National College of Ireland

BSc in Computing

2017/2018

Lee Murray

X14538387

X14538387@student.ncirl.ie

Poker AI

Technical Report



**Table of Contents**

[Executive Summary 4](#_Toc499742002)

[1 Introduction 5](#_Toc499742003)

[1.1 Background 5](#_Toc499742004)

[1.2 Aims 9](#_Toc499742005)

[1.3 Technologies 10](#_Toc499742006)

[1.4 Structure 11](#_Toc499742007)

[1.5 System 11](#_Toc499742008)

[1.6 Requirements 11](#_Toc499742009)

[1.6.1 Functional requirements 11](#_Toc499742010)

[1.6.2 Use Case Diagram 12](#_Toc499742011)

[1.6.3 Requirement 1 <User Registration> 12](#_Toc499742012)

[1.6.4 Requirement 2 <User Login> 15](#_Toc499742013)

[1.6.5 Requirement 3 <View User Profile> 17](#_Toc499742014)

[1.6.6 Requirement 3 <Play Poker Menu> 19](#_Toc499742015)

[1.6.7 Requirement 3 <Play Versus AI> 21](#_Toc499742016)

[1.7 User Requirements 24](#_Toc499742017)

[1.7.1 User requirements 24](#_Toc499742018)

[1.7.2 Environmental requirements 24](#_Toc499742019)

[1.7.3 Usability requirements 25](#_Toc499742020)

[1.8 Non-Functional Requirements 25](#_Toc499742021)

[1.8.1 Performance/Response time requirement 25](#_Toc499742022)

[1.8.2 Availability requirement 26](#_Toc499742023)

[1.8.3 Data requirements 26](#_Toc499742024)

[1.8.4 Security requirement 26](#_Toc499742025)

[1.8.5 Reliability requirement 26](#_Toc499742026)

[1.8.6 Maintainability requirement 26](#_Toc499742027)

[1.8.7 Portability requirement 27](#_Toc499742028)

[1.8.8 Extendibility requirement 27](#_Toc499742029)

[1.9 Design and Architecture 29](#_Toc499742030)

[1.10 Implementation 29](#_Toc499742031)

[1.11 Graphical User Interface (GUI) Layout 30](#_Toc499742032)

[1.12 Testing 35](#_Toc499742033)

[1.13 Customer testing 35](#_Toc499742034)

[1.14 Evaluation 36](#_Toc499742035)

[2 Conclusions 37](#_Toc499742036)

[3 Further development or research 38](#_Toc499742037)

[4 References 39](#_Toc499742038)

[5 Appendix 40](#_Toc499742039)

[5.1 Project Proposal 40](#_Toc499742040)

[5.2 Background 40](#_Toc499742041)

[5.3 Technical Details 45](#_Toc499742042)

[5.4 Project Plan 46](#_Toc499742043)

[5.5 Monthly Journals 46](#_Toc499742044)

[5.6 Other Material Used 49](#_Toc499742045)

# Executive Summary

Maximum 300 words. The abstract should mention the problem being addressed, describe the technical solution and briefly report the findings of the evaluation.

In today’s world, Artificial Intelligence exists in every game we play. It was a challenge for computer scientists to beat the professional players since computers were invented. Many games are deterministic perfect information games like chess or checkers where there is no chance and there is no any hidden information from the opponent. This thesis investigates non-deterministic imperfect information games like poker that are very popular in real world. There exists a technique that calculates the strategies over time to win the player by reaching the Nash equilibrium.

Poker is currently the world’s most played card game. Hundreds of thousands of people play poker every day, and can play in a real-life environment or over the internet using a distributed application running a simulation of the game. One of the biggest reasons for poker’s recent success is its fundamental dynamics. The ‘hidden’ elements of the game means players must observe their opponent’s characteristics to be able to arrive at good decisions, given their options. A very good poker player will consistently dominate a sub-optimal opponent, although stochastic elements apply heavy statistical variation to the game, allowing weak players to win occasionally.

The game of poker offers a well-defined domain in which to investigate some fundamental issues in computing science, such as how to handle deliberate misinformation, and how to make intelligent guesses based on partial knowledge. This project will aim to investigate what Artificial Intelligence techniques can be applied to the domain in order to play up to a human standard of decision making.

The findings of the research have application beyond the realm of poker, and can be applied to financial, weather and military domains, or more generally, any domain with a nondeterministic outcome that incorporates stochastic elements.

# Introduction

## Background

In the domain of Artificial Intelligence there has been a plethora of games that have been solved to date. Some examples of these agents are, IBM’s “Deep Blue” for chess, The University of Alberta’s “Chinook” for checkers and Michael Buro’s “Logistello” for Othello.

These agents have effectively solved those games and have beaten the best human minds in the world, demonstrating the power of computational processing. However, what all these games have in common is, they are all “perfect information” games. Perfect information games refer to the game in which each player, at any point in the game has complete knowledge of the current game state. Games like Chess, Checkers and Backgammon are “perfect information” games. Players who play these have perfect knowledge of the game state as they can see all remaining pieces on the game board.

Well known game-search trees, such as alpha-beta search can be used to explore deep into the game tree to find and choose the worse-case action the opponent cannot compete against.

In contrast to this, Poker is a “imperfect information” game, this means that certain information within the game is private, in terms of poker, each player receives private cards. As a result, no player can know the current position in the game tree.

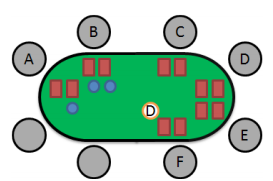
Poker is a non-deterministic game. A player’s actions within the poker domain can never guarantee the same outcome.

Poker has stochastic outcomes. This element of change through random shuffling of the cards creates uncertainty, and adds a great deal of variance to the results.

Texas Hold’em is a poker variation that uses community cards. This variant of Poker was chosen because its rules have specific characteristics that allow new developed methodologies to be adapted to other Poker variations with reduced effort.

**Rules**

At the beginning of every game, two cards are dealt to each player. The dealer player is assigned and marked with a dealer button. The dealer position rotates clockwise from game to game. After that, the two players to the left of dealer post the blind bets. The first player is called small blind, and the other one is called big blind. They respectively post half of minimum and the minimum bet. The player that starts the game is the one on the left of the big blind. One example of an initial table configuration is shown in Figure 2. The dealer is the player at seat F and the small and big blind players are respectively the A and B seats.



