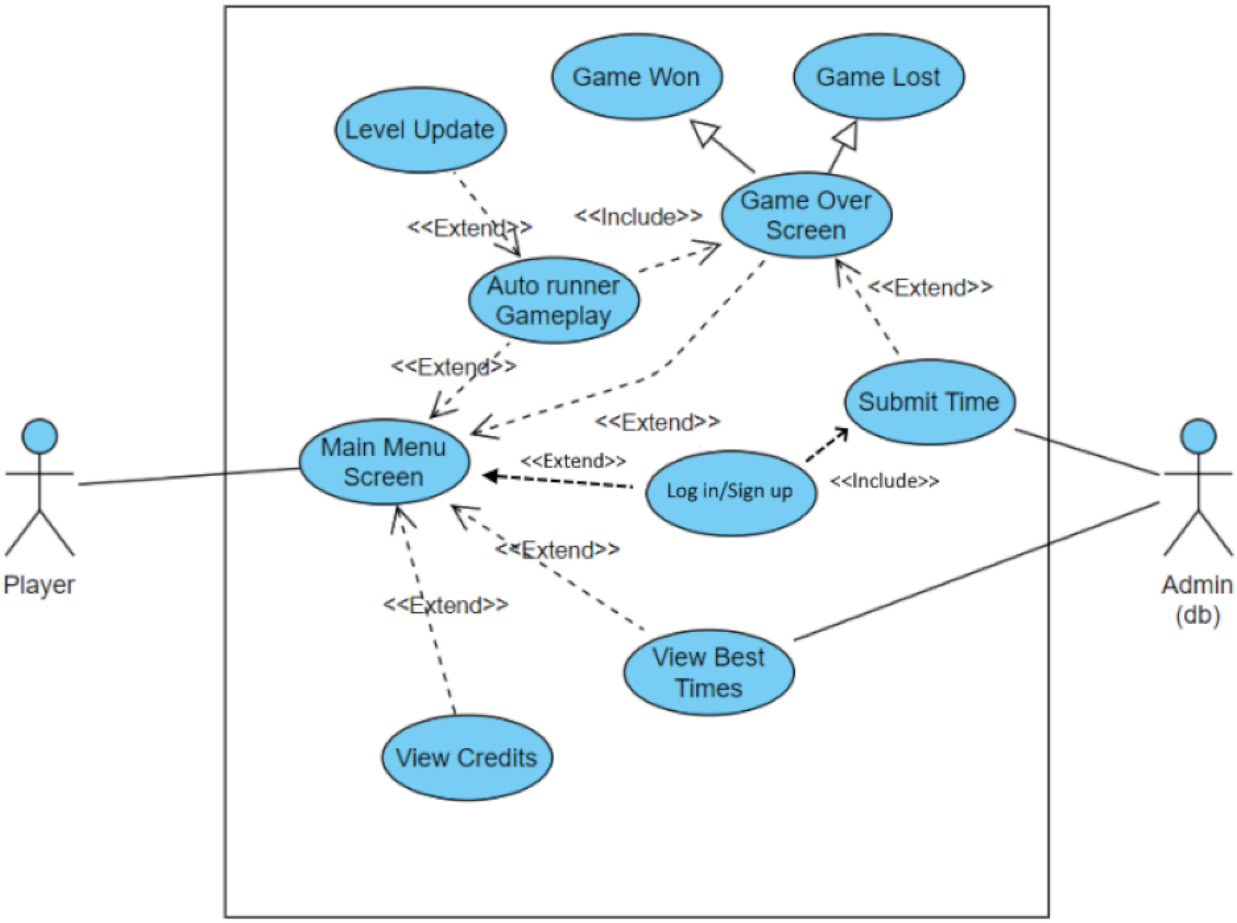
Sprint 1: Time Platformer

Elton Gbollie, Brandon Price, Scott Serafin, Luis Cruz Quispe, Reid Cook

