Functional Specification

CA400 Year 4 Project

Members: Immanuel Idelegbagbon (17393433) & Stefano Puzzuoli (17744421)

Project Title: No-Limit Texas Hold'em Poker AI

Date Finished: 03/12/2020

Table of Contents

1. Introduction

1.1. Purpose

The following functional specification is designed to give an extensive overview of the proposed No-Limit Texas Hold'em Poker AI web application. It consists of a listing and the corresponding descriptions of the core requirements and functionalities that the application will necessitate. Regarding features for which possible, high-level descriptions of how they will be designed, managed and implemented will be provided. This document permits a rigorous assessment of the requirements before the designing/implementation of the system begins, in order to reduce later redesigning further on in the software development life cycle. Accessibility to this documentation will be limited to the software product developers, the project coordinator, the project supervisor, and any other member of Dublin City University staff that has an involvement in the CA400 module.

1.2. Scope

The product to be discussed is a No-Limit Texas Hold'em Poker AI application. It consists of a web application that allows users to test their Texas Hold'em Poker skills against one or more Artificial Intelligence Agents in a game of No-Limit Texas Hold'em Poker.

Users will be able to select from 4 different levels of difficulty and train and assess their skills according to their desired challenge magnitude.

The user interface will consist of a user-friendly web UI which users will be able to access through their web browsers. This UI will allow the users of the application to visualise the poker game, with the table, opponents, cards and all the possible accepted commands when playing each hand.

The application will also include a sign up/login functionality which will allow users to keep track of their game statistics (e.g. winning percentage). Users which do not want to avail of this functionality will have the option to play games as guests, which will not involve any statistics recording.

Since with No-Limit Texas Hold’em Poker there are 10^{71} possible game states, the web application will be making use of a personally implemented poker hand evaluation library, which we will have to make lightweight and fast. This evaluation library will handle 5, 6 and 7 card hand lookups and all lookups will be done with bit arithmetic and dictionary accesses.

The goal is to implement a Reinforcement Machine learning algorithm that will allow each Artificial Intelligence Agent to try to predict the maximum expected return value on each different game state throughout a game. These AI Agents will be trained

repeatedly with the final objective of generating interesting and viable poker winning strategies, allowing users to test and improve their Texas Hold'em Poker skills.

1.3. Definitions, Acronyms, and Abbreviations

AI: Artificial intelligence.
API: Application programming interface.
Client: A third-party web application used to play Texas Hold'em Poker.
GUI: Graphical User interface.
Machine Learning: Application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
No-Limit: Rule in Poker that allows players to bet an unlimited amount of money when the action is on them.
Reinforcement Learning: Area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize the notion of cumulative reward.
SQL: Structured Query Language.
Texas Hold'em Poker: One of the most popular variants of the card game of Poker. Two cards, known as hole cards, are dealt face down to each player, and then five community cards are dealt face up in three stages. Each player seeks the best five card poker hand from any combination of the seven cards; the five community cards and their two hole cards.
UI: User interface.
UX: User experience.

1.4. References

- *Micheal Richardson, Software Requirements Specification Guidelines*, 2017 (http://www.michael-richardson.com/processes/rup_for_sqa/core.base_rup/guidances/guidelines/software_requirements_specification_91456CD0.html)

- *Rohan Wadiwala, Cloud Database – DBaaS (Database as a Service), 2019* (https://labs.sogeti.com/cloud-database-dbaas-database-as-a-service/)

- *Błażej Osiński and Konrad Budek, The Complete Guide to Reinforcement Machine Learning, 2018* (https://theappsolutions.com/blog/development/supervised-machine-learning/#contents_3)

- *Danny Denenberg, Guide to Password Hashing: How to Keep your Database Safe, 2019*
(https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide)

1.5. Document Outline

The rest of the document is laid out as follows:

Section 2 comprises an overall description of the product. This section delves into the perspective of the product and briefly explains its functions, and additionally, a detailed outline of the user class and characteristics can also be found in this section. Furthermore, this section explains the operating environment of the product and any constraints that need to be taken into consideration during the design and implementation of the system.

Section 3 contains system requirements that have been set for the product. It also contains a detailed and thorough explanation of the product's functions.

Section 4 revolves around the system architecture of the product. An architecture diagram is used to represent the structure of the system and is explained in detail here.

Section 5 contains a high-level design of the system. It includes a model that shows the data flow within the system and the relationships between system components.

Section 6 contains additional information on the software, hardware, and wetware requirements and a detailed preliminary schedule.

Section 7 contains supporting information which presents and explains the Texas Hold'Em Poker Ruleset allowing readers to understand the rules of the game we are implementing.

2. General Description

2.1 Product Functions

The general functionality of this product is to give users the ability to play No-Limit Texas Hold'em Poker against a number of Artificial Intelligence Agents. Users can use this product to hone their Poker skills or to enjoy a casual game of Poker. The AI Agents, that users will be competing against, will learn from being trained on a large number of games and change their play style by adapting to how the user plays. The AI Agents will be trained using Reinforcement Machine Learning to achieve this. Users will also be given the option to choose the desired difficulty level of the AI Agents, they will play against. Other functionalities provided by the product will be the function to create a user account that will keep a record of game statistics. Users who avail of this functionality will be able to view their all game statistics (e.g. winning percentages for each difficulty level). The function to play as a guest user will also be provided for users who wish to not have their statistics recorded.

2.2 User Classes and Characteristics

The purpose of this product is to help users hone their Poker skills via playing against AI Agents. The product is not intended to teach users how to play Poker. It is expected that users will have a basic knowledge of the No-Limit Texas Hold'em rules at the very least. Therefore the product is intended for Poker players of all levels and for those who have an interest in the game.

The product will be presented as an application that can be accessed by means of a web browser. Users will be expected to have hands-on experience and a familiarity with computers and accessing the web. However, the simple user-friendly user interface will make the app easily accessible by users with little or no hands-on experience. Any disabilities will not inhibit or affect the type of user that can use the app by virtue of the app's accessibility.

2.3 Operational Scenarios

User Sign up/Log in
A user will be presented with a menu upon loading the web app which will prompt the user to either sign up, log in or continue as a guest. Choosing sign up or log in will present the user with a create an account form or log in form respectively. Choosing continue as a guest will enable the user to continue to the game menu by logging them in as a guest.

Game Menu
After logging in users will be presented with a menu. This menu will give users the option to start a game and also to select the difficulty level of the AI Agents. After selecting their desired difficulty level, users can then choose the start a game by clicking the join game button. On this menu, users who chose to sign up and then log in will also be able to view their game statistics.

User Playing
Once a user clicks the join game button, the game will be initialised. When it is the user's turn, they will be prompted to select actions such as call/check, raise and fold. These actions will be presented to the user as buttons.