**Table Layout**

The first player to act is the player to the left of the big blind (Player C)

And the next player is the closest one to the left of the current player. Each player can choose one of the following actions

* Call: Match the current highest bet
* Raise: Bet higher than the current highest bet
* Fold: Forfeit the hand, thus give up the pot

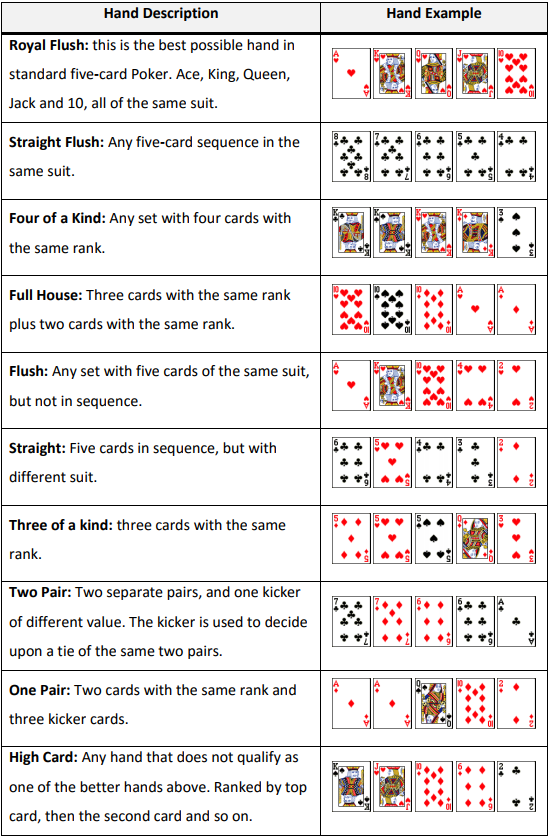
There are four betting rounds in Texas Hold’em, where each round new community cards are revealed.

* Pre-Flop: No community cards
* Flop: three community cards revealed
* Turn: The fourth community card is revealed
* River: The fifth community card is revealed

After the river, if at least 2 players agree to call the pot, the showdown round comes. This is when all players may show their cards and the one with the best hand wins the pot. If players have similar ranked hands, there is a tie and the pot is divided. This is otherwise known as a “chop-pot”

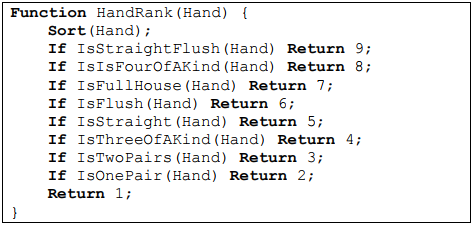
**Hand Rankings**

A poker hand is a set of five cards that identifies the score of a player in poker. The hand is made by combing the player’s personal cards with the community cards. The table below presents the ranking of each combination with a short description.



**Hand Evaluation Algorithms**

The algorithm in which that is used to quantify the agent’s Hand Strength, regardless of all cards being dealt. This algorithm is key, as it considers all the possible better hands the agent could have, the same, and all the worse hands at the point of calculation. The algorithm iterates through all possible starting hands and returns a percentage as a result.



**Hand Potential**

Hand Potential is an algorithm that calculates the possible evolution of the hand quality throughout the game. In Texas Hold’em, when the game reaches the Flop round, there are still two more community cards to be revealed. This means that the current hand rank may improve, since the hand is composed of the set of five available cards that has the highest rank among all available cards. This is an extension of the hand evaluation, but instead of only considering the current available cards, it considers the possible community cards that have not been revealed yet. This also considers that the opponent’s hands might improve as well.

**Non-Deterministic Game**

Non-deterministic games are often described as games with an element of chance. These games do not result in predictable outcome. Examples of such games are Backgammon and Poker (their source of chance is dice and card respectively). What makes these games different from deterministic games are the additional nodes called ‘Chance’ or ‘Nature’ in their game trees.

**Imperfect Information**

Imperfect information game corresponds to the game in which certain information is private, meaning that other players cannot see it. For example, in Poker each player 10 | P a g e receives private cards. As a result of this, no player can clearly know the current position in the game tree.

**The utility or payoff**

The utility in the game is the expected value when a round of a game is played. In the poker game, it is the number of chips that was acquired or lost at the end of the hand (round).

**Nash Equilibrium**

Nash equilibrium is a strategy profile σ where no player can increase their utility by unilaterally changing their strategy (Johanson, 2007): This means that for player 1, there is no other strategy in Σ1 that would produce more utility against σ2 than its strategy in σ. The same is true of player 2. (Johanson, 2007)

## Aims

The game of poker sets the stage for a well-defined domain that allows for the investigation of various fundamental issues in computer science and artificial intelligence, such as how to handle misinformation and how to develop an intelligent agent to process Reponses based on this partial knowledge.

This project will aim to investigate what Artificial Intelligence techniques can be applied in order to,

* Create an agent that will perform on a human standard of decision making that is capable of playing strong no-limit Texas Holdem
* Investigate the characteristics of strong poker playing and compare these results to the agent solution.
* Measure the performance of the agent against human opposition over many hands, and document the results
* Create a fun an interactive experience for the user
* Design an agent to play the no-limit poker variant

## Technologies

**Implementation**

This application will be developed using an Agile software development process. The specific agile practice I intend to adopt for this project is Iterative Development. The main idea behind Iterative Development is to break down the whole project into smaller parts or iterations. This was chosen with the goal of completing significant parts of the project at the end of each iteration. I believe this will be proving to be beneficial as it allows for the development of the more basic aspects of the project such as implementing the Counterfactual Regret Minimization algorithm before jumping straight into Poker AI development.

**Java**

Java is the programming language that will be used to develop the application. This is because Java allows applications to be easily integrated with web application using Java Applet, another reason behind using Java was for future extension of the application to be ported to mobile, using Android application development which uses Java. The object orientated nature of Java, helps in separating the features of the application for easier management and debugging.

**HTML & CSS**

HTML5 and CSS3 are now the standard markup languages for the front end of web applications and websites. In 2017 there is no reason to use earlier versions of either, especially now responsive web design is so prevalent. HTML5’s meta tag, and CSS3’s media queries are key aspects of responsive design.

**MySQL**

MySQL was the primary relational database management system taught to this year’s 4th Year NCI computing students during their time at college. Thus, it made sense to use MySQL for this project rather than having to learn an entire new database system.

## Structure

The structure of this document is as follows, Requirement which contain functional and non-functional requirements. Design and Architecture, which include UML diagrams, use cases, system architecture, hardware and software architecture diagrams. Implementation, Testing, GUI layout, Customer testing, Evaluation and Conclusion

## System

## Requirements

The requirements were among the first things to be considered at the outset of the Poker AI project. To produce these requirements, I posted on various Poker forums to ask them various questions as what they would like to see in a Poker Bot application. I also read comments reviews of different types of Poker applications seen on the Google Play Store that incorporated an AI agent and took their opinions on what they valued into account.