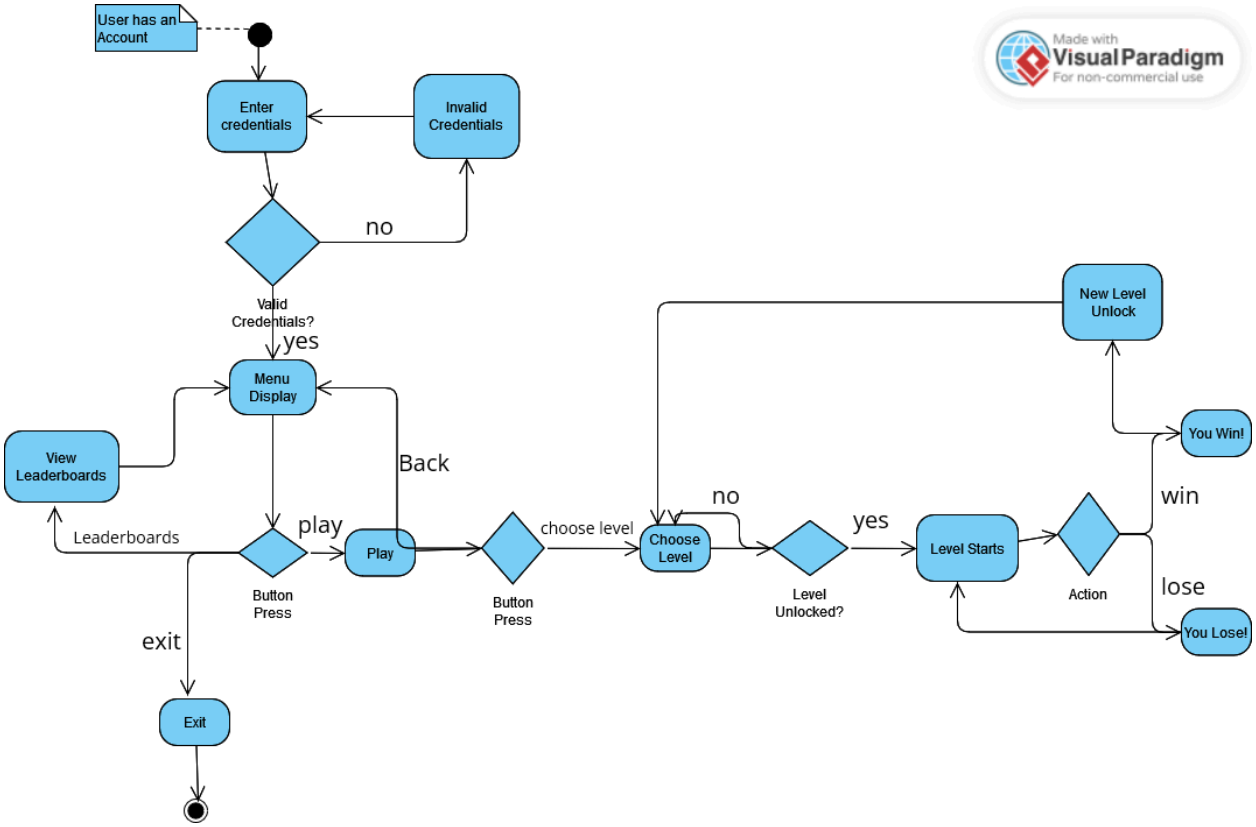
GitHub Repo:
https://github.com/reidcook/cmsc447-sp2024-Ragamuffin

