle, this adds yourself to their CoreCircle as well.

There will be of course a home tab where everyone that wants to share a specific song that they feel comfortable with everyone connected to the platform will be able to do so.

The last use case will be a feature for the future part of the project, which will be Direct Message section, where you can communicate with a specific person, including regarding a specific song that they may have posted, or that you may want to send, and start a conversation.

# Initial list of high-level functional requirements

**This refers to the high-level functionality that you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep the users in mind. Develop these functions to be consistent with use cases and requirements above. Number each requirement and use these numbers consistently from now on. For each functionality use 1–5-line description.**

1. User can sign up for an account
2. User can log into account
3. Users can friend other users by searching for them in the friend search bar
4. User can post a song with or without a message for followers to see
5. User can delete posts
6. User can send your first CoreCircle request to create your CoreCircle
7. User can accept, deny, or ignore CoreCircle requests sent to them
8. If user accepts friend’s CoreCircle request, then they are added to their friend’s CoreCircle & their friend gets added to their own CoreCircle
9. User can remove a friend from their CoreCircle, which removes themselves from their friend’s CoreCircle
10. Users can view what their CoreCircle friends are listening to currently, if they are, or what they have listened to recently.
11. Users can reply to their friend’s post by direct message

# List of non-functional requirements

**For example, performance, usability, accessibility, expected load, security requirements, storage, availability, fault tolerance etc. Number each. When possible, try to quantify these quality attributes.**

1. The very first requirement is that the service should be highly available. The users should be able to send songs and should be able to see other users’ songs posted on their home timeline without any downtime.

2.Generating the timeline should be fast within half a second (at most)

3. The system does not need to be strongly consistent--eventual consistency is fine.

* It will be okay to see other user songs post after some time

4.The system should be scalable with the increasing data load.

* Increasing user
* Increasing song post

5.User data (user information and song post/ and friends) should be durable

* It would not be good if a user login and find out that some of his/her songs post are gone or the users he was friends with, he is not long friend with them anymore

# High-level system architecture

**Lists of main software products, tools, languages, and systems to be used, list of core APIs available at this point, supported browsers etc.**

**You also have to decide on which frameworks you will use if any. These provide both user interface, as well as cross-platform and cross browser layout/CSS. All external code you plan to use must be listed along with their license.**

Lists of main software products:

* Visual Studio Code
* GitHub
* Git
* Cordova
* PyCharm
* VMWare
* WinSCP
* MySQL

Languages:

* Html
* CSS
* JavaScript
* Python
* C++
* SQL

API

* Spotify API
* SoundCloud API

Supported browsers:

* Chrome
* Brave
* Edge
* Mozilla Firefox
* Opera

Frameworks:

* Bootstrap
* Cordova
* Django

# Team

**List student group names, name of Scrum master, product owner and initial roles for each member**

Scrum master - Mac Consolabe

Product owner - Greydi Lora

GitHub master - Dominic Robbins

Development team -

1. Frontend:
   1. Victor Marinho Correa
   2. Greydi Lora
2. Backend:
   1. Mac Consolabe
   2. Dominic Robbins
   3. Michael Cevallos

# Checklist

**For each item below you must answer with only one of the following: DONE, ON TRACK (meaning it will be done on time, and no issues perceived) or ISSUE (you have some problems, and then define what is the problem with 1-3 lines). Reflect these items in your Jira project space:**

**a) Team decided on basic means of communications** - DONE

**b) Team found a time slot to meet outside of the class** - DONE

**c) Front and back-end team leads chosen** - DONE

**d) GitHub master chosen** - DONE

**e) Team ready and able to use the chosen back and front-end frameworks** -

ON TRACK

**f) Skills of each team member defined and known to all** - DONE

**g) Team lead ensured that all team members read the final M1 and agree/understand it before submission** - DONE