Call/Check
This action will enable the user to call or check depending if any bets have been made in the current round.

Raise
This action enables the user to make bets when it is their turn in a round. When this action button is clicked, a textbox will be displayed and the user will be promoted to enter the amount they would like to raise by.

Fold
This action will enable the user to fold their hand.

2.4 Constraints

2.4.1 Time Constraint

As this product is under a strict deadline set by DCU, making sure that all requirements are met and that the product is fully functionally within the given timeframe is an important constraint to be wary of.

2.4.2 AI Agents Constraint

The AI Agents are a main component of the product. Making sure they work efficiently and play in tune to their set difficulty can be considered a constraint. The efficiency of these agents will be down to our ability to create a sufficient algorithm.

2.4.3 Speed Constraint

It is important that our AI Agents are fast and able to respond to user inputs and game changes promptly. This will keep the flow of the game running smoothly.

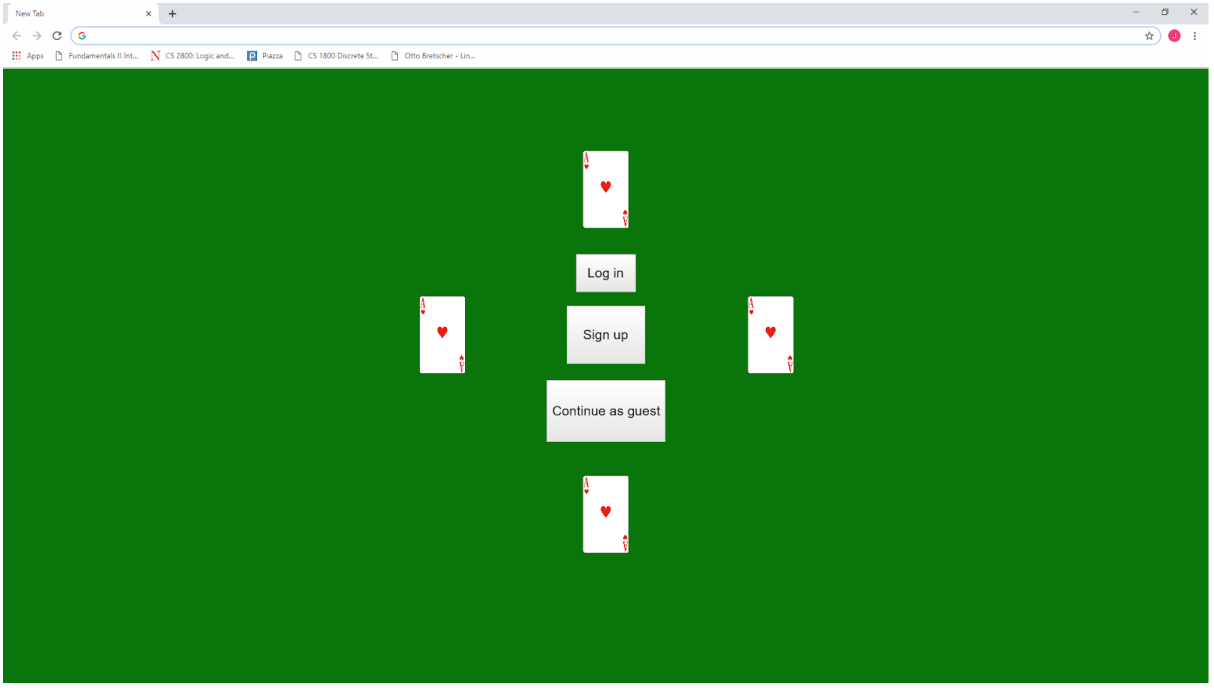
3. System Requirements

3.1. External Requirements

3.1.1. User Interfaces

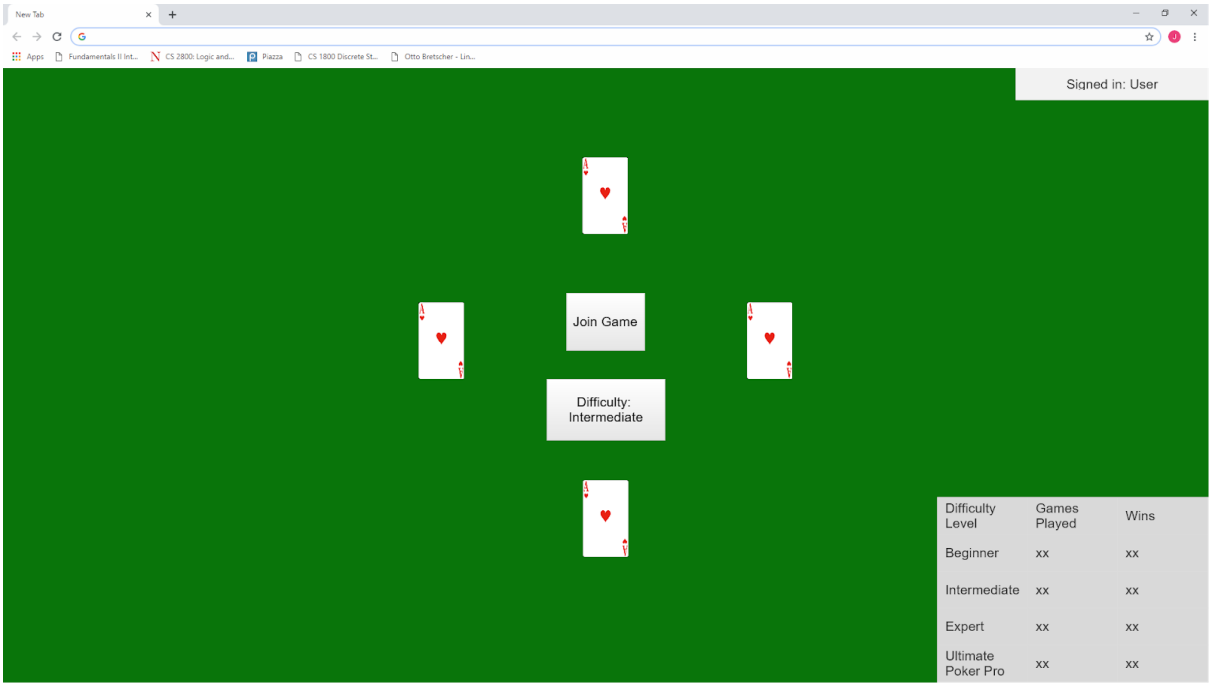
The user interface will consist of a simple user-friendly design throughout the web application. The application will consist of three main webpages. These will be the log-in, game menu and game pages.

