

DK OSRS

A website dedicated to learning and searching about the online
game known as Old School RuneScape

By

Dylan Kelly

Project Overview

The project consists of building and hosting a functional website that will allow the users to browse and learn more about the online game RuneScape. The site will offer in game item searching, in game player searching and in game monster searching. The user will have option to register for the site which will unlock more advance features such as, Item saving, player comparison highlighting, profile section and voting for their favorite RuneScape streamer and youtubers. The site will be full functional frontend and backend with working database.

Project Deliverables

The project will be maintained through GitHub and correspond to the gantt chart listed below. Using git timestamp to measure and record when the task is completed. The repository link can be found here : https://github.com/Dylandk10/cosc_412_solo

Development Approach

This project will be built using agile methodology. Starting first with the planning phase which will consist of completing the following diagrams: gantt chart, HLA (High Level Architecture), WBS, Use-case diagrams. Following completion of the planning phase the next phase will be the implementation and maintenance phase, this is an ongoing life cycle phase. The implementation and maintaining phase will use a spiral approach to the development of the site starting with implementation -> testing -> fixing and when ready deployment -> repeat. This will allow myself to build test and deploy the website on an ongoing basis while maintaining bug handling and added features.

Justification

I selected the agile approach with a spiral iteration because this would allow me to continue updating and fixing new features and bugs as they are discovered through the site. I prefer to use a spiral approach during the development phase so I can keep iterating over a task to make sure I keep up with the demanding change from the website users.

Managing Changes

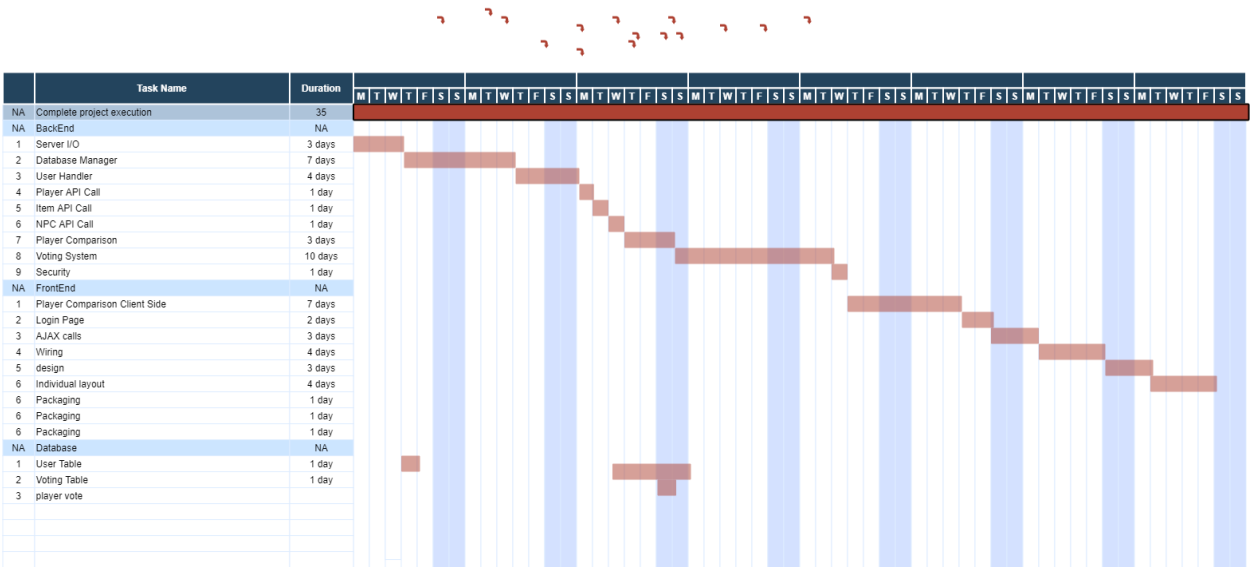
Alongside the implementation and maintenance phase corresponding diagrams will be updated to reflect the ongoing architecture, use-case and class diagrams for how the code is maintained. Diagrams can be found below and on the git repo under diagrams.

Project Support Functions

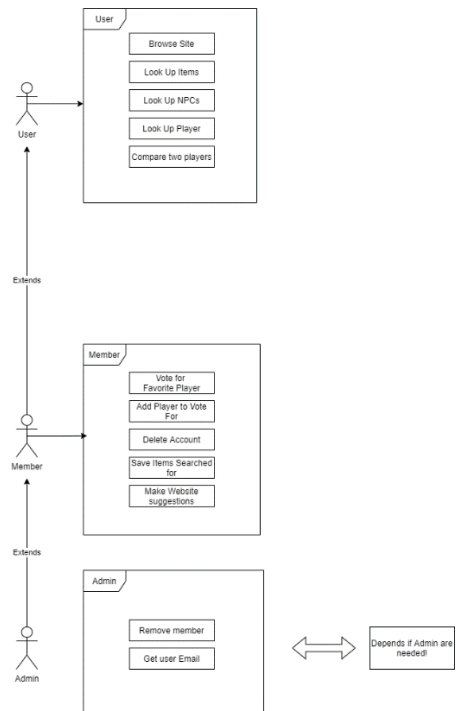
To ensure quality I will have unit testing, use case testing and general testing also known as Integration testing. The configuration will be handled by Heroku and Django platform. These to technologies work well together and having them communicate through GitHub will be no problem. The data is taken from the official OSRS website to ensure quality data.

