Project Proposal

Otito Mbelu (G00397738)

December 7, 2023

Project Proposal

Supervisor: Dr. Damien Costello
Project Name: Biometric Data Analysis in Digital Game Scenario

Project Context

Introduction

PUBG: Battlegrounds (previously known as PlayerUnknownś Backgrounds) is a battle royale style player versus player shooter game developed by PUBG Studio. Players face-off with each other using various types of battlefield weapons in a last man standing deathmatch and the last person to remain alive wins. The game is available in all major platforms and as of March 2021, the mobile version of the game has accumulated more than a billion download outside of China with revenue of over $9billion while the PC and console versions have accumulated a total revenue of $4billion [1].
Since its first release in 2017, the game has since become one the fans favorite and has over '350,000' peak concurrent monthly users [1]. As a multiple award-winning game with proven longevity records and a large community. Interest in the game cut across diferent demography and is equally far-reaching across the globe. The game playing scenario requires players to face-off with other players and there is where some skills like 'eye-hand-coordination', 'ear-hand-codination', 'fine-motor' skills, etc: are required to compete favorably against other players. Players have access to ora varieties of weapons with different capabilities and can make in-game adjustments to their control to suite their various preferences. This project is a continuation of research work previously done by Fourth Year Software Design Students titled 'Biometric Data Collection for Performance Optimization in a Digital Game Scenario' in collaboration with the Department of Sports & Excercise Science, Atlantic Technological University.

Previous Projects

The originiating project titled **'Biometric Data Collection for Performance Optimization in a Digital Game Scenario'**, posed the question

'can a players biometrica data be used to optimise their performance in a first-person shooter game'? And a subsequent follow up project which sought to create a Chart API capable of displaying all relevant information previously displayed on different pages on a single page.

The former research was geared towards creating a test environment where players can practice and horn their skills in a similar scenarios (Weapons, controls, user perspective, ect:) obtainable in PUBG : Battlegrounds in the form of a Unity Desktop `Application`. Collection and storage of Biometric data from an `Activity Monitor` in the form of a `Smart Watch`. With the enventual goal of finding correlation between their performance and their Biometric data.

Project Objective

The objective of this research project will seek to address some of the limitation listed in the previous projects and complete the future development goals of both projects. These includes :

- Chart API Integration: Integrating the developed Chart API into the Test Application.

- Offline Data Storage: Provision will be made for offline temporary file storage to improve the overall reliability of the whole system

- PUBG API: Further research on new developments in the PUBG API for better user experience.

And eventually seek to answer the following research questions :

- Can users' current physical condition as indicated by their Biometric data, have any direct relationship with their performance in such gaming scenario?

- Can Biometric and test data help suggest the most suitable settings for different game scenarios?

Technologies & System Architecture

Legacy Architecture

The overall system architecture is depicted in Figure 1. The system is made up of three modules interacting with an external API and user devices. The modules are listed as follows:

- Test Application A Unity designed desktop application on Windows Platform that where users can play test for different game scenarios.

2

- Node Js Web Application An Express Node JS Web application deployed on Amazon EC2 Virtual machine, serving as a callback endpoint for OAuth 2.0 Authentication for the Polar Flow API.

- Firestore Data Storage Permanent storage medium for users' test and Biometric data.
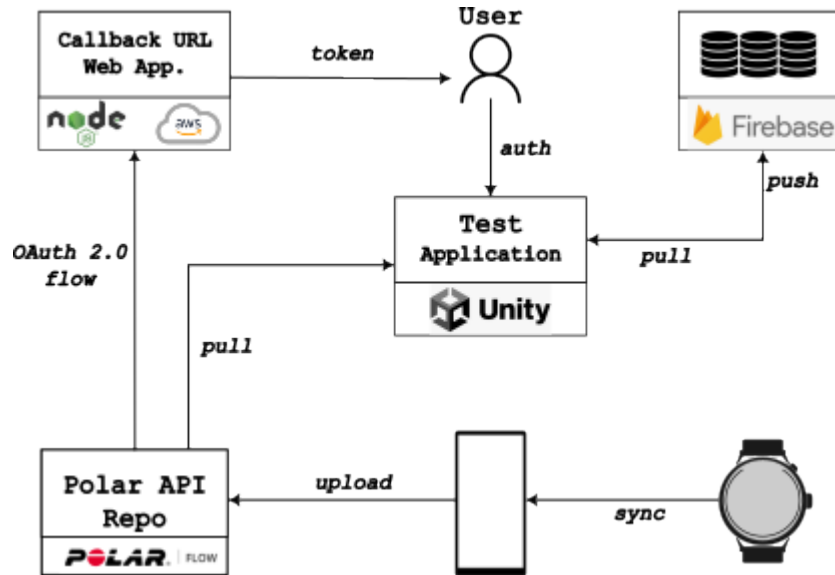
User Devices:

- Smart Watch

- Smart Phone

Figure 1: System Architecture

The actions in the Figure 1 can be summarized as follows:

- sync: passing of biometric data to a smart phone from a Polar watch.

- upload: uploading of biometric data to Polar API repository.

- OAuth 2.0 flow: Multi step OAuth 2.0 authentication protocol.