Log-In Page



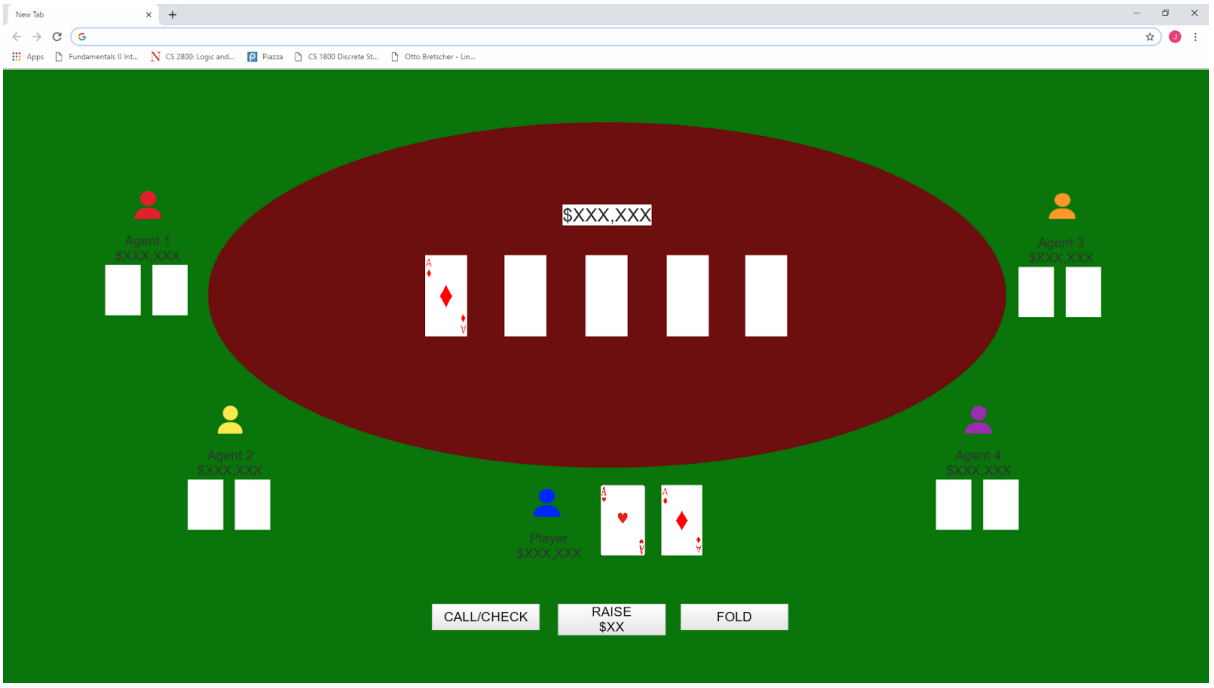
Above is a mock-up of the log-in page. This is the first page users will see when they load the web application on their browsers. This page gives the users the option to log into an existing account, sign up and create a new account or continue to the game as a guest user.

Game Menu Page



Next is a mock-up of the game menu page. This page loads after a user has successfully logged in or continued as a guest. On this page, users can change the difficulty of the AI Agents before starting the game. A user can also view their statistics on this page. Once ready, the user can start a game by clicking on the join game button.

Game Table Page



Finally is a mock-up of the game in play. In the centre of the screen, will be the poker table. On the table, users can see the community cards that have been revealed that round and the amount that is in the pot. A user can see their cards in hand, player icon and name positioned centrally underneath the table. At the bottom of the screen are actions that the user can avail of when it is their turn. These options are dimmed out when it is not the users turn and become solid colours when it is the users turn.

3.1.2. Hardware Interfaces

There will be no mandatory hardware interfaces required to run/interact with the application.

3.1.3. Software Interfaces

Software	Description
Web Browser: Any modern Web Browser	The execution of the application for a user shall be conducted on a modern web browser. Any modern web browser on any operating system shall be able to access and use the application.
Web Server: Apache Web Server	An Apache Web Server will be utilized to deploy our web application into production. The services offered by this server environment are satisfactory for our needs.

Cloud Database: Google Cloud Platform - Cloud SQL	The cloud database will be the component of the application in which the users' credentials and game statistics will be stored. Google Cloud Platform, through its Cloud SQL service, will provide an adequate solution to store our users' details, perform authentication and retrieve game statistics when necessary.

3.1.4. Communication Interfaces

There will be no communication interfaces as all the information that the users will need will be presented in the frontend of the web application.

3.2. System Functions

3.2.1. Poker AI Agent Performance

3.2.1.1. Description

This is one of the key factors around which the entire web application is founded. This functionality is what will allow the users to train and test their Texas Hold'em Poker skills. The Artificial Intelligence Agents will learn from previously simulated games and implement strategies to counteract their current opponents' actions in order to maximise their own winnings. A machine learning Gradient Boosting Regressor will be used to train them repeatedly with the final objective of generating interesting and viable poker winning strategies.

3.2.1.2. Criticality

The AI Agents playing functionality is essential to the application as it is a key function of the app. Without successfully trained Agents, they would not be able to simulate real players and real poker player strategies and this would limit users when training and testing their poker skills with our application.

3.2.1.3. Technical Issues

One of the main difficulties when creating skilled Artificial Intelligence Agents can be successfully training them to reach the desired level. No-Limit Texas Hold'em Poker contains an enormous strategy space, imperfect information and stochastic events, all elements that characterize most of the highest level challenging problems in multi-agent artificial intelligence systems. With 10^{71} possible game

states, creating fast, efficient and winning AI Poker Agents will be a significant challenge.

3.2.1.4. Dependencies with other requirements

This functionality is very tightly connected to the difficulty selection option that users can avail of before starting a game. Since the difficulty selected by a user infers how much the Artificial Intelligence Agents will need to be trained, these two functionalities are strongly dependent on each other. Additionally, in a less direct way, the performance of the Agents will also influence the game statistics recording of each player functionality.

3.2.2. No-Limit Texas Hold'em Poker Game Implementation

3.2.2.1. Description

Another key functionality of our web application is the No-Limit Texas Hold'em Poker game implementation which will actually allow users to play and compete against the AI Agents. This is the component which will allow games to go ahead, controlling and tracking all the events that occur at the table. Who's turn it is, the amount of virtual chips that each player has, when a player is eliminated from the table, dealing the cards, what hand wins and much more are only some of the aspects that the Poker game implementation will have to manage.

3.2.2.2. Criticality

As a key element of the entire web application, the implementation of the Poker game will be quite critical for the correct functionality of the app. All of the elements mentioned in the description and on top of them, many more, are the responsibilities that the Poker game implementation will handle, assuring that each of these are correctly completed and managed. Any small issue in any of these factors could significantly negatively impact the user experience, so guaranteeing that all is executed correctly will be imperative.

