Elysium: A New Generation Music Sharing Application

Code Review and Merge Policy

Garrett Phillips — Benjamin Boardley — Jack Gardner
Purdue University, Elmore Family School of Electrical and Computer Engineering
{gwphilli@purdue.edu — bboardle@purdue.edu — gardne97@purdue.edu}

January 2024

1 Introduction

This document outlines the repository architecture, code review policy, and merge polices intended for the developers of the Elysium project.

2 Repository Architecture

```
$ tree -S
.
|- aws/
|  |- scripts/
|- elysium_backend/
|  |- elysium_backend/
|  |  |- settings.py
|  |  |- urls.py
|  |- media/
|  |- music/
|  |- social/
|  |- user/
|  |- db.sqlite3
|  |- requirements.txt
|  |- Dockerfile
|- elysium_frontend/
|  |- android/
|  |- ios/
|  |- node_modules/
|  |- src/
|  |  |- components/
|  |  |- screens/
|  |  |- navigation/
|  |  |- App.js
|  |  |- index.js
|  |- assets
|  |- app.json
|  |- package-lock.json
|  |- package.json
|- scripts/
|- apispec.yaml
|- appspec.yml
|- README.md
```

We will be using Git as our Source Control System, managing our directories on GitHub. Above are the details of the directory tree we will be using for the duration of our project, Elysium.

This repository structure is a combination of adapted project initialization code for Django (elysium-backend) and npm React Native (elysium-frontend). Django and React Native are commonly used together for full-stack capabilities and application creation. These base-initialized folders were modified to fit our application according to new screens, components, APIs, etc. It was our attempt to keep the structure of these folders as close to their standards as possible, allowing for more resources for the continuous adaptation and advancement of Elysium.

3 Review and Merge Policy

1. Pre-Commit hooks for linting code
   (a) Flake8 for python (Django)
   (b) ESLint for Javascript (React Native)
2. Merge Policy
   (a) Our merge policy dictates that before a merge is finalized, pull requests, along with passing all unit tests created, must be reviewed and accepted by at least one reviewer other than the author who made the pull request.
3. Branching Policies
   (a) At the lowest level, each issue of our project will have its own feature branch where multiple authors can commit features and changes to the specific issue. These feature branches will be branched off of the development branch.

(b) The development branch sits below the production branch, acting as a working directory where all merges and commits from feature branches will be gathered. It will also serve as an integration point where merge conflicts and the most recent code is located.

(c) The production branch will only be used for production-ready code. Merge requests from the development branch must be accepted by both other persons before the merge is finalized into the Master.