API ARCADE

SOFTWARE REQUIREMENTS DOCUMENTATION (SRD)

1. INTRODUCTION

1.1 TITLE

1.1.1 PROJECT NAME

API Arcade

1.1.2 TEAM NAME

The Elon Musketeers

1.1.3 DATE

Project Started: 08/21/2020

1.1.4 TEAM MEMBERS

Ryan Cabell, Andrew Kawabata, Francisco Serratos-Prudencio

1.1.5 STAKEHOLDERS/COMPANY - HONOR CODE

Academic integrity is founded upon and encompasses the following five values: honesty, trust, fairness, respect, and responsibility. Supporting and affirming these values is essential to promoting and maintaining a high level of academic integrity.1 Each member of the academic community must stand accountable for his or her actions. As a result, a community develops in which students learn the responsibilities of citizenship and how to contribute honorably to their professions.

If knowledge is to be gained and properly evaluated, it must be pursued under conditions free from dishonesty. Deceit and misrepresentations are incompatible with the fundamental activity of this academic institution and shall not be tolerated. Members of the UNCG community are expected to foster in their own work the spirit of academic honesty and not to tolerate its abuse by others. Responsibility for academic integrity lies primarily with individual students and faculty members of this community. A violation of academic integrity is an act harmful to all students, faculty and, ultimately, the University.

1.2 TABLE OF CONTENTS

1.3 PURPOSE

This application serves as a hub of minigames for the users to test their knowledge of movies and trivia. Users can undertake a variety of different games, each one containing different elements and rules giving the user a unique experience every time. Users can also compete with friends for the top spot on the leaderboard and title of trivia master. This is all supported by our robust login system, that allows for unique account creation, which saves all of the users progress in-game.

1.4 DOCUMENT CONVENTIONS

N/A

1.5 INTENDED AUDIENCE

This document is for internal use only. Those looking to work on the application will find this document helpful in understanding the current state of the application and identifying what areas need to be addressed in future updates.

1.6 DEFINITIONS/JARGON

JVM refers to Java Virtual Machine. API is Application Programmer Interface. Desktop Application means a program that can be run on a desktop computer with the proper dependencies.

1.7 PROJECT SCOPE

The scope of this project will consist of building a fully functioning arcade desktop application with various technologies incorporated throughout the application from the ground up. The technologies included in the application are the integration of various API's pulling information for the games and the use of a database to store usernames, passwords, high scores, etc.

1.8 TECHNICAL CHALLENGES

This project marks the first time everyone in this group will be working with API's and incorporating them into an application. We will also implement software engineering principles like MVC and many more to create an optimized application. Another challenge we face is the implementation of a database while incorporating the CRUD and SOLID principles.

1.9 REFERENCES

Learning Github: https://www.youtube.com/watch?v=SWYqp7iY_Tc

Scene builder: https://www.youtube.com/watch?v=Z1W4E2d4Yxo

Scene switching: https://www.youtube.com/watch?v=5yQbt6lYRqk

2. OVERALL DESCRIPTION

2.1 PRODUCT FEATURES

API Arcade features include a leaderboard, login system, and the games themselves. The leaderboard is a quick and efficient way for our users to compare scores and compete with other users. The login system complements the former by allowing the user to register an account that will save all of the user's progress, username, and high scores. The games will feature options like the format of the game, difficulty choice, length of the game.

2.2 USER CHARACTERISTICS

Users will mostly consist of hardcore trivia buffs looking to test their knowledge in a wide range of topics. Causal users will also enjoy testing their knowledge with options directly geared for them.

2.3 OPERATING ENVIRONMENT

API Arcade is intended for use on desktop for entertainment purposes this can be in home, school, or even at work. API Arcade can be played anywhere there is a computer with a basic internet connection.

2.4 DESIGN AND IMPLEMENTATION CONSTRAINTS

Given that none of the group members are familiar with MySQL or other database technologies, we decided to use a CSV file to hold all our database information. We also decide to keep this a full Java application, for easier team collaboration.

2.5 ASSUMPTIONS AND DEPENDENCIES

API Arcade will continue to be updated with new games in the future, all of which will include an API to support its function. It is possible that many of the games will need to be updated to work with different API's if the current one is no longer working properly or is discontinued. The ReadMe on our GitHub repository lists the required dependencies and their versions. If these versions of these packages somehow become unavailable, this could pose a problem, however this is unlikely to happen anytime soon.

3. FUNCTIONAL REQUIREMENTS

3.1 PRIMARY

The primary functions of API Arcade include playing games to get a high score, and being able to view the current high scores of each game. The games are made to test the user's knowledge of movies and trivia

3.2 SECONDARY

Secondary functions include the register/login system, API translators, and database translators. The API translators are what pull data from an online database which the application uses to form questions for the user to answer during a game instance. API's are self-contained and can be replaced with minimal changes to the code. The database translator implements the CRUD operations on our application's database, which is currently in the form of csv files. The registration system allows someone to create an account with a unique username. The login system authenticates a user and stores their session so their stats can be saved in the database. There is also a reset password feature which allows a user to reset their password by providing their username and desired password.

4. TECHNICAL REQUIREMENTS

4.1 OPERATING SYSTEMS/COMPATIBILITY

API Arcade is compatible with any machine capable of running Java Virtual Machine. Windows and MacOS compatibility are confirmed for release date.

4.2 INTERFACE REQUIREMENTS

4.2.1 USER INTERFACE

Primary form of user interaction will be through keyboard and mouse.

4.2.2 HARDWARE INTERFACE

There is no specific hardware necessary for playing API Arcade.

4.2.3 SOFTWARE INTERFACE

API Arcade connects via internet to two different API's, OMDB and OTDB.

4.2.4 COMMUNICATIONS INTERFACE

N/A

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

The application load times should not exceed the average time it takes to play a single game. Ideally the application will consistently run-in linear time.

5.2 SAFETY/RECOVERY REQUIREMENTS

Currently there is no system to recover a user's score if the game abruptly crashes and there are currently no plans to implement such a feature. There is a feature for users to recover their password if it is forgotten.

5.3 SECURITY REQUIREMENTS

The security measures are very minimal, given that we do not require any sensitive data from the user to create an account. Besides the user's username and password, their high scores are the only other piece of data that we save to the database.

5.4 POLICY REQUIREMENTS

N/A

5.5 SOFTWARE QUALITY ATTRIBUTES

5.5.1 AVAILABILITY

API Arcade is available 24/7 so long as the user is connected to the internet and the API's used for core functionality are in working order.

5.5.2 CORRECTNESS

If the application fails to upload a score or pull up a question it will not be the end of the world.

5.5.3 MAINTAINABILITY

In its current stage updates will be available on a weekly basis to fix bugs and add new features.

5.5.4 PORTABILITY

Currently API Arcade is cross-platform and will work on any machine that can run JVM

5.6 PROCESS REQUIREMENTS

5.6.1 DEVELOPMENT PROCESS USED

Agile development

5.6.2 TIME CONSTRAINTS

The only time constraint is the four-month deadline to plan and code a working version of our app while applying all learned software engineering principle.

5.6.3 COST AND DELIVERY DATE

The full cost of the current build includes the cost of tuition for the entire group of software engineers. The final version of API Arcade is expected for release on December 02, 2020 at 7PM est.