|  |
| --- |
| HDSSD |
| Requirements Specification (RS) |
| Poker AI |

|  |
| --- |
| Lee Murray  [Pick the date] |

Requirements Specification (RS)

Document Control

Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Version** | **Scope of Activity** | **Prepared** | **Reviewed** | **Approved** |
| 14/10/2005 | 1 | Create | AB | X | X |
| 21/10/2005 | 2 | Update | CD |  |  |

Distribution List

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Version** |
| Eamon Nolan | Lecturer |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Related Documents

|  |  |
| --- | --- |
| **Title** | **Comments** |
| Title of Use Case Model |  |
| Title of Use Case Description |  |

**Table of Contents**

[1 Introduction 4](#_Toc316977392)

[1.1 Purpose 4](#_Toc316977393)

[1.2 Project Scope 4](#_Toc316977394)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc316977395)

[2 User Requirements Definition 4](#_Toc316977396)

[3 Requirements Specification 4](#_Toc316977397)

[3.1 Functional requirements 4](#_Toc316977398)

[3.1.1 Use Case Diagram 5](#_Toc316977399)

[3.1.2 Requirement 1 <User Registration> 5](#_Toc316977400)

[3.1.3 Requirement 2 <User Login> 6](#_Toc316977401)

3.1.4 Requirement 3 <View User Profile> 6

3.1.5 Requirement 4 <Play Poker Menu> 6

3.1.6 Requirement 4 <Play Versus AI> 7

[3.2 Non-Functional Requirements 7](#_Toc316977402)

[3.2.1 Performance/Response time requirement 8](#_Toc316977403)

[3.2.2 Availability requirement 8](#_Toc316977404)

[3.2.3 Recover requirement 8](#_Toc316977405)

[3.2.4 Robustness requirement 8](#_Toc316977406)

[3.2.5 Security requirement 8](#_Toc316977407)

[3.2.6 Reliability requirement 8](#_Toc316977408)

[3.2.7 Maintainability requirement 8](#_Toc316977409)

[3.2.8 Portability requirement 8](#_Toc316977410)

[3.2.9 Extendibility requirement 8](#_Toc316977411)

[3.2.10 Reusability requirement 8](#_Toc316977412)

[3.2.11 Resource utilization requirement 8](#_Toc316977413)

[4 GUI 8](#_Toc316977414)

[5 System Architecture 8](#_Toc316977415)

[6 System evolution 8](#_Toc316977416)

# Introduction

## Purpose

The purpose of this document is to set out the requirements for the development of showcasing the power of artificial intelligence in an imperfect information games such as Texas Hold’em.

The intended customers are people who are interested in the development of AI through the means of game theory and who also share a love of playing poker.

## Project Scope

The scope of the project is to develop a …………….The system shall have a ……………

## Definitions, Acronyms, and Abbreviations

AD Another Definition

……..

# User Requirements Definition

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 three hours
* A user should be able to purchase play money chips if they so choose to
* A user should be able to view the leaderboards through the application

# Requirements Specification

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

## Functional requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

* 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 leaderboard
* The Poker-AI application must work on mobile devices
  + From market research the majority of Poker applications are developed for mobile devices