3.2.2.3. Technical Issues

Since a No-Limit Texas Hold'em Poker game involves so many different rules, assuring that everything runs exactly as it should will certainly prove quite difficult. Where and how the game information will be stored, when the community cards should be dealt, what hands need to be compared when deciding the winner and much more are all elements that need to be handled carefully and detailedly in order to provide a successful Poker game implementation.

3.2.2.4. Dependencies with other requirements

Being the main functionality of the web application, the Poker game implementation will undoubtedly be interconnected with all the other functionalities of the app.

3.2.3. Web Application User Interface

3.2.3.1. Description

The User Interface of the web application will be another critical functionality without which users would not be able to interact with our Poker application . This will be the point of contact between the users and the app and it will be key to allow users to sign up, login, select options for a new game, visualise their statistics and finally play against the AI Agents. It will be of significant importance that the UI will be easy to use and straightforward. During a game, users should effortlessly know at what point of the game they are, who's playing next, when it is their turn and much more. Angular will be the framework used to develop the UI of the web application.

3.2.3.2. Criticality

As mentioned, the User Interface is a fundamental component of the web application as it will be what the users see and what they use to interact with the app, making it the key element through which they can train and test their Poker skills. Without the UI, all the functionalities would not be able to be displayed, users would not be able to sign up/login, view their statistics, join games and consequently improve their skills.

3.2.3.3. Technical Issues

Since the User Interface is such an important part of our web application and it is the main and only component through which users can access all the provided functionalities, there is lots to implement from this point of view. A user sign up screen, login screen, menu, statistics and game screen will be required and since the feeling of the UI is a crucial part of any web application, it will be necessary to ensure that it is as smooth and easy to use as possible. Additionally, another imperative requirement will be that the UI is accessible in order to guarantee that it can be accessed and utilised without difficulties from users with different disabilities and impairments too.

3.2.3.4. Dependencies with other requirements

As this will be the only point of contact/interaction between the users and the web application, all other functionalities strongly depend on the UI. The sign up/login feature, the viewing of the player statistics, the set up of a game and finally the playing of the game all require the UI in order for them to be performed in an easy to use/user-friendly manner.

3.2.4. Sign up / Login

3.2.4.1. Description

A user will have the possibility to create an account for the No-Limit Texas Hold'em Poker web application via a sign up functionality. The user will be required to enter a username and a password to avail of this service and this information will be stored in a cloud database to be retrieved when the user tries to login with the same credentials, allowing to keep track of the player performances and statistics. Alternatively, if a user of the web app does not want to record their results, they can simply play games as a guest without creating an account.

3.2.4.2. Criticality

The sign up/login functionality is quite important for the purpose of the application as it allows to keep track of each player's performance which can be helpful when users are training and testing their Poker skills as they can visualise their own statistics to know how well they have been doing. This being said, the functionality is not essential as users that do not create an account can equally train and push their skills to the limit with the AI Poker Agents, with the limitation that their past results will not be recorded, if not done by themselves in some other manner not connected to the application.

3.2.4.3. Technical Issues

This feature can involve some difficulties and mainly security challenges. As we will be storing user credentials, consisting of a username and a password, it will be essential to securely store and protect this information from external third parties. For this purpose, we decided on a cloud database platform that guarantees the level of security needed. Additionally, another concern regarding this functionality is enforcing that all the usernames associated to our players' accounts are unique, meaning that before accepting the creation of a new account, it will be necessary to verify that the entered username does not already exist.

3.2.4.4. Dependencies with other requirements

As is clear, our sign up and login functionalities are strongly interconnected as a login cannot occur successfully if an account with the specified username and password was not created prior to the login attempt. Aside from depending on each other, these functionalities also depend on the web application User Interface, as in order to perform sign up and login operations, users will require that certain visuals are presented to them when accessing these functions in our app.

3.2.5. Player Statistics Recording

3.2.5.1. Description

An additional functionality that our web application will include is the possibility to record player statistics, allowing each user to keep track of their performances, by knowing their win percentages at each different level of difficulty. To avail of such a functionality users will be required to sign up and create accounts with a username and password which are credentials they will then have to use to log in to allow this performance tracking functionality. Alternatively, if users do not want their results saved, they can simply play games as guests which will not impact their record.

3.2.5.2. Criticality

This feature will not represent a functionality without which the application cannot succeed. Users would still be able to play and improve their Poker skills without their performances being recorded, although, as the main purpose of the web application is to allow users to improve by training and testing their Poker skills, we believe that allowing users to keep track of their of wins, will push them to perform better and aid them in knowing how much they are improving with time.

3.2.5.3. Technical Issues

In order to store and retrieve results and performances from a cloud database, we will need to correctly purchase and set up a cloud database service. Issues that could be encountered when data is being transitioned from one location to another are potential bottlenecks in the cloud database server and upload/update intensive errors.

3.2.5.4. Dependencies with other requirements

Since the player statistics recording cannot be performed without a user signing up and logging in to the web application, this latter functionality is essential for the statistics feature. Additionally, as the web application will keep track of winning percentages for each level of difficulty, the game level function is also interconnected with this functionality. Furthermore, the UI will be necessary to present these statistics in a user friendly manner so these will also be linked.

3.2.6. AI/Game Difficulty Level Selection

3.2.6.1. Description

Additionally to being able to play against AI Agents trained using Reinforcement Machine Learning, users will also be provided with the option to select the level of difficulty at which such Agents perform. The AI Agents will differ in skill due to the different levels of training that will be provided for them depending on the difficulty selected by the user. At higher levels, users will face Agents which have been trained much more intensely compared to Agents that users will face when selecting lower levels. The levels to select from, from easiest to hardest, are: Beginner, Intermediate, Expert, Ultimate Poker Star.

3.2.6.2. Criticality

The varying difficulty levels at which the AI Agents can perform, although not an indispensable feature, are certainly an important factor for the purpose of our web application. As it would have been possible to provide only one skill level of AI Agent, this would have limited the audience that can be reached with such an application. With the differing levels, this No-Limit Texas Hold'em Poker game will be able to attract a large variety of Poker players of all different levels, from novice players which simply want to learn the rules and get into the game, to expert Poker players which want to push their skills to the limit; the web application will have a bit for everyone that loves the game of Poker.

3.2.6.3. Technical Issues

Regarding the challenges to implement the differing levels for our AI Poker Agents, not many should be encountered as we will be using the same Reinforcement Machine Learning algorithm to train the different level Agents, but the factor making a difference in the level of skill of the Agents will be the number of simulated games that they will be trained against. This means that the same algorithm will train less

skilled Agents and more skilled Agents, but with the latter being trained on a significant amount more of simulations.