Jira Board:
https://ragamuffin.atlassian.net/jira/software/projects/RG/boards/1

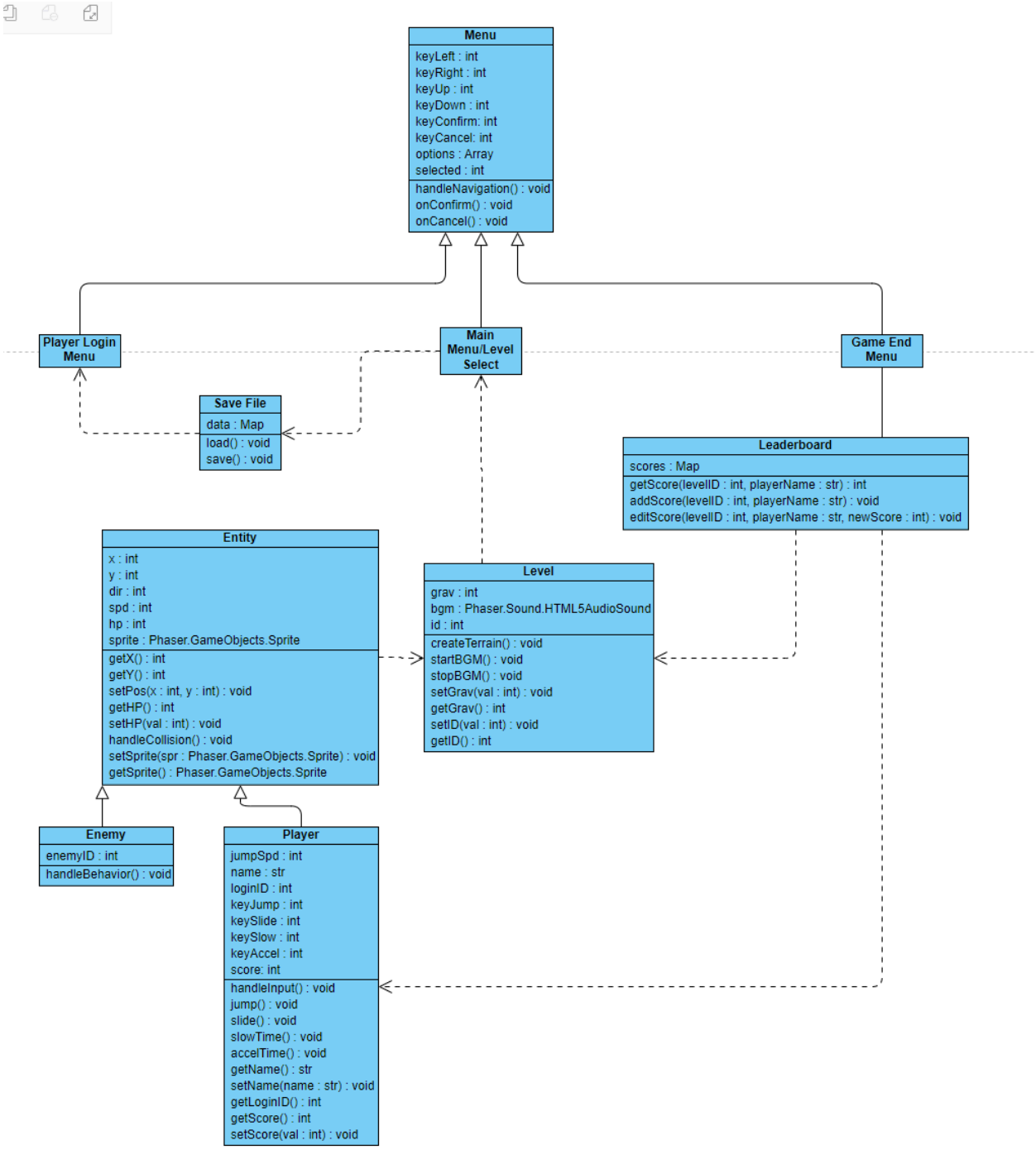
Use Case Diagram:



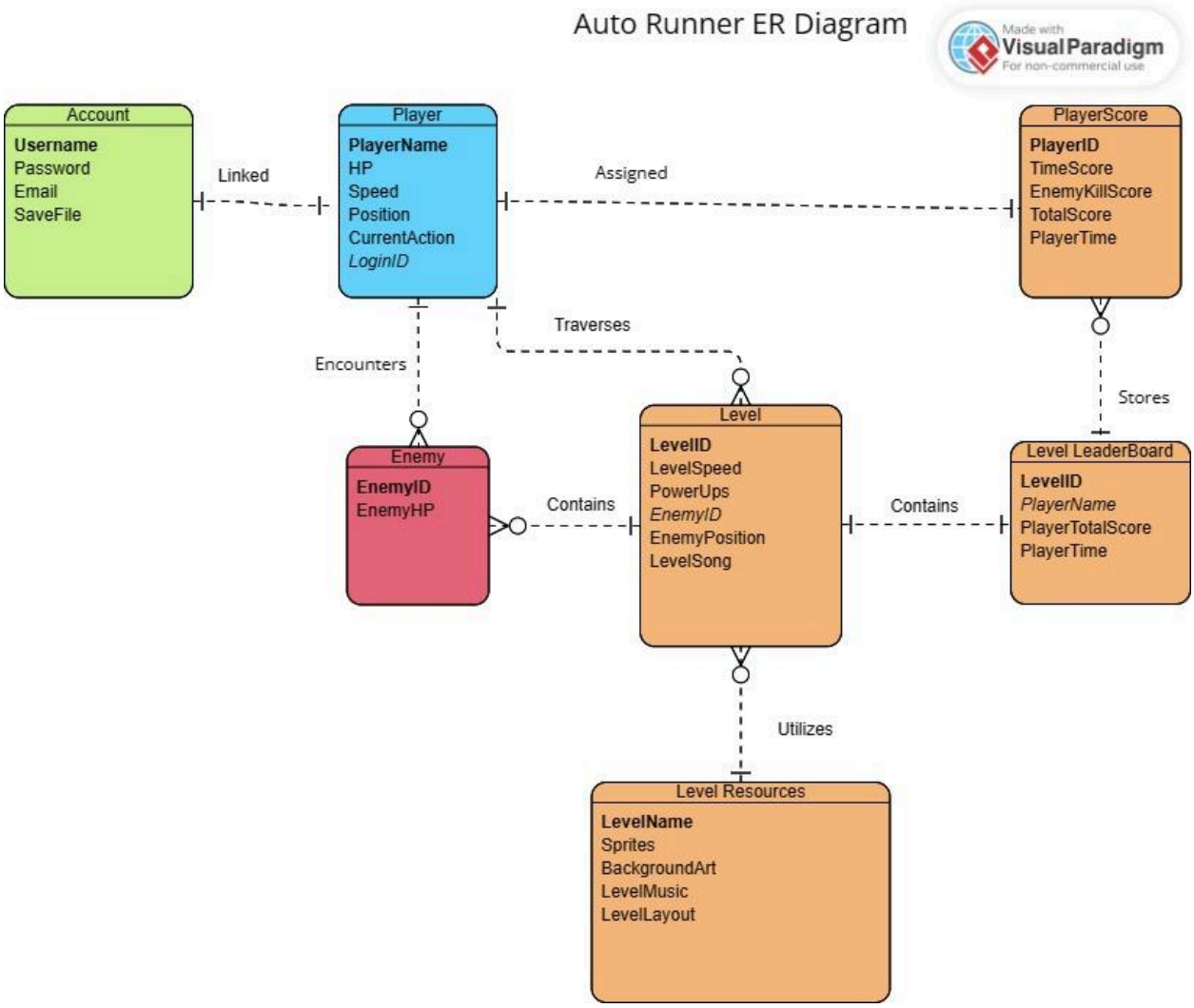
Activity Diagram:



Class Diagram:



ER Diagram:


Auto Runner ER Diagram
Made with Visual Paradigm For non-commercial use
Account
Username
Password
Email
SaveFile
Linked
Player
PlayerName
HP
Speed
Position
CurrentAction
LoginID
Assigned
PlayerScore
PlayerID
TimeScore
EnemyKillScore
TotalScore
PlayerTime
Traverses
Encounters
Stores
Level
LevelID
LevelSpeed
PowerUps
EnemyID
EnemyPosition
LevelSong
Enemy
EnemyID
EnemyHP
Contains
Contains
Level LeaderBoard
LevelID
PlayerName
PlayerTotalScore
PlayerTime
Utilizes
Level Resources
LevelName
Sprites
BackgroundArt
LevelMusic
LevelLayout

Use Case Document:

Main Menu Screen:

The main menu screen shows different options that can be accessed. It is extended directly by auto runner gameplay, view credits, view best time, and game over screen. The only actor involved is the End-User. The preconditions to this screen is that the game is launched, and the user is logged in. The basic flow would be the following: The user launches the game. The user logs in. The main menu is displayed. Then, the user can select a variety of options.

View Credits:

The view credits use case will be a displayable screen for the user to see all the creators of the game and any additional mentions. It's an extension of the main menu screen, so in order to access it, the user must first be at the main menu screen. The only actor involved is the End-User. The preconditions to viewing this screen is to be on the main menu screen. Once on the screen, the user can choose to move back to the main menu.