Figures

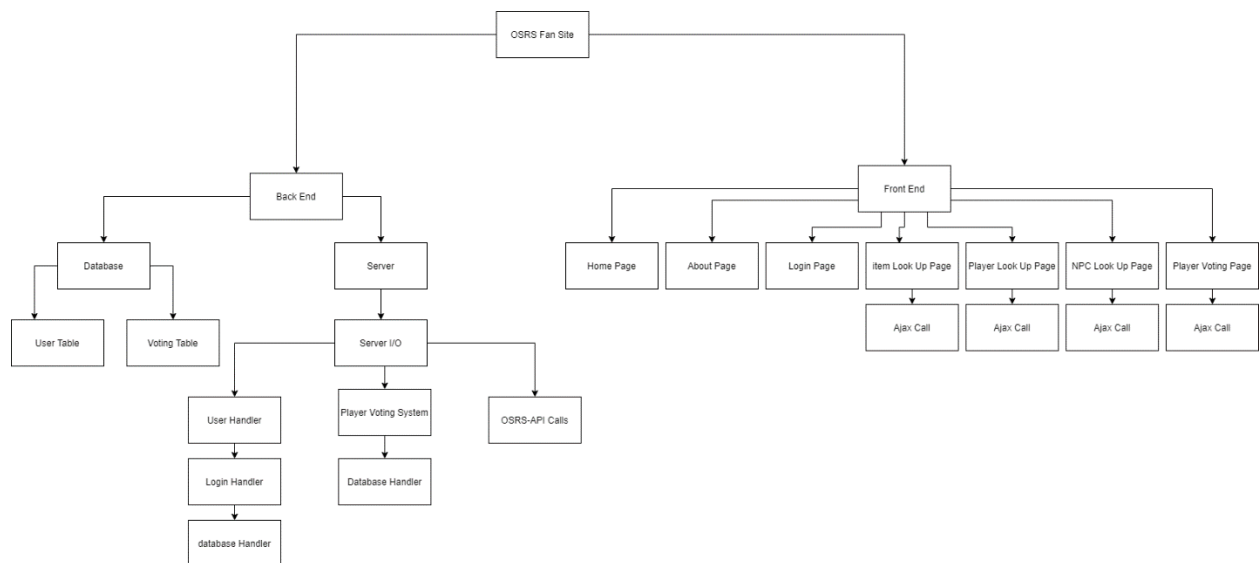
Gantt Chart



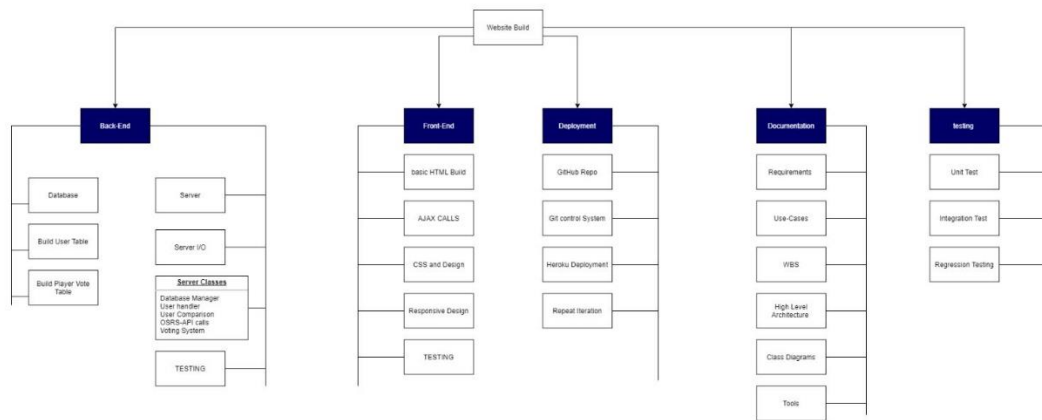
User-use case diagram



HLA



WBS



Inside the folder there are many files for specific use case, api tools, requirements, class-diagrams, and other files related to the project.

Tools:

Back End tools

Django - version - 3.1.7

Python version - 3

SQLite

OSRS-API

Django mitigation

Front-End tools

HTML/CSS

Bootstrap

JavaScript -ES6

jQuery - AJAX

General Tools

Git

GitHub

Heroku

Requirements

General Requirements

Landing page (home page)

login page

log out page

player comparison page

-search a single player

-search two players and compare

-high light stats that are better

Item page- users should be able to search in game items

Favorite player page - players should vote on favorite players/youtubers

About page

Game Break Page

Profile Page

Functional Requirements

User (name, game user name, age, email, password)

OSRS-API

SQLite Database

player lookup and stat highlighting

Vote page that list the favorite OSRS players

Security - secure login and password hashing

player voting system

user sign in

user sign out

user vote for player

user change password

user look up item

user look up NPC

user look up player

little game for players to play (game break)

Non-Functional Requirements

Database scales to new users

Database scales to new players to vote

Voting System is responsive and quick to sort

Users information should be secure

User password is hashed and salted

reCaptcha for non bots

Interface Requirements

Interactive web-interface

Responsive design for mobile devices

user player lookup

user login

about page

fixed navigation menu