3.2.6.4. Dependencies with other requirements

The AI/Game Difficulty level selection will certainly be linked to the AI performance functionality, as both of these are strongly dependent on the performance of the Reinforcement Machine Learning algorithm. The difficulty selection will also be connected to the player statistics recording as player winning percentages will be stored for each different level of the game. Finally, this functionality is also dependent on the UI as this will be the only interface through which users will be able to select the difficulty of the game they will be playing.

3.3. Non-Functional Requirements

3.3.1. Accuracy Requirements

As the entire application depends on the accuracy with which our Artificial Intelligence Agents are able to implement Poker strategies in order to maximise their own winnings, it is clear how training and enabling said Agents will significantly impact the overall outcome of the final product. Failing to do so would undoubtedly defeat the purpose of the web application which is to allow players to train and test their Poker skills, as without competent AI Agents, this would not be possible for our users. For this reason, implementing an accurate and precise algorithm will be one of our leading concerns.

The first aspect to focus on is the data being used for our Reinforcement Machine Learning algorithm. Regarding this, we will be making use of a publicly available and very well documented Poker Hand Dataset provided at the UCI Machine Learning Repository. This dataset contains 1.025.010 Poker hand instances, features 11 attributes and has no missing values, so it is basically perfect for our needs. Additionally to this, we will also be making use of a pre-existing Python synthetic Poker dataset. When implementing our algorithm, we will carefully analyze how our model performs and select which dataset to focus on, if not both, to obtain the most accurate and viable AI Poker Agents.
If when initially creating our model, a certain level of accuracy is not satisfied, some algorithm tuning will be applied, by either interchanging the data being used to train and test the model, or by manually tweaking some aspects of the model too.

3.3.2. Performance Requirements

One of the most important aspects to consider when developing the No-Limit Texas Hold'em Poker web application is the performance. As the system will be managing such large quantities of data, assuring the computations and transactions are performed efficiently and in non-critical moments will be essential to guarantee a quality user experience.

Although the large quantity of training required by the AI Agents could have caused some significant overhead, this will not be an issue as this will occur one single time per each level of difficulty before the app is available to the users.

An issue that cannot be overlooked, alternatively, is the performance of the Poker hand evaluation executed by each AI Agent on each of their turns. Since No-Limit Texas Hold'em Poker involves such an enormous strategy space, the speed at which this occurs will be crucial as it will notably impact the user game experience. For this reason, we will be implementing a custom Python Poker hand evaluation library, which will have to be lightweight and fast, meaning that it will have to execute in reasonable thinking time (human time scale).

The other minor functionalities in which performance will be important are the signing up, logging in and recording of player statistics which all involve simple database queries and shall take no longer than 3000-4000 ms.

Overall, aside from applying the training of the model/AI Agents which will be performed without the user being alerted and the hand evaluation which will occur at a reasonable human thinking time scale, the other interactions that the user will have with the application should be quite immediate and allow the user to have a friendly experience with the No-Limit Texas Hold'em Poker web application..

3.3.3. Scalability Requirements

Since the web application will allow users to only take part in individual poker games with a maximum of 6 players, it is clear how in the future more functionalities such as tournaments, leagues, and maybe even online play could be added to the product.

As the scalability required is often driven by the lifespan and the maturity of the system, the correct cautious measures of the workload will need to be applied in the current implementation of the web application. This will require to design the current project in a manner that will allow for future increase of the system workload, without redesigning the entire application and model.

To implement such improvements, additional data would be required to be stored in our cloud database to keep track of the progress of tournaments and

leagues for example. To then manage all the additional data, user requests, additional processing and transactions, the cloud server storage space would be increased by acquiring the extra storage space necessary (one of the main advantage of cloud databases), and if needed, supplementary optimizations could be applied to the machine learning model, in order to still reach the minimum performance requirements that are currently set.

Moreover, adding an online poker game functionality which would allow multiple different users to compete against each other would require an exponential growth of the system and would certainly require a few significant tweeks to the design without a doubt.

3.3.4. Maintainability Requirements

As maintenance is a key concept for almost any software application, designing the system to have a high maintainability level will be essential. Maintainable software is easy to extend and fix, and this will encourage the software's uptake and use. The modular design that our system will have, with the four main components of it being all separate and distinct (UI, Machine Learning Model, the Cloud Database, and the Web Server), will simplify the handling of issues in any of the segments, and in most cases, it will be clear where a particular problem occurs, if it does.

Additionally, once the system is set up and running as desired, all the actions necessary will be automated, and a manual aid will only be needed when larger, unexpected issues occur. Such situations should not arise, but in case the unexpected happens, if the time constraints allow it, we will implement a feature in the app to send notifications to users to inform them that the system is currently unavailable.

3.3.5. Portability Requirements

One of the explicit design goals for the web application is for it to be directly portable to multiple platforms. When designing and implementing the No-Limit Texas Hold'em Poker web application, it will be specifically planned and modelled to be able to run and be accessible via all modern web browsers.

3.3.6. Security Requirements

Given the application will be storing user credentials in order to provide authentication and authorization of users for recording their personal Poker statistics, it will be essential to securely store and protect this sensitive information from external third parties. For this purpose, we decided on a cloud database platform that guarantees the level of security needed.

Additionally, since the app will not be interacting with any other explicit device components (eg. camera, microphone, etc.), it is not necessary to request

these permissions from the user. The only device feature that the app will require, is the Internet access, for which the permissions is generally automatically given through the web browser, but for specific scenarios where this is not the case, the user will be prompted to approve the request of Internet usage, and if the permission is denied, the app will display an error message.

Regarding the communication with the cloud database, we will be making use of signature-based permissions on top of the user credentials.. These permissions require user authentication and also check that the apps accessing the data are signed using the same signing key. Therefore, these permissions offer a more streamlined, secure user experience.

Since the app will not require to receive data from other external applications on the device, other apps will be disallowed to access and communicate with the Poker web application unless the user explicitly grants these permissions manually. This setting will guarantee an additional level of security.

3.4. Other Requirements

3.4.1. Cloud Database Requirements

As mentioned in section 3.1.3., the cloud database being used to store the large quantities of data for the system is Google Cloud Platform, through its Cloud SQL service.

It is essential that the cloud database being used provides the following features:

- Fully managed MySQL Community Edition databases
- Minimum 400 MB of RAM
- Standard GB/month PD storage
- Continuous support/availability
- Portable API
- Results Visualisation
- Automation modules
- Automated and on-demand backups, and point-in-time recovery.

3.4.2. Web Server Requirements

As also mentioned in section 3.1.3., the Web Server being utilized to deploy our web application into production is Apache Web Server.