Auto Runner Gameplay:

The auto runner gameplay use case will be the screen that displays to the user once the game actually starts. In this screen the player will automatically run in one direction, and the user will have multiple inputs they can press to interact with the game. A jump button, as well as several buttons to activate abilities will be implemented. It's an extension of the main menu screen, so in order to access it, the user must first be at the main menu screen. The only actor involved is the End-User. The preconditions to viewing this screen is to be on the main menu screen. Once on the screen, the user can either win, or lose. After winning the game the user will be taken back to the menu.

View Best Score:

The view best score use case will be a displayable screen for the user to see all the submitted scores of people who have played the game. It's an extension of the main menu screen, so in order to access it, the user must first be at the main menu screen. The only actor involved is the End-User. The preconditions to viewing this screen is to be on the main menu screen. Once on the screen, the user can choose to move back to the main menu.

Game Over Screen:

The game over screen use case will be a displayable screen that shows the user sees after they have completed a full game. This will display regardless whether the user has won or lost. It extends the submit time use case, is included by the auto runner gameplay use case and is extended by the main menu screen use case. The actors in this case are the Admin(database) and the End-User. The precondition is that time has been submitted. The flow is that once you see the game-over screen, you go back to the main menu.

Submit Time:

Once the user has completed the game and reached the game over screen, which means to either reach the end of the final level or to have died, they have the choice to submit their time. This functionality will involve both the user and admin, the user will communicate with the admin and will post it to the database. Afterwards it'll show the "View Best Scores" screen with the user's score at the bottom and their rank in comparison to the rest of the scores.

Log In/Sign up:

This screen appears when the user first loads the game. It requires the user to input a username and password to login into their account. If the credentials are correct they login, if not it requires them to try again. There will also be an option to create an account. This functionality will require both the user and administrator, as registering an account is an administrative process. On logging in, you are taken to the main menu screen.

Requirement Analysis Document:

Introduction:

This document outlines the requirements for a PG-13 rated, web-based auto-runner designed for CMSC 447. The game will feature several levels of increasing difficulty, incorporate a database for tracking scores and game states, and utilize both a provided and an application-specific API.

Overall Description:

Game Environment: The game will be accessible via web browsers, ensuring broad compatibility.

Game Levels: Players will progress through many distinct levels, each offering unique challenges and gameplay mechanics.

Features and Functions: Core gameplay involves navigation, strategy-based elements, augmented with two additional, innovative features to enhance player engagement.

Specific Requirements:

User Interfaces: The game will feature an intuitive UI, including start menus, level selection, in-game HUD for score tracking, and a leaderboard.

Hardware Interfaces: Designed for keyboard and mouse input.

Software Interfaces: Utilizes sqlite3 for database management, with flexibility for alternative databases if they provide enhanced functionality. Integration with both a provided API and a custom, application-specific API is required.

Gameplay Requirements: The game will include clear objectives for each level, with gameplay mechanics explained through an initial tutorial or help section.

Performance Requirements: The game will load within a reasonable time frame and maintain responsive controls and interactions throughout.

Security Requirements: User data, including scores and game states, will be securely stored, with measures in place to prevent unauthorized access.

Database Design:
The database will store user information, scores, game states, and leaderboard rankings. It will be designed for efficiency and scalability, ensuring quick access and updates as needed.

Design Constraints:
The game must follow the PG-13 content guidelines, be web-based, and use specific technologies (sqlite3, designated APIs) as outlined in the project specifications.

Software System Attributes:

Reliability: The game will feature robust error handling and uptime.

Usability: Designed with a focus on user experience, ensuring ease of use for the target audience.

Maintainability: Code and resources will be organized in a manner that facilitates updates, maintenance and testing.

Appendices:
Above diagrams provide visual representations of the system's architecture, including use case, activity, class, and ER diagrams, supporting a clear understanding of the project's structure.

Management and Planning:
The project will follow an agile development process, divided into three sprints focusing on design, build, and test/integrate/deploy phases. Project documentation will be managed via JIRA, with code hosted on GitHub.

Submission Requirements:
Each sprint will conclude with a submission of deliverables as specified, culminating in a final product presentation.

This document provides an overview of the game project's requirements, ensuring all team members and stakeholders have a clear understanding of the objectives, design considerations, and technical specifications necessary for successful development and deployment.