### Functional requirements

* The AI should function as intended
  + This requirement at its most basic level is the core functionality. The AI agent must be able to respond accordingly to the user’s turns
* Hand Evaluation
  + Assessing the probability of a hand improving as more community cards appear
* Better Strategy
  + Determine whether to fold, call/check, or bet/raise in any given situation
* Bluffing
  + Allow the AI to make a profit from a weak hand and to create a false impression about your play
* Unpredictability
  + Make it difficult for your opponent to form an accurate model of the AI’s strategy
* Opponent Modelling
  + Used to determine a likely probability distribution for the opponents hidden cards.
* Registration
  + A new user must be able to sign up to the system
* Login
  + An already registered user must be able to log into the system
* Select AI type
  + Choosing between the three types of AI, chump, conservative and optimal play.
* Leaderboard
  + The system should maintain updated user’s play money amounts and represent it on a leader board
* The Poker-AI application must work on mobile devices
  + From market research, the majority of Poker applications are developed for mobile devices

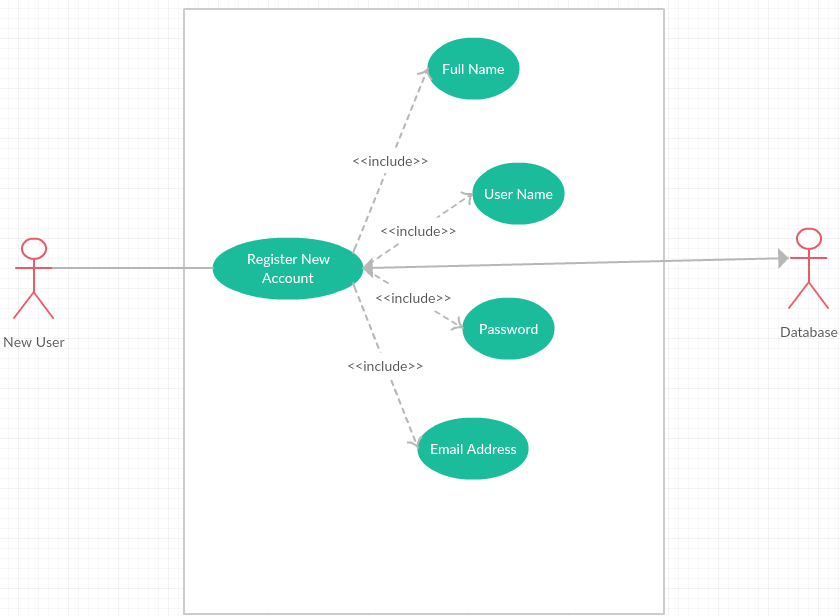
### Use Case Diagram

### Requirement 1 <User Registration>

#### Description & Priority

This requirement relates to an “unregistered” user who is required to create a new account to become an active “registered” user. This process is crucial as it is required to be a “registered” user to have access to the application.

#### Use Case



**Scope**

The scope of this use case is to register a new user to the system

**Description**

This use case describes the registering of a new user to the system, the user is required to make an account to become an active or “Registered User”. This process is crucial as without it, no users can have accounts on the system and thus cannot access the functionality of the application

**Flow Description**

**Precondition**

User has not registered an account

**Activation**

“New User” accesses the application and clicks on the “Registration” button

**Main flow**

* The user enters all requested information i.e name, email etc
* Application displays a confirmation message that the user has successfully created an account

**Alternate flow**

**Fields not completed**

* User has not completed all relevant fields, so the application will highlight all required fields and wait for the user to re-submit with the correct information

**Username Already exists**

* User supplies a “username” that is already registered. The application will inform the user the username is taken.

**Termination**

**Main flow**

* User successfully registers an account

**Alternative flow**

* User must attempt to register again

**Post condition**

**Main flow**

* User is directed to the menu page

**Alternative flow**

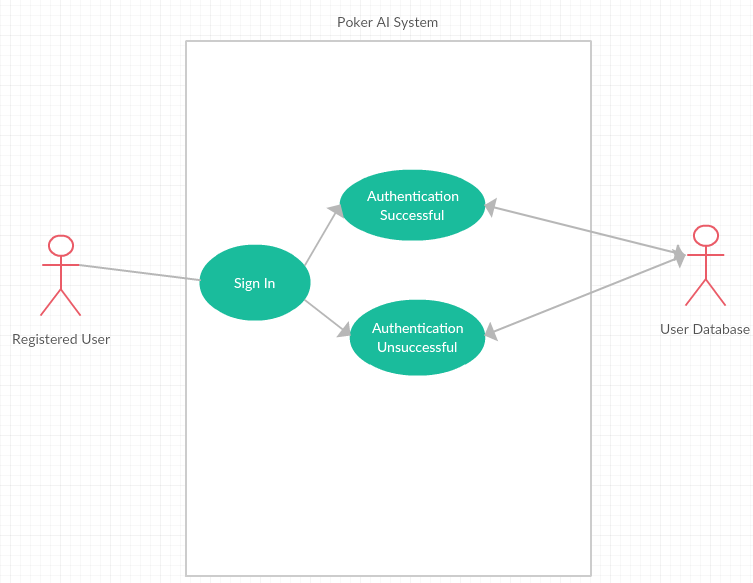
* User must attempt to register again

### Requirement 2 <User Login>

#### Description & Priority

This use case describes the “Registered User” logging into the system. This requirement is key to the system in terms of allowing the user access to functionality as well as their own profile.

#### Use Case



**Scope/Description**

The scope of this use case is to log an existing user into the application, gaining access to the functionality and their own profile.

.

**Flow Description**

**Precondition**

User holds a valid account but has not yet authenticated onto the application during their session

**Activation**

This use case starts when a “Registered User” enters their credentials and presses the “Sign in” button

**Main flow**

* “Registered User” enters their correct credentials and presses the “Sign in” button
* System validates their credentials and provides the user access to the application
* User can now access their profile and play against the poker agent

**Alternate flow**

* “Registered User” enters invalid credentials and presses the “Sign in” button
* Application displays an error message, stating the user needs to re-enter their credentials and attempts to authenticate again

**Termination**

**Main Flow**

* Credentials authenticated

**Alternate flow**

* Credentials unauthenticated

**Post condition**

**Main Flow**

* User can access application and functionality

**Alternate flow**

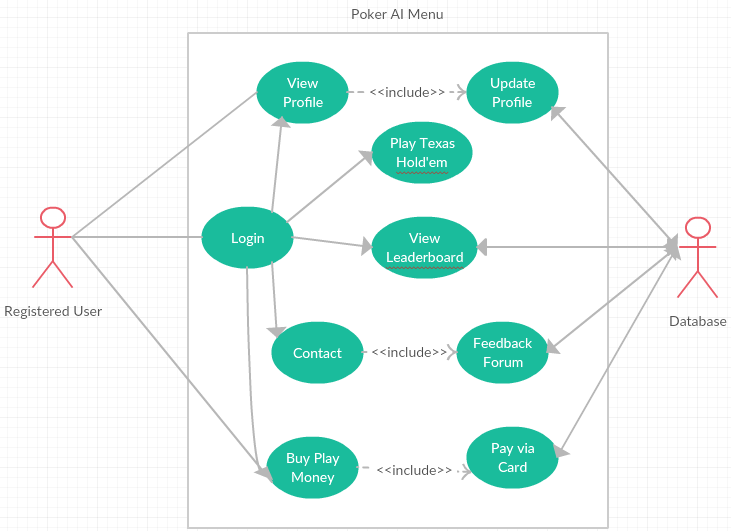
* User returned to the sign in screen

### Requirement 3 <View User Profile>

#### Description & Priority

This requirement presents the ability for the ‘Registered User’ to view their account details, check the leader boards and purchase play money chips. Offers the opportunity for a user to review their details so that they can confirm they are still relevant etc. This requirement is not deemed critical however it is a useful function for each individual user.

#### Use Case



**Scope**

The scope of this use case is to log an existing user into the application, gaining access to the functionality and their own profile.

**Description**

This use case describes the “Registered User” logging into the system. This requirement is key to the system in terms of allowing the user access to functionality as well as their own profile.

**Flow Description**

**Precondition**

User holds a valid account but has not yet authenticated onto the application during their session

**Activation**

This use case starts when a “Registered User” enters their credentials and presses the “Sign in” button

**Main flow**

* “Registered User” enters their correct credentials and presses the “Sign in” button
* System validates their credentials and provides the user access to the application
* User can now access their profile and play against the poker agent

**Alternate flow**

* “Registered User” enters invalid credentials and presses the “Sign in” button
* Application displays an error message, stating the user needs to re-enter their credentials and attempts to authenticate again

**Termination**

**Main Flow**

* Credentials authenticated

**Alternate flow**

* Credentials unauthenticated

**Post condition**

**Main Flow**

* User can access application and functionality

**Alternate flow**

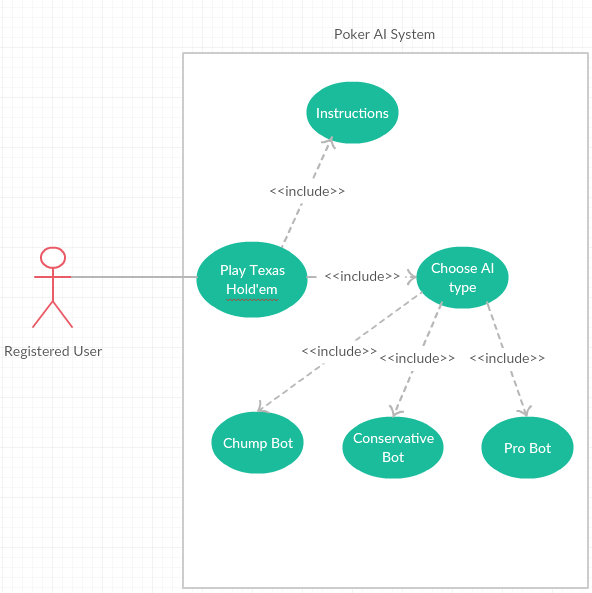
* User returned to the sign in screen

### Requirement 3 <Play Poker Menu>

#### Description & Priority

This requirement relates to the user accessing the core aspect of the application, playing Texas Hold’em, the user is presented with either learning the rules of Texas Hold’em or getting the option to choose three different AI types to play against.

#### Use Case



**Scope**

The scope of this use case is allow the user to learn about Texas Hold’em if they are new, then choose to play versus an AI agent type.

**Description**

This use case describes the “Registered User” accessing the Play Texas Hold’em section of the application.

**Flow Description**

**Precondition**

User holds a registered account, i.e. the account is validated through the system

**Activation**

This use case starts when a “Registered User” presses the “Play Texas Hold’em” button

**Main flow**

* “Registered User” presses the “Play Texas Hold’em” button
* User is presented with Instructions menu or Choose AI type

**Alternate flow**

* User remains on the original page

**Termination**

**Main Flow**

* User navigates back to the main menu

**Alternate flow**

* User navigates back to the main menu

**Post condition**

**Main Flow**

* User can access application and functionality

**Alternate flow**

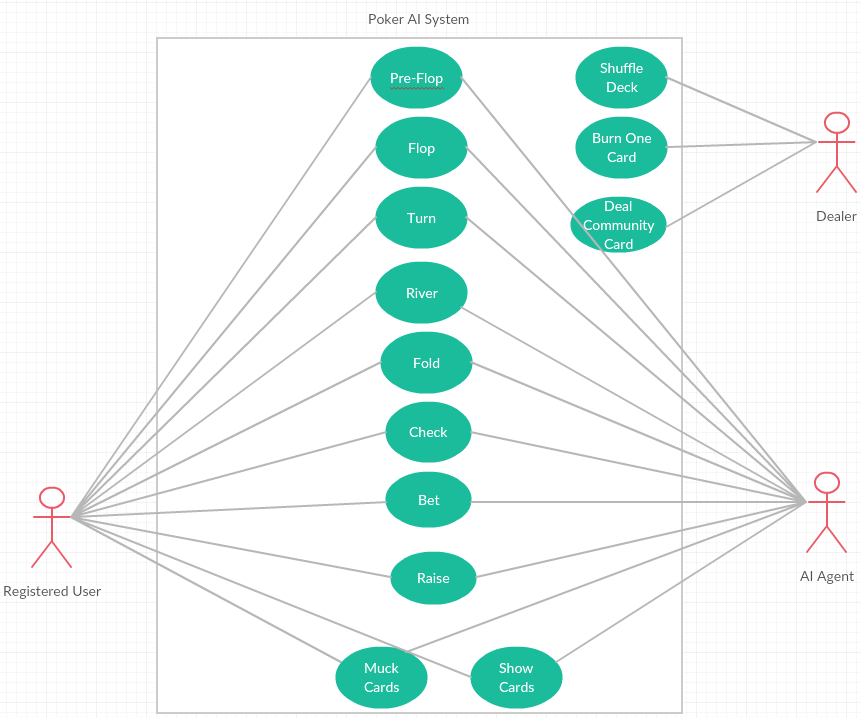
* User returned to the sign in screen

### Requirement 3 <Play Versus AI>

#### Description & Priority

This requirement is the core aspect of the application. The “Registered User” plays Texas Hold’em against the AI agent type of their choice.

#### Use Case



**Scope/Description**

The scope of this use case is the “Registered User” plays Texas Hold’em against the AI agent type of their choice.

**Flow Description**

**Precondition**

User holds a registered account, i.e. the account is validated through the system and has chosen the AI agent type they wish to play against

**Activation**

This use case starts when a “Registered User” chooses the AI agent type they wish to play against

**Main flow**

* “Registered User” choose the AI agent type they wish to play against
* User is put into a heads up or one on one Texas Hold’em game against the AI agent they chose

**Alternate flow**

* User remains on the “Play Poker Menu”

**Termination**

**Main Flow**

* User Win’s versus the AI agent by taking all the agent’s chips
* User Loses versus the AI agent by losing all their chips to the agent

**Alternate flow**

* User navigates back to the “Play Poker Manu”

**Post condition**

**Main Flow**

**Alternate flow**

## User Requirements

The user requirement of Poker AI is to incorporate an application to allow Poker and AI enthusiasts to enjoy fun and simple Poker.

**Internet access:** The device will need Internet access to use the application because it must connect to a server to retrieve information. The faster the internet the faster the server get and post requests (Adekanmbi, 2015/2016).

### User requirements

This section describes the set of objectives and requirements for the system from the customer’s perspective. What are the clients saying they want?

* A new user must be able to sign up to the system
* An already registered user must be able to log into the system
* A user should be able to check the rules on Texas Hold’em through the application
* Should be able to play Texas Hold’em
* A user should be able refill their play money chips that is capped to a certain amount every few hours
* A user should be able to purchase play money chips if they so choose to
* A user should be able to view the leader boards through the application

### Environmental requirements

These are the vital requirements that must be present when developing the application.

* Internet Access: Internet access is required to test functions in application and connecting to database.
* Laptop(Window): This application will be developed Windows laptop with android studio as the Android development IDE.
* Photoshop/Paint: Photoshop was used to customize any images and graphical assets used during the development of my application.

### Usability requirements

This requirement will cover and evaluate the usability requirements for the system application. This outline the standards and objectives to be met regarding the systems.

* **Ease of use:** The application must be user friendly and easy to use.
* **Understandability:** The system should be understandable to the use. Easy to follow the functionates.
* **Operability:** The system should perform as mentioned in the requirement. The app should be consistent in terms of functionality.
* **Attractiveness:** The application should be appealing to users (GUI, Design and layout). The app should use colours that is easy to for the eye.

## Non-Functional Requirements

The divergence between Functional Requirement and Non-Functional Requirement is Functional Requirement deal with what the system shall and much do, while Non-Functional Requirement focuses on, How the system operate.

### Performance/Response time requirement

The AI agent should play their turn in a timely manner, to maintain the flow of the game although play in an efficient way to not reveal information. Taking your time in Texas Hold’em is a key aspect of the game to not reveal information to your opponent, although if the AI takes too long to act every hand, the user will get frustrated and possibly stop playing.

### Availability requirement

The application should be available to be used at any given time. Once a user has accessed the application it should be fully functional and accessible to the user. The application should be free from downtime and if any bugs or errors appear they must immediately be repaired or removed to insure the application services remain up and running to the users.

### Data requirements

User’s data is stored within a MySQL database. This data will be secure and encrypted using an encryption library. As this application will be free and include no monetary aspects, this reduces the impact of storing sensitive payment

### Security requirement

The application should include a robust level of security that ensures users personal information they provided is secured and encrypted on the server. The application files must be secure to avoid any attacks or information leakage. The application should be designed to only accept passwords that contain the strong password criteria.

### Reliability requirement

In many systems reliabilities are a big consideration during the development stage. If a system keeps crashing or has a lot of software bugs this will affect the overall reliability of the application and affect the user’s use of the application services. I will take a range of measures into account ensuring that all software bugs have be eliminated through varies testing methods such as verification, validation, integration testing, functional testing, system testing and. Logical errors will be removed where possible. The system should be able to cope with minor issues that may arise because of internal factors therefore making it reliable.

### Maintainability requirement

The application should maintain updated, with the leader boards showcasing the correct information in real time. The Poker AI’s will be constantly reviewed to asses any weaknesses they may have and address them. This will allow for future improvement of the AI’s as time goes on.

From a security prospective there are many black hat hackers in the current time. One of the goal is to develop an application that not only completes its intended functionality with ease but also impresses and exceeds users' expectations. Where the application happens to fall in terms of performance, functionality or it needs to be updated, several key aspects will be taking to help maintain the application and to help security:

* Defects/Vulnerability

This involves as a developer that reviewing the area of concern for possible abnormalities and addressing them appropriately.

* Code Quality

Looking over the existing code where there lies a potential problem and either patch or improve it.

* Reducing Redundancy

To make the application easy to understand and to make sure that two pieces of code are not doing the same thing, the aim is to eliminate redundancy altogether, assisting in maintaining the steadiness of the overall system.

### Portability requirement

The application and website must be accessible from any device which can use an internet browser. The application must work as smooth as possible across all modern browsers such as Edge, Safari, Chrome and Firefox.

### Extendibility requirement

The system shall be extensible enough so that during future development there can added additional functionalities such as various types of AI agents or expand further on the art style of the application

#### 1.8.9 Recover requirement:

The recovery is the area of security that have to be put In place if there were a significant negative events.

In the case of system shutdown and no response to the user. The system should be down and shouldn’t take no more than one to two working day for the system to be back. Also, Information will be send to the user to let them know when the system is back working.

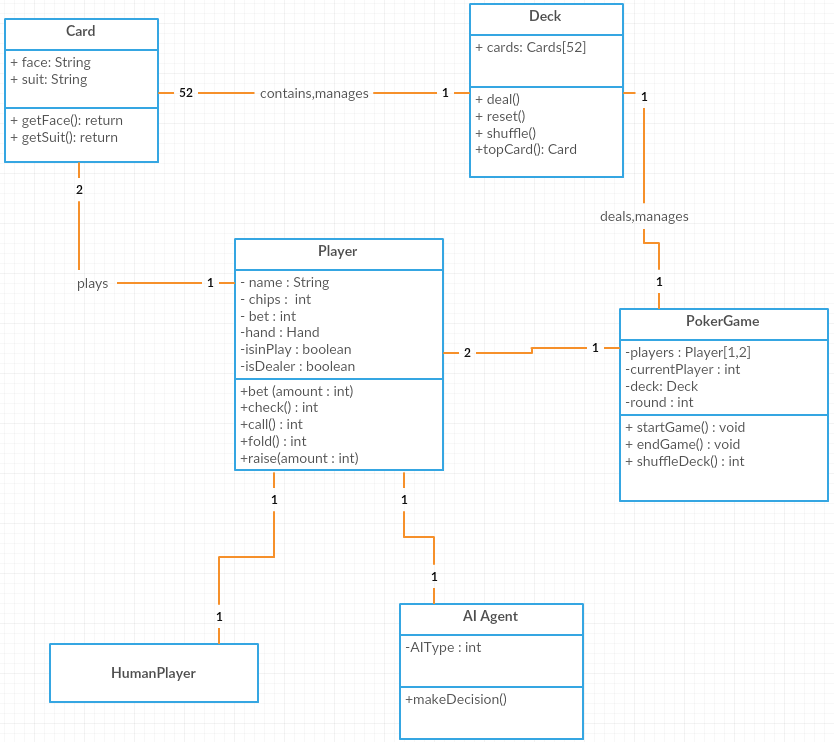
**1.9.0 Reusability requirement:**

Reusability is a valid requirement for all projects that involve software development. The application will aim to have multiple functions and other code that can be re-used and implemented into further or new developments going forward or indeed when creating something new.

Reusability is an important factor and requirement as applications in the real world are based hugely on re-used code. The goal is to develop unique code that can be understand and that am comfortable implementing into new environments.

The login/registration system is a prime example of this, the vast majority of mobile application with user interaction require user login. The database which will be implemented and connected should be flexible and reusable with other applications where some type of information is being stored.

## Design and Architecture



## Implementation

Describe the main algorithms/classes/functions used in the code. Consider to show and explain interesting code snippets where appropriate.

**Components**

Some of the various classes that will be used to create this application and their roles they will play.

**HandEval**

a unique number representing the hand strength of the best 5-card poker hand in the given cards and board. The higher the number, the better the hand is.

**Card**

Represents a play card from a set of cards {0, 51} which map to cards having a suit {0, 3} clubs, diamonds, hearts, spades and a face valu {0, 12} 2, Ace

**Deck**

A deck of 52 Cards which can be dealt and shuffled

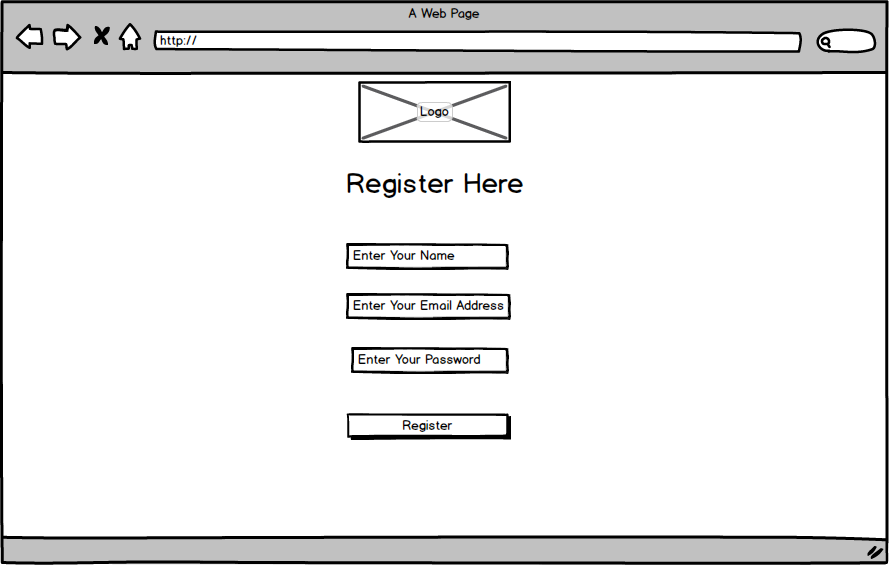
**Player**

This class creates intances of the player which enables quicker access and efficient storage of the information, regardless of which player it is for. The PokerGame class creatss new instances of player at every new interation. The player instance stores information such as history and actions.

## Graphical User Interface (GUI) Layout

Provide screenshots of key screens and explain.

**User Registration**

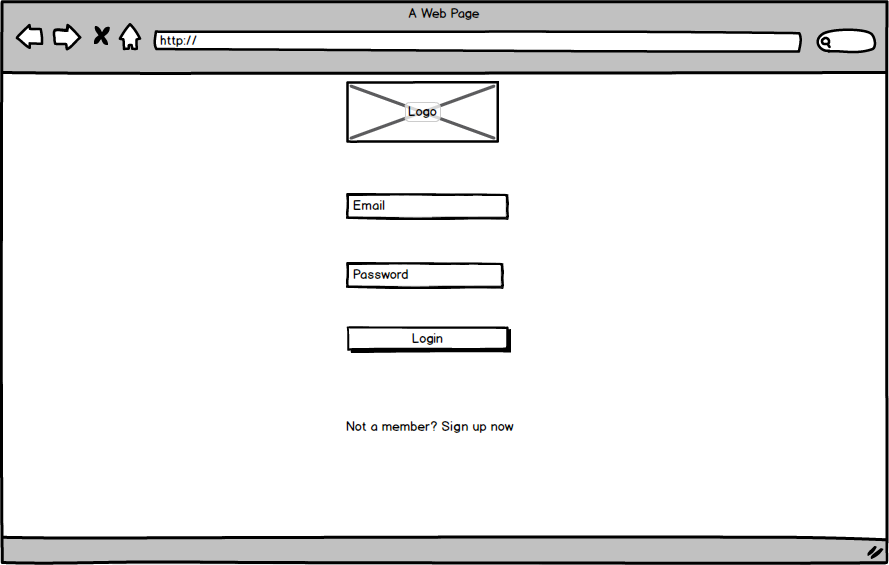
****

This design allows the user to create an account for identity authentication

It takes the following input:

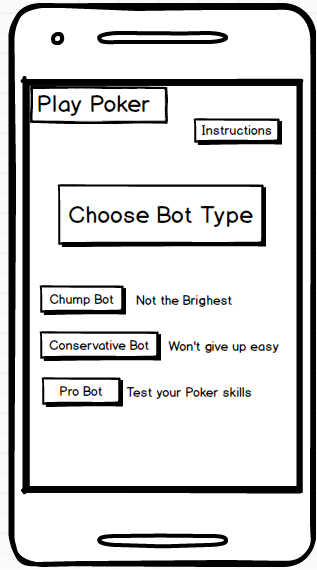
* Username as a unique identifier to recognise the user
* Name to display on the leaderboard
* Email to register the account and confirm
* Password to authorise the user

**Login**



This design attempts to following the use case outlined in the functional requirements on taking the user’s personal information via their registered account.

**Poker Menu**

****

This design shows the user accessing the Play Poker menu, this aspect of the application allows the user the pick the type of AI agent they wish to play against and access the instructions to find out the rules and the card rankings of poker.

**Poker Game**

****

This design is the main aspect of the application, this showcases the user playing No-Limit Texas Holdem versus the AI agent. The user can begin the game by clicking the “Begin Playing” button, this will shuffle the deck and give both players two cards.

## Testing

**Testing**

Since the application will be heavily fragmented, it will be needed to constantly be tested to ensure the algorithm is working correctly.

**Unit Testing**

Manual testing will be done to ensure the calculations are correct, i.e they give the predicted results.

All created objects will be tested to ensure that the expected data is assigned to them and consequently used in game class. The card shuffling which forms a crucial aspect of the system will need to be exhaustively tested.

The program will also need to be tested to ensure the randomness of its actions are maintained.

**User Testing**

This method of testing will be used during every stage of the development.

During User testing, information such as the player’s cards or amount of chips they hold will need to be correct.

The program’s actions must be heavily tested to prevent as little as much bugs as appearing, such as a user may be able to press the bet button twice during their hand. This could cause unexpected and unstable behavior in the application, and could never be detected without user testing.

## Customer testing

Provide evidence for and results of customer testing. This may include ratings or quotes from the customer.

## Evaluation

The system has multiple ways in which it could evolve over time. Some of which include, expanding to other variations of Texas Hold’em, such as six and nine table Texas Holdem, which would just see an influx of different AI agent types on the tables. The system could also expand to different Poker games, such as Pot limit Omaha or Limit Poker, which are less popular games, but still carry a big following of players.

## Definitions, Acronyms, and Abbreviations

**Action.** Refers to a players action on the poker table. Actions may include folding, checking, calling, betting or raising.

**All-in.** When a player commits his entire stack of chips or money to the pot.

**Big Blind.** In Texas Hold'em, the player two places to the left of the dealer posts a forced bet called the big blind. This amount is set depending on the stakes being played. The amount of the big blind is usually twice that of the small blind, and equal to a small bet.

**Community cards.** Refers to the open cards dealt to the table in which players can all observe to make their hands.

**Button**. Also refered to as the ‘Dealer marker’. The player with the button is the player just before the small-blind, and has positional advantage over the entire table for the duration of the hand.

**Flop**. (a) The first three community cards dealt in Texas Hold'em. (b) The betting round that commences after the first three community cards have been dealt.

**Turn.** (a) The fourth community card dealt in Texas Hold'em. (b) The betting round after the fourth community card is dealt.

**River.** (a) The fifth community card dealt in Texas Hold'em. (b) The final betting round after the fifth community card is dealt.

**Range**. A collection of hands usually consisting of one or more hole-cards. This term is commonly used when an oppositions hands are not known, and can only be estimated.

**Pot.** The collection of all the betting amounts on the table at that point in time

**Pre-flop**. Refers to the betting round before the flop has been dealt.

**Post-flop.** Refers to the betting’s rounds that extend during and after the flop has been dealt. This includes the flop, turn and river.

**Hole Card.** A player's private card in poker, unknown to the opponents.

# Conclusions

Describe the advantages/disadvantages, opportunities and limits of the project.

# Further development or research

With more resources, where could the results of this project lead to?

# References

Zakir Baghirov. (2106). *Creating a state-of-the-art based poker player.*Available: http://studentnet.cs.manchester.ac.uk/resources/library/3rd-year-projects/2016/zakir.baghirov.pdf. Last accessed 30th November 2017.

Luís Filipe Guimarães Teófilo. (2016). *Methodologies and Tools for Creating Competitive Poker Playing Agents.* Available: file:///C:/Users/Lee/Downloads/phd-luisteofilo.pdf. Last accessed 30th November 2017.

Trevor Cappallo. (2013). *Development of a Heads-Up Autonomous Poker-Playing Agent.* Available: http://by.tc/assets/cappallo\_poker\_project.pdf. Last accessed 30th November 2017.

Patrick Mccurley. (2009). *An Artificial Intelligence Agent for Texas Hold'em Poker.* Available: http://poker-ai.org/archive/pokerai.org/public/aith.pdf. Last accessed 30th November 2017.

Nick Abou Risk. (2010). *Using Counterfactual Regret Minimization to Create Competitive Multiplayer Poker Agents.* Available: http://poker.cs.ualberta.ca/publications/AAMAS10.pdf. Last accessed 30th November 2017.

Erwin Schuijtvlot. (2011). *Application of AI in poker.* Available: https://beta.vu.nl/nl/Images/werkstuk-schuijtvlot\_tcm235-225501.pdf. Last accessed 30th November 2017.

Michael Bowling, Neil Burch, Michael Johanson. (2010). *Heads-up Limit Hold’em Poker is Solved.* Available: https://webdocs.cs.ualberta.ca/~bowling/papers/15science.pdf. Last accessed 30th November 2017.

Johanson, M. B. (2007). Robust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player. Edmonton: University of Alberta.

Pandolfini, B. (1997). Kasparov and Deep Blue: The Historic Chess Match Between Man and Machine. Fireside Chess Library.

Risk, N. A. (2009). USING COUNTERFACTUAL REGRET MINIMIZATION TO CREATE A COMPETITIVE MULTIPLAYER POKER AGENT. e University of Alberta Libraries

Shor, M. (2006, may 2). Strategy in Game Theory. Retrieved November 30th, 2017, from Game Theory: http://www.gametheory.net/dictionary/PureStrategy.html

# Appendix

## Project Proposal

The main objective is to develop an Artificial Intelligence agent that is capable of good decision making when playing Texas Hold’em poker.

In doing this, the objectives are comprised of 5 points.

* To investigate the characteristics strong poker players possess, and compare these results with the agent solution.
* To measure the performance of the agent against human opposition over a multitude of hands, and document these results.
* Design an efficient Hand Evaluation program
* Design the agent to play no-limit poker variant of Texas Hold’em
* To produce a finalized agent, that produces positive results over a certain number of hands.

The game of poker offers a well-defined domain in which to investigate some fundamental issues in Artificial Intelligence, such as how to handle a game tree that presents *Imperfect Knowledge*, through the means of concealed opponent’s cards.

This project will aim to investigate what Artificial Intelligence techniques can be applied to the domain in order to play up to a human standard of decision making. This will hopefully result in creating a bot that plays Texas Hold’em in a profitable manner.

To simplify the problem, some limitations to the game are made. Firstly, though this is No-Limit Hold’em, the bot’s bets will be fixed. A common raise made by many players is 3 times the bet made in front of them. If the bot is leading out, the bet will be 2/3 of the pot. Lastly, there will be no concept of bankroll, however I will aim to track the winnings and losses of the bot.

## Background

In the domain of Artificial Intelligence there has been a plethora of games that have been solved to date. Some examples of these agents are, IBM’s “Deep Blue” for chess, The University of Alberta’s “Chinook” for checkers and Michael Buro’s “Logistello” for Othello.

These agents have effectively solved those games and have beaten the best human minds in the world, demonstrating the power of computational processing. However, what all these games have in common is, they are all “perfect information” games. Perfect information games refer to the game in which each player, at any point in the game has complete knowledge of the current game state. Games like Chess, Checkers and Backgammon are “perfect information” games. Players who play these have perfect knowledge of the game state as they can see all remaining pieces on the game board.

Well known game-search trees, such as alpha-beta search can be used to explore deep into the game tree to find and choose the worse-case action the opponent cannot compete against.

In contrast to this, Poker is a “imperfect information” game, this means that certain information within the game is private, in terms of poker, each player receives private cards. As a result, no player can know the current position in the game tree.

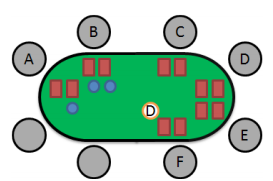
Poker is a non-deterministic game. A player’s actions within the poker domain can never guarantee the same outcome.

Poker has stochastic outcomes. This element of change through random shuffling of the cards creates uncertainty, and adds a great deal of variance to the results.

Texas Hold’em is a poker variation that uses community cards. This variant of Poker was chosen because its rules have specific characteristics that allow new developed methodologies to be adapted to other Poker variations with reduced effort.

**Rules**

At the beginning of every game, two cards are dealt to each player. The dealer player is assigned and marked with a dealer button. The dealer position rotates clockwise from game to game. After that, the two players to the left of dealer post the blind bets. The first player is called small blind, and the other one is called big blind. They respectively post half of minimum and the minimum bet. The player that starts the game is the one on the left of the big blind. One example of an initial table configuration is shown in Figure 2. The dealer is the player at seat F and the small and big blind players are respectively the A and B seats.



**Table Layout**

The first player to act is the player to the left of the big blind (Player C)

And the next player is the closest one to the left of the current player. Each player can choose one of the following actions

* Call: Match the current highest bet
* Raise: Bet higher than the current highest bet
* Fold: Forfeit the hand, thus give up the pot

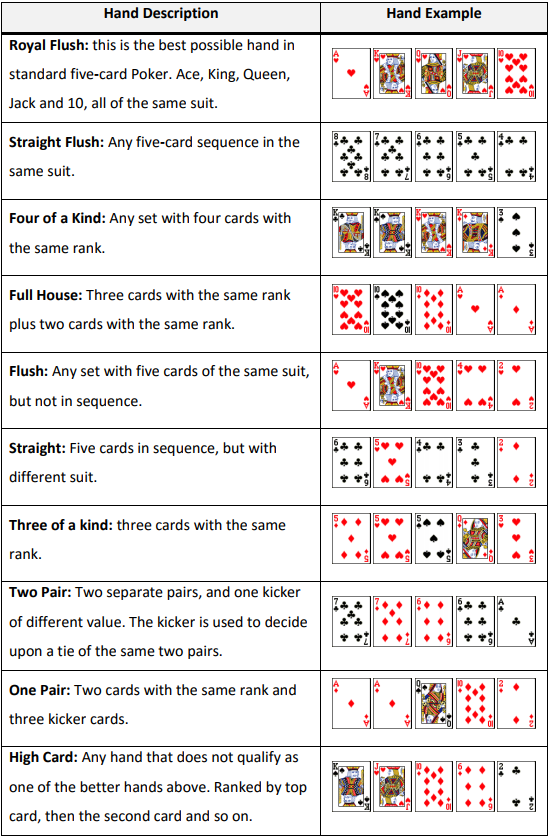
There are four betting rounds in Texas Hold’em, where each round new community cards are revealed.

* Pre-Flop: No community cards
* Flop: three community cards revealed
* Turn: The fourth community card is revealed
* River: The fifth community card is revealed

After the river, if at least 2 players agree to call the pot, the showdown round comes. This is when all players may show their cards and the one with the the best hand wins the pot. If players have similar ranked hands, there is a tie and the pot is divided. This is otherwise known as a “chop-pot”

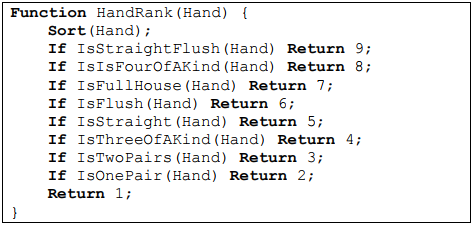
**Hand Rankings**

A poker hand is a set of five cards that identifies the score of a player in poker. The hand is made by combing the player’s personal cards with the community cards. The table below presents the ranking of each combination with a short description.



**Hand Evaluation Algorithms**

The algorithm in which that is used to quantify the agent’s Hand Strength, regardless of all cards being dealt. This algorithm is key, as it considers all the possible better hands the agent could have, the same, and all the worse hands at the point of calculation. The algorithm iterates through all possible starting hands and returns a percentage as a result.



**Hand Potential**

Hand Potential is an algorithm that calculates the possible evolution of the hand quality throughout the game. In Texas Hold’em, when the game reaches the Flop round, there are still two more community cards to be revealed. This means that the current hand rank may improve, since the hand is composed of the set of five available cards that has the highest rank among all available cards. This is an extension of the hand evaluation, but instead of only considering the current available cards, it considers the possible community cards that have not been revealed yet. This also considers that the opponent’s hands might improve as well.

**Non-Deterministic Game**

Non-deterministic games are often described as games with an element of chance. These games do not result in predictable outcome. Examples of such games are Backgammon and Poker (their source of chance is dice and card respectively). What makes these games different from deterministic games are the additional nodes called ‘Chance’ or ‘Nature’ in their game trees.

**Imperfect Information**

Imperfect information game corresponds to the game in which certain information is private, meaning that other players cannot see it. For example, in Poker each player 10 | P a g e receives private cards. As a result of this, no player can clearly know the current position in the game tree.

**The utility or payoff**

The utility in the game is the expected value when a round of a game is played. In the poker game, it is the number of chips that was acquired or lost at the end of the hand (round).

**Nash Equilibrium**

Nash equilibrium is a strategy profile σ where no player can increase their utility by unilaterally changing their strategy (Johanson, 2007): This means that for player 1, there is no other strategy in Σ1 that would produce more utility against σ2 than its strategy in σ. The same is true of player 2. (Johanson, 2007)

## Technical Details

**Methodology**

An agile software development process will be implemented in the development for this project. The specific agile practice that will be adopted for this project is the Iterative Development. The main idea behind iterative development is the breaking down of the whole project into smaller parts or iterations.

**Java**

The Java programming language will be used to develop the application this is because the object-oriented nature of the language, will aid in separating the features of the application for easier management and debugging.

**HTML & CSS**

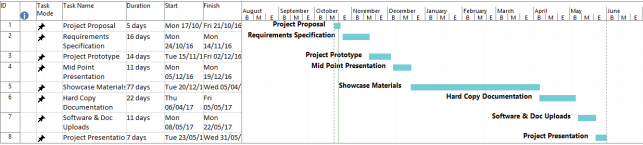
HTML & CSS will be used to develop the web page in which the Java applications will be embedded in. The web page will need to be appealing to the user and incorporate 5 pages,

* Main/Poker GUI Page
* Leader boards
* Login/Signup
* Account Page
* About Page

**MySQL**

MySQL will be used to store the user’s login information and keep track of user’s chips in the database.

## Project Plan



## Monthly Journals

**September Journal**

Student Name: Lee Murray

Programme: BSc Computing

Month: September 2017

When attempting to come up with an idea for my final year project, I wanted to do something aligning to my cyber security stream, although I was unable to come up with a solid idea in time for the pitch so instead of going in and saying I don’t have an idea, I pitched a Poker application which included an Artificial Intelligent agent, which I presumed would get rejected. To my surprise it got accepted. I have a keen interest in Poker and have played it for a couple of years, although developing an AI agent seems as an ambition task, I’m positive it will also be very interesting.

**Supervisor Meetings**

I didn’t have a supervisor meeting during September

**October Journal**

Student Name: Lee Murray

Programme: BSc Computing

Month: October 2017

I am very glad to say that, my project idea was approved by the college committee, who queried me about the application that I want to build. So, this month my achievements were: First, I prepared for the presentation Pitch and then I presented my project idea in front of the college committee. Second, I started and completed the project proposal and then I have uploaded it to Moodle on time. Thirdly I began working on my Requirements specifications. I also did some research on the topic of Game Theory in Artificial Intelligence and Bots in Poker

**Supervisor Meetings**

I didn’t have a supervisor meeting during September

**November Journal**

Student Name: Lee Murray

Programme: BSc Computing

Month: November 2017

During October, I met with my Supervisor Vikas, whom pointed in a direction to find the best programming language I can use to develop the algorithm for the AI agent, with the use of document surveys. As I already have previous knowledge of Java from using it in college, I decided to stick with what I know, rather than take on a new language in this already difficult enough task. We also discussed about the domain of the project, whether to develop it on a web application or on Android. Vikas recommended doing it on a web application to keep it as simple as possible as I have not worked with Android studio before. Although he offered a good suggestion to include on the extensibility of the application that as further development takes place, using Android studio is a possibility.

**My Achievements**

Finished the Requirements Spec

**Supervisor Meetings**

Date of Meeting: 14th November Items discussed: Discussing Project Idea, What language and domain to use.

## Other Material Used

Any other reference material used in the project for example evaluation surveys etc.