The services offered by this server environment are satisfactory for our needs and the minimum requirements provided will be:

- Minimum 8 GB of RAM

- Minimum 50 GB of disk space for cache
- Minimum 4 CPU cores
- Support for up to 50 concurrent users.

3.4.3. Internationalization Requirements

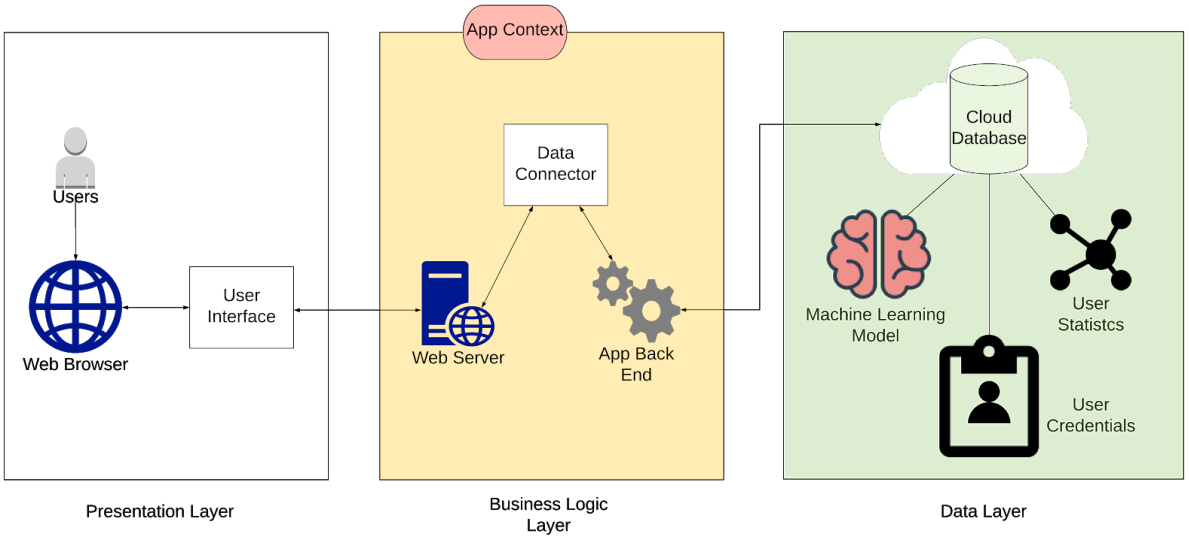
As No-Limit Texas Hold'em Poker is a game played by users all over the world, the application will follow the standard Web Application Internationalization guidelines and will support the following languages:

- English
- Italian
- German
- French
- Portuguese
- Spanish

The language of the application will be set by default according to the language of the web browser the app is being accessed from.

4. System Architecture

4.1. Architecture Diagram



The above Architecture diagram is a representation of how the product interacts with users, entities and data sources. The diagram includes three layers. These layers depict a breakdown of the various components that make up the product. The first layer is the Presentation layer. This layer contains components related to user

interaction with the application. Next is the Business logic layer, in which we find the core functionality of the application. Finally, the lowest layer called the Data layer, contains components responsible for communicating with the data storage.

4.2. Presentation Layer

The Presentation layer is responsible for displaying data. It consists of the user interface component, the user and the web browser. The user interface is the only means for the user to interact with the application. The user does this by using their web browser to access and utilise the features offered by the user interface. Data can flow from the application to the user and vice versa. There are many cases of data flowing back and forth from the user and the application during the course of a game.

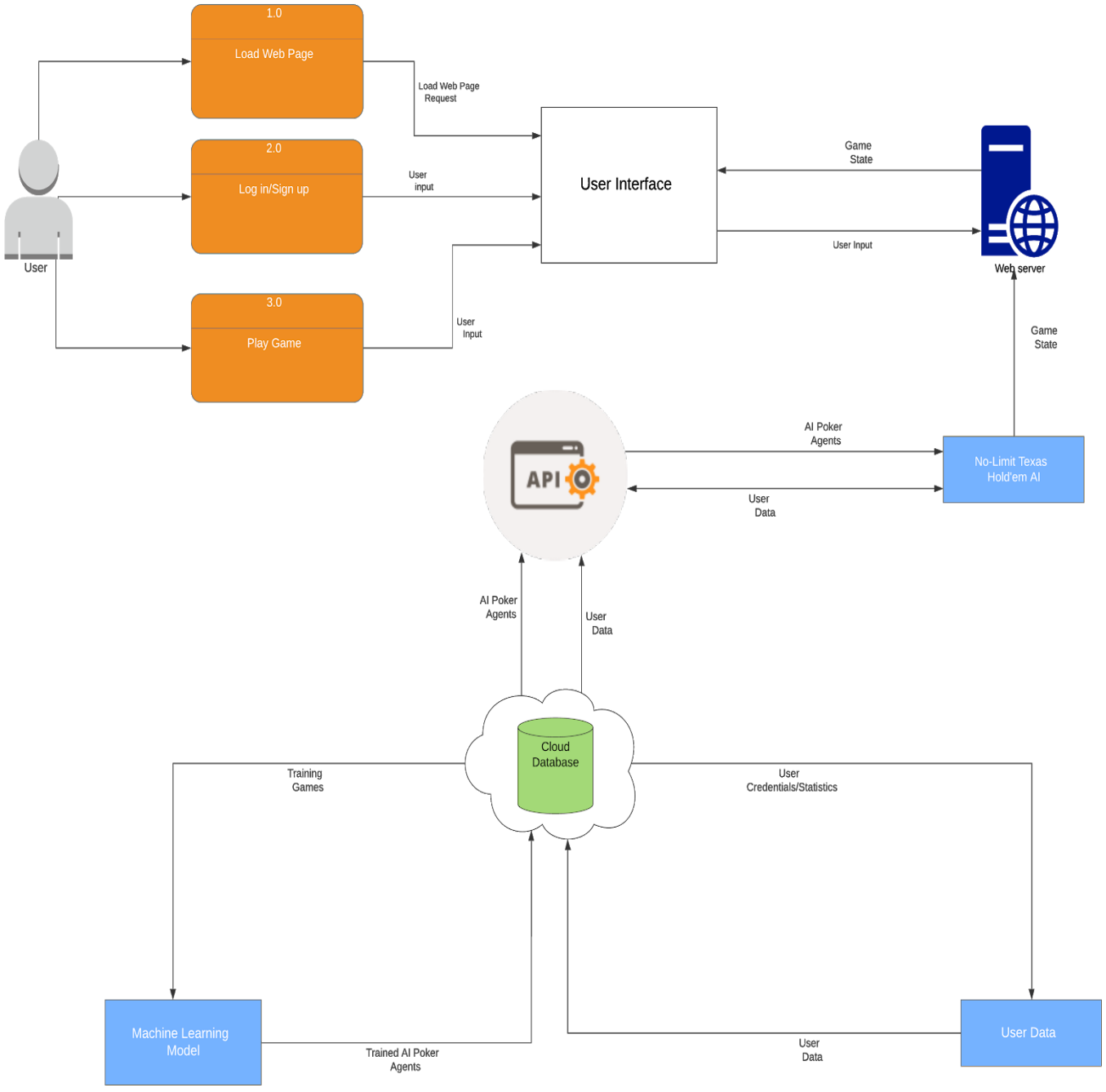
4.3. Business Logic Layer

Within the Business Logic layer lies the main function of the system - the No-Limit Texas Hold'em App along with the Data Connector and Web Server. The Business Logic layer provides data that is required by the Presentation layer. It is also responsible for fetching data from the Data layer and performing necessary tasks with the data. The data is then either sent to the Presentation layer or back to the Data layer. The data is transferred to and from using the data connector. The web server will be used to connect the application to the web so it can be accessed by users online.

4.4. Data Layer

The Data layer consists of a cloud database that will be used to store all data regarding user credentials and user statistics. The machine learning model that will be used to train agents will also be stored here. As more users create accounts and new statistics are created, the database will be updated. These user statistics will be stored on the cloud database, ready to be fetched by the Business Logic layer.

5. High-Level Design



6. Preliminary Schedule

6.1. Project plan

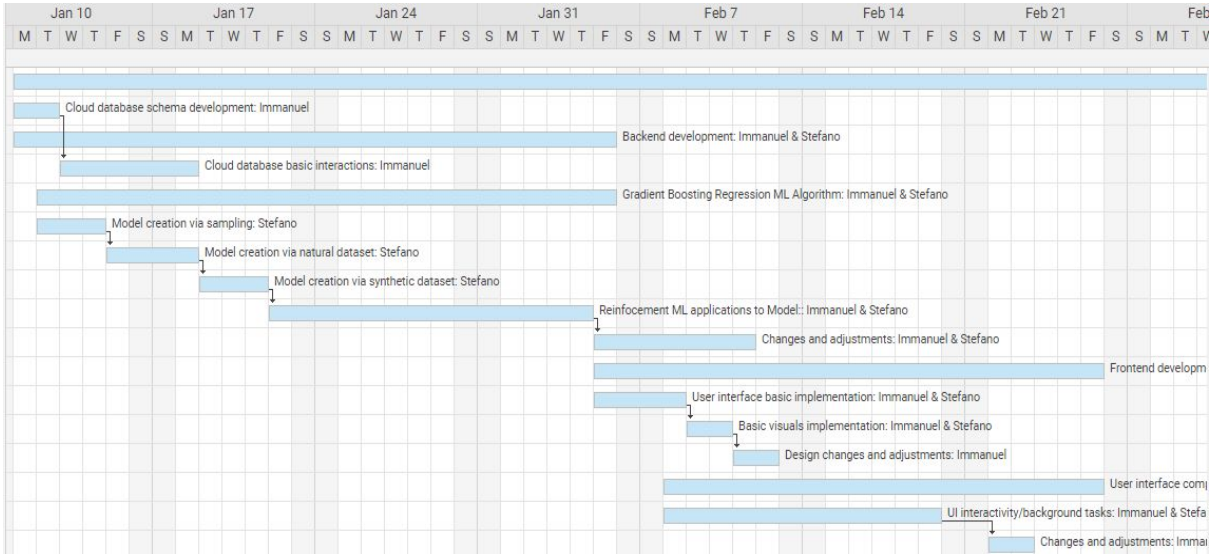
The following table represents an initial version of the project plan, including the tasks to complete, who will complete them, their tentative start and stop dates, and the overall estimated duration of each task. Being a preliminary schedule, it is expected that this will slightly change as the development process proceeds, to take account of unforeseen issues/situations.

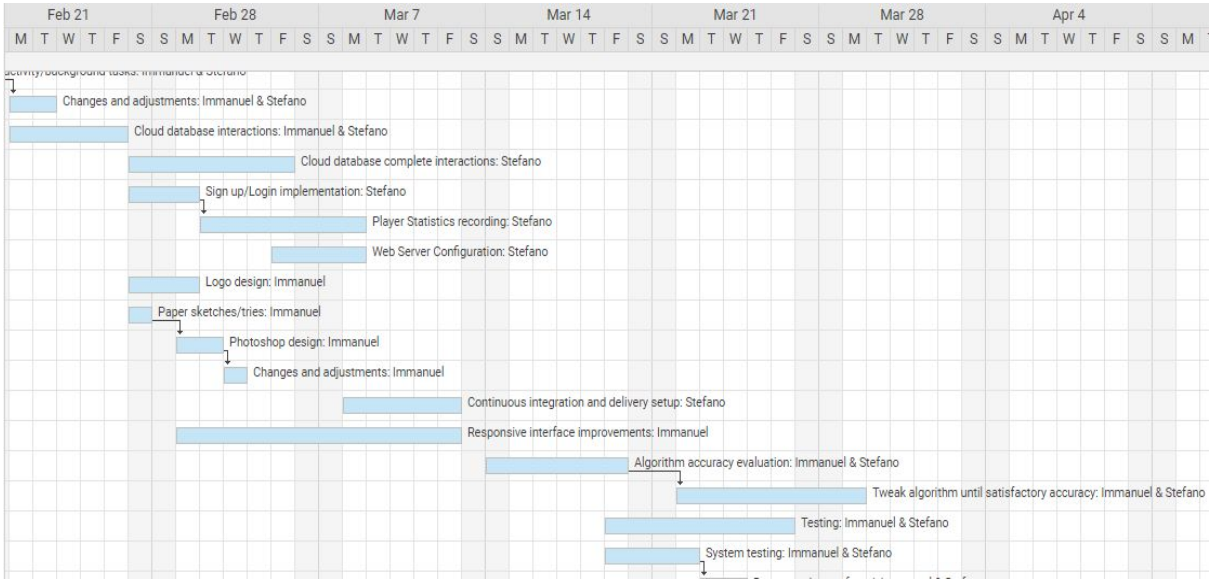
	Title	*Assignee*	*Start date*	*Due date*	*Duration*
2	**No-Limit Texas Hold'em Application**	Immanuel, Stefano	11/01/2021	31/05/2021	
3	Cloud database schema development	Immanuel	11/01/2021	12/01/2021	1d
4	Backend development	Immanuel, Stefano	11/01/2021	05/02/2021	25d
5	Cloud database basic interactions	Immanuel	12/01/2021	15/01/2021	3d
6	Gradient Boosting Regression ML Algorithm	Immanuel, Stefano	12/01/2021	05/02/2021	24d
7	Model creation via sampling	Stefano	12/01/2021	14/01/2021	2d
8	Model creation via natural dataset	Stefano	14/01/2021	16/01/2021	2d
9	Model creation via synthetic dataset	Stefano	16/01/2021	19/01/2021	3d
10	Reinfocement ML applications to Model	Immanuel, Stefano	19/01/2021	01/02/2021	13d
11	Changes and adjustments	Immanuel, Stefano	01/02/2021	05/02/2021	4d
12	Frontend development	Immanuel, Stefano	05/02/2021	27/02/2021	22d
13	User interface basic implementation	Immanuel, Stefano	05/02/2021	08/02/2021	3d
14	Basic visuals implementation	Immanuel, Stefano	05/02/2021	07/02/2021	2d
15	Design changes and adjustments	Immanuel	07/02/2021	08/02/2021	1d
16	User interface complete implementation	Immanuel, Stefano	08/02/2021	27/02/2021	19d
17	UI interactivity/background tasks	Immanuel, Stefano	08/02/2021	20/02/2021	12d
18	Changes and adjustments	Immanuel, Stefano	20/02/2021	22/02/2021	2d
19	Cloud database interactions	Immanuel, Stefano	22/02/2021	27/02/2021	5d
20	Cloud database complete interactions	Stefano	27/02/2021	05/03/2021	8d
21	Sign up/Login implmentation	Stefano	27/02/2021	01/03/2021	4d
22	Player Statistics recording	Stefano	01/03/2021	05/03/2021	4d
23	Web Server Configuration	Stefano	05/03/2021	08/03/2021	3d
24	Logo design	Immanuel	27/02/2021	01/03/2021	4d
25	Paper sketches/tries	Immanuel	27/02/2021	28/02/2021	1d
26	Photoshop design	Immanuel	28/02/2021	01/03/2021	3d
27	Changes and adjustments	Immanuel	01/03/2021	01/03/2021	1d
28	Continuous integration and delivery setup	Stefano	08/03/2021	14/03/2021	6d
29	Responsive interface improvements	Immanuel	01/03/2021	14/03/2021	13d
30	Algorithm accuracy evaluation	Immanuel, Stefano	14/03/2021	19/03/2021	5d
31	Tweak algorithm until satisfactory accuracy	Immanuel, Stefano	14/03/2021	19/03/2021	5d
32	Testing	Immanuel, Stefano	19/03/2021	28/03/2021	9d
33	System testing	Immanuel, Stefano	19/03/2021	22/03/2021	3d
34	Document issues found	Immanuel, Stefano	22/03/2021	23/03/2021	1d
35	Correct issues found	Immanuel, Stefano	23/03/2021	28/03/2021	5d
36	Technical Specification and User Guide	Immanuel, Stefano	28/03/2021	02/04/2021	5d
37	Final Checks/Verifications	Immanuel, Stefano	02/04/2021	05/04/2021	3d
38	Review	Immanuel, Stefano	05/04/2021	07/04/2021	2d
39	Demo preparation	Immanuel, Stefano	07/04/2021	12/04/2021	5d
40	Submission	Immanuel, Stefano	07/05/2021	07/05/2021	1d
41	Demo	Immanuel, Stefano	17/05/2021	17/05/2021	1d

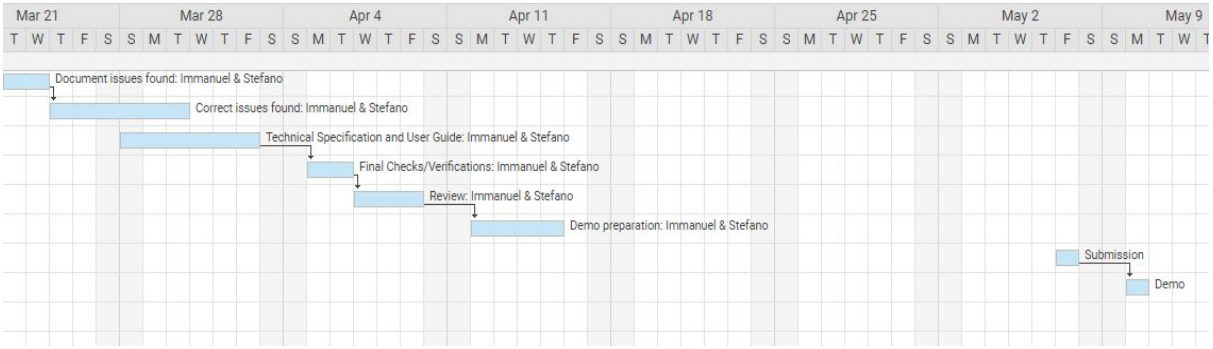
6.2. Gantt Chart

We will use a Gantt Chart to show a preliminary project schedule. The interdependencies between tasks are represented by arrows, indicating that the task to which the arrow points is interdependent to the task from which the arrow starts.

Our Gantt Chart is the following:







It is important to note that the above preliminary schedule indicates that our last day of development and project implementation is arranged to be on the 12th April 2021, a significant amount of days before the submission. The reason for this is that, in case we encounter additional difficulties or unexpected situations, we will have these additional days to allow us to handle inconveniences that occur.

6.3. Hardware, Software, Wetware Requirements

Hardware Requirements:

- PC or Laptop

Software Requirements:

- Windows, Linux, or Mac OS
- Modern Web Browser
- Apache Web Server
- Google Cloud Platform - Cloud SQL
- Jenkins

Wetware Requirements:

- Reinforcement Machine Learning
- Data and File Cloud Storage
- User Credentials Authentication and Authorization
- Web Application Development

7. Supporting Information

7.1. Texas Hold’Em Poker Ruleset

The following table represents a comprehensive list of all Texas Hold’em Poker hand rankings (Rank 1st being the best hand):

Rank	Hand Name	Description
1st	Royal Flush	Ten, Jack, Queen, King and All all of the same suit.
2nd	Straight Flush	Five cards in sequence all of the same suit.
3rd	Four of a Kind	Four cards of the same denomination.
4th	Full House	Three cards of one denomination and two cards of another denomination
5th	Flush	Five cards all of the same suit
6th	Straight	Five cards of any suit in sequence
7th	Three of a Kind	Three cards of the same denomination and two unmatched cards
8th	Two Pairs	Two sets of two cards of the same denomination and any fifth card
9th	Pair	Two cards of the same denomination and any three unmatched cards
10th	High Card	Card of highest denomination when all five cards are unmatched

Note: In Texas Hold'em Poker players use their two in hand cards in combination with the community cards to make the best possible five-card Poker hand.