- token: getting verification token from the Callback URL by the user.

- pull (Polar API - Test Application): retrieving biometric data from Polar API by the Test Application.

- auth: user supplies token to the Test Application as final step of authentication.

- pull (Firebase - Test Application): retrieving user test/biometric data for rendering by Test Application.

- push: saving users' test/biometric data to a Datastore.

Proposed Architecture

The final system architecture will include new modules to facilitate data analytics using A.I models. The new system calls for a relational database replica of the Firebase storage to ensure data consistency, predictability and structures suitable for relevant statistical analysis. A.I models as a mini service that perform operations with supplied data and returns result of analysis. A Web Application that functions as a Controller that interfaces the data and A.I modules, Executive Dashboard that renders user data, results from the A.I Modules and any other relevant information about the system.

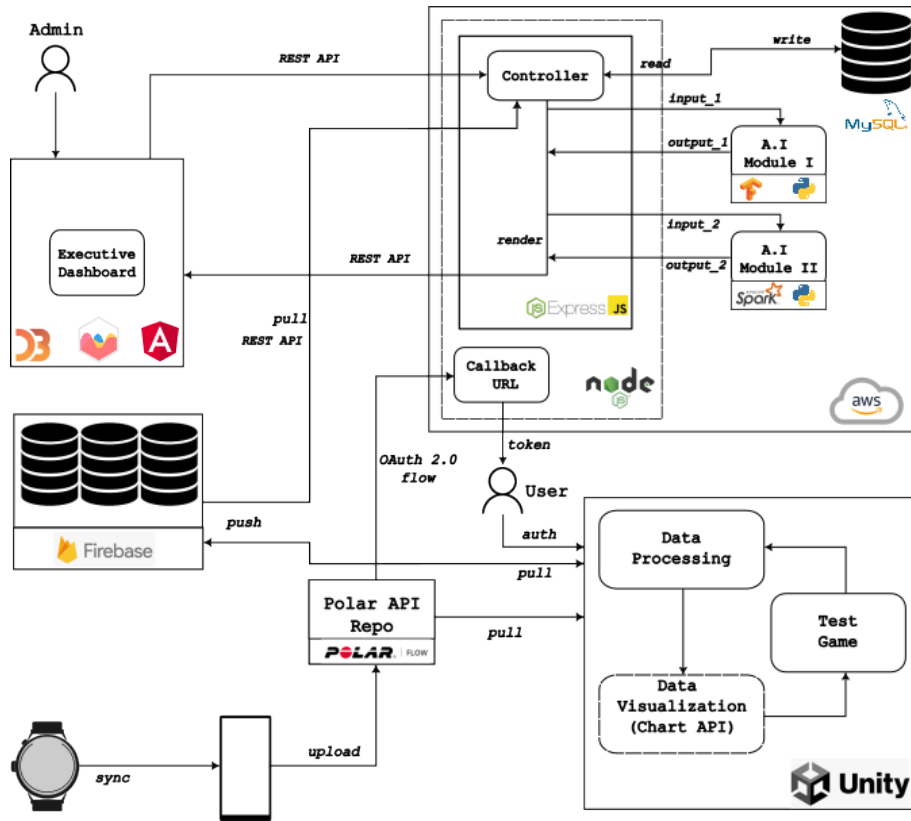Figure 2 shows the proposed architecture for the System. From the diagram

Figure 2: Proposed Architecture

above, additional modules have been added to the system which are: the back-

end 'Controller', front-end 'Executive Dashboard', a 'Relational Database', and A.I Modules. The functions of these modules are explained below:

- Controller: This module is responsible for coordinating data fetch from the Relational Database and making sure that the Relational Database is always in sync with the Firebase Datastore. It is also responsible for filtering and structuring data being fed into the A.I. Modules and when necessary, using the output for an A.I module as input to another A.I module.

- Executive Dashboard: This module provides a Graphical User Interface that displays the working dataset, and offers interfaces to filter, tune and if necessary, bias data being sent to the A.I Modules for analysis. This module will be developed using the Angular Framework with integration of D3.js or Chart.js for displaying relevant charts. The choice for Angular Framework over other solutions namely ReactJS and Vue is because of the more structured nature of the Angular Framework which is more suited for collaborative development.

- A.I Modules: These modules are responsible for running analysis on data. The modules accepts input from the 'Controller' and outputs a comprehensive summary of the results of the analysis which is displayed on the 'Executive Dashboard' Module. For this purpose Tensorflow & Apache Spark were chosen for the following reasons:

  - Capabilities: Apache Spark is capable of performing linear and logical regression which are the common tools used for this class of problem. Tensorflow is also capable of performing the afore-mentioned techniques and can perform regression using neural network models.

  - Accessibility: The both tools are open-source and available to the general public.

  - Python API interface: The both tools provide a Python API. This provide ease of deployment as Python is platform independent and relatively easy to setup in an Ubuntu Virtual machine.

  - Large Community: Both have large community of contributors. Documentation and tutorials are readily available on the internet.