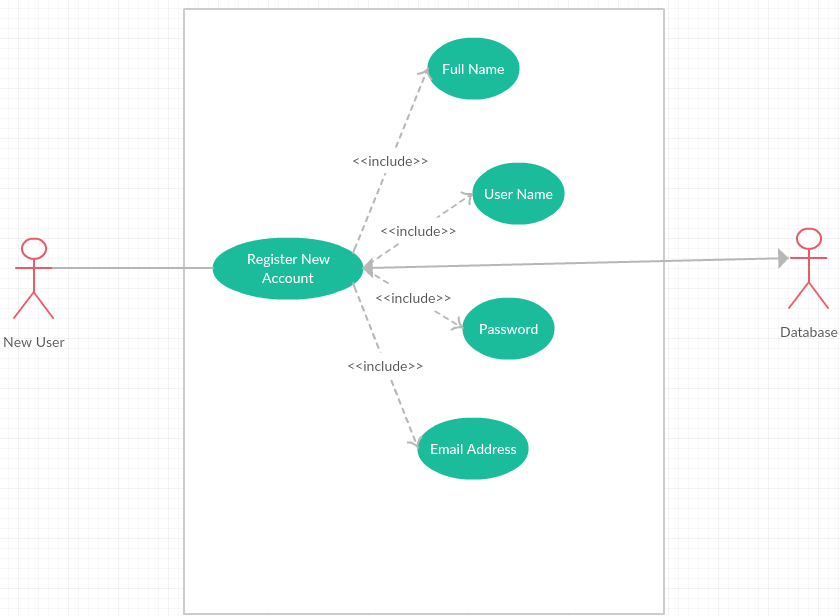
### Use Case Diagram

### 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**

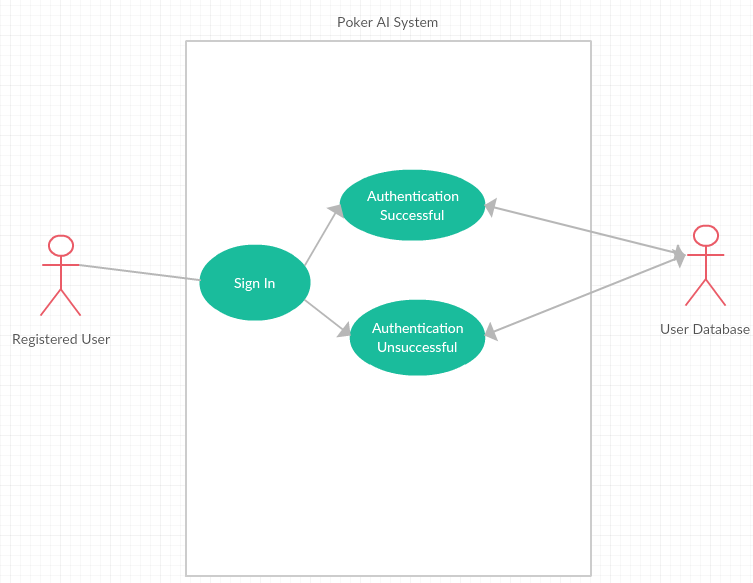
* User must attempt to register again

### 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**

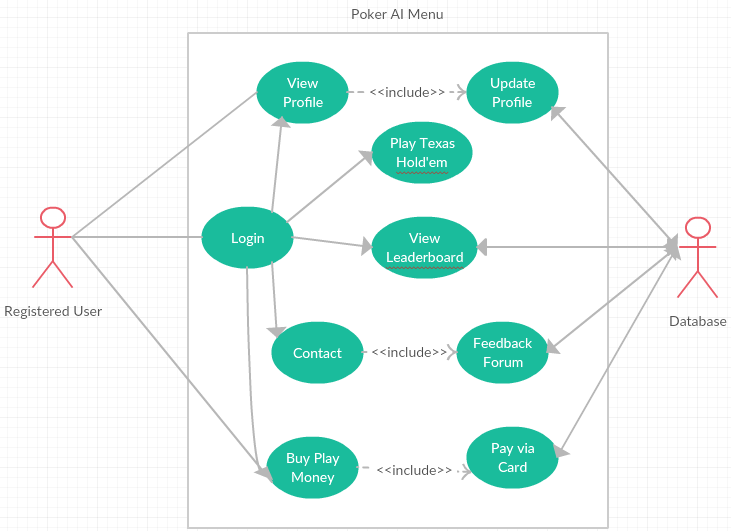
* User returned to the sign in screen

### 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**

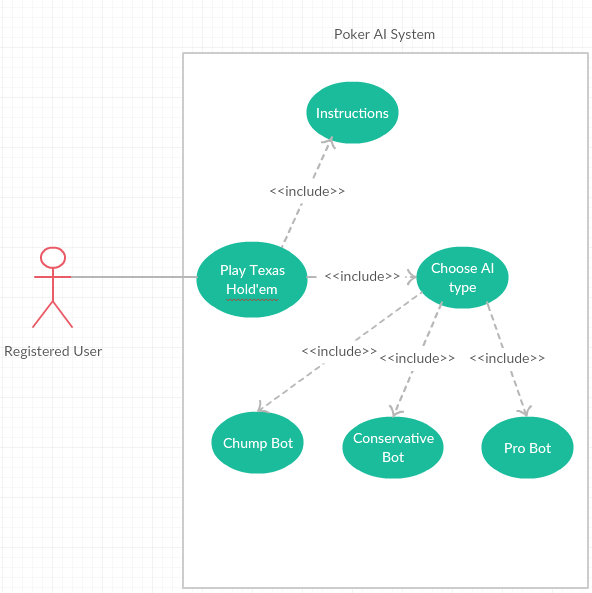
* User returned to the sign in screen

### 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**

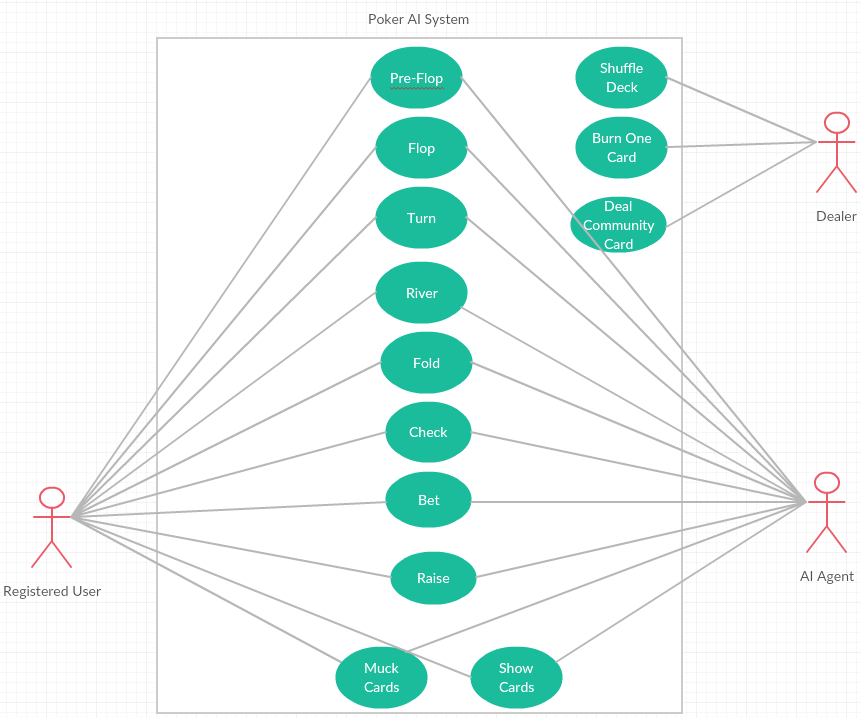
* User returned to the sign in screen

### 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**

## Non-Functional Requirements

### 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.

### Recover requirement

### Robustness requirement

### 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.

### Portability requirement

The application should be able to run on any device connected to the internet.

### Extendibility requirement

### Reusability requirement

### Resource utilization requirement

# Interface requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include APIs, web services, shared memory, data streams, and so forth. Most systems would have a GUI. Add more subsections for other interfaces as reuired.

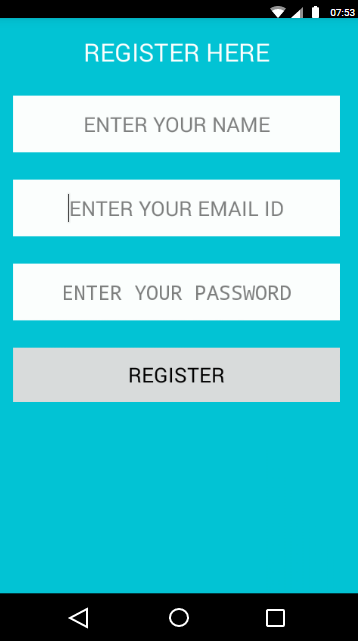
**Meerkat API**

* Easily allows the creation of Poker agents that also can be tested using game simulations that support this API such as as Open Meerkat Test Bed

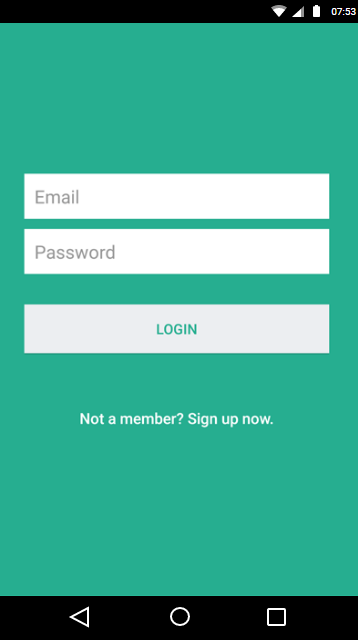
## GUI

Include mock-ups of the key pages or stages of the system. Explain how they are linked. Explain how you addressed above requirements in the design. It is important that the mock-ups are in line with the functional requirements above, e.g., if one of your requirements is “user registration” then one of the screens listed in this section should show a registration page.

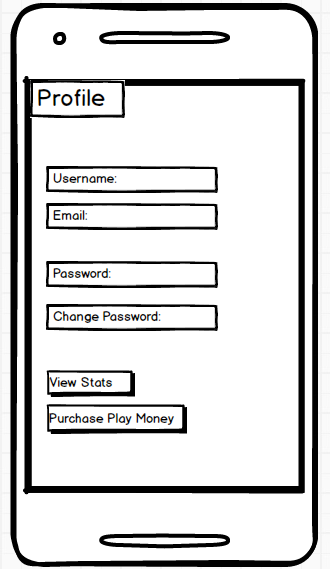
**User Registration**

****

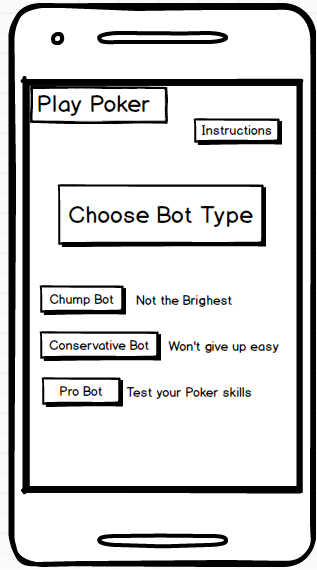
**User Login**

****

**View Profile**

****