- Relational Database: Data for the analysis will be sourced directly from this Database which is a structured replica of the Firestore Database as obtained from the Legacy System. The rationale for this Module is to provide a flexible means of retrieving data in various forms and shape as may be needed for analysis and also overcome the high complexities involved in Firestore for compound queries. The MySQL Database was chosen for this purpose for the following reasons:

  - Available: MySQL is open-source and free, thus suitable for this project.

– Secured: Offers security with its User account management system.

– Durability: MySQL API is available for most programming language and widely used.

- REST APIs: Communication Interface between various components of the system. The JSON format which is considered standard will be used for this purpose.

- Chart API: is responsible for displaying users' test results and biometric data to the users. This already developed module will be integrated into the 'Test Application' and undergo some modification to be able to display custom data as requirements change.

Schedule of work

I will be solely responsible for the design and implementation of the Controller, The Relational Database and Tensorflow A.I Module. In collaboration, design the REST APIs between the Controller, Executive Dashboard, and the A.I Modules. Adaptation and Integration of the Chart API into the Test Application.
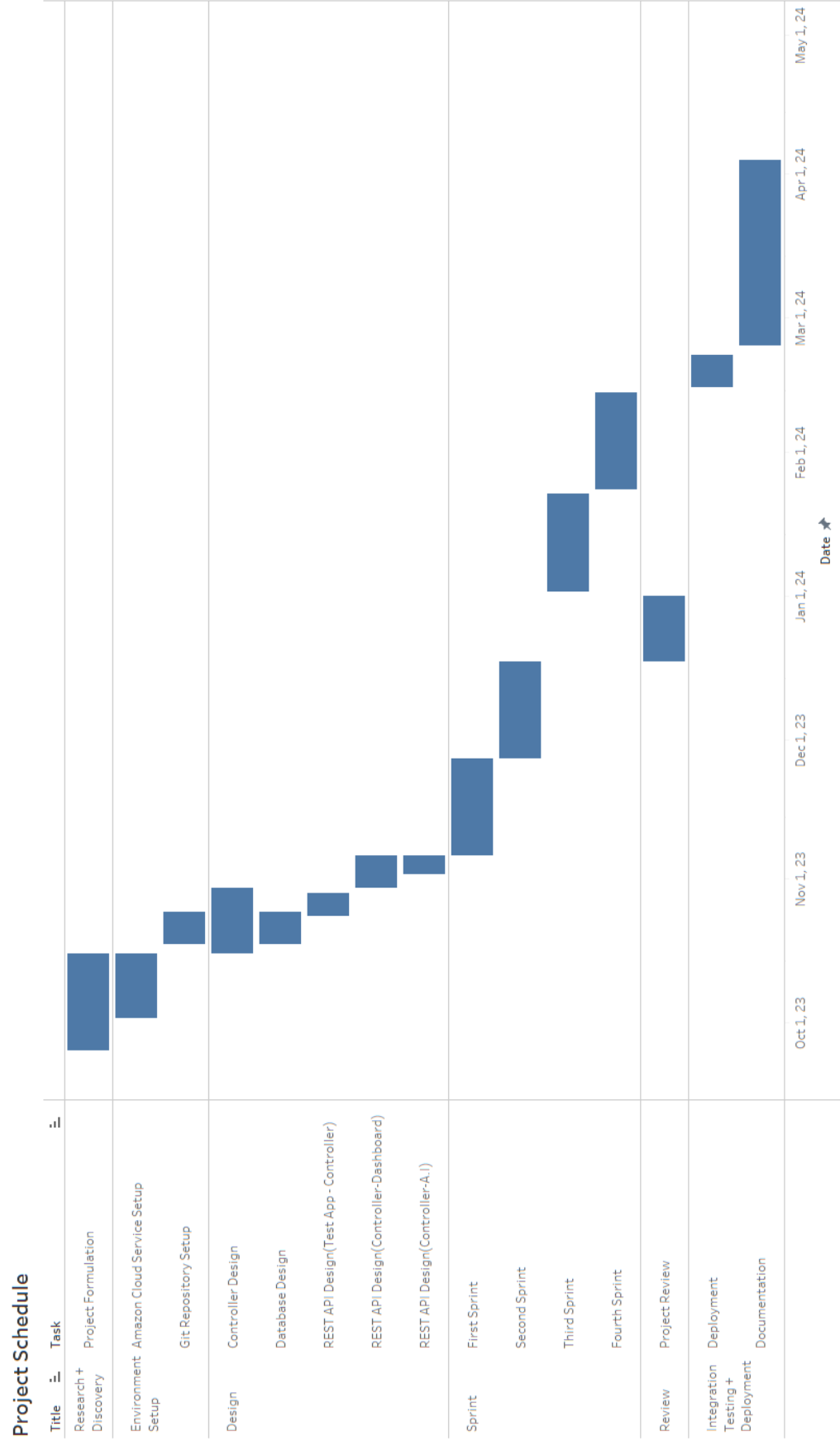
Figure 3: Project Schedule

Bibliography

[1] J. Clement. *Monthly peak concurrent players of PUBG: Battlegrounds on Steam.* https://www.statista.com/statistics/755111/pubg-number-players/. Accessed on October 3, 2023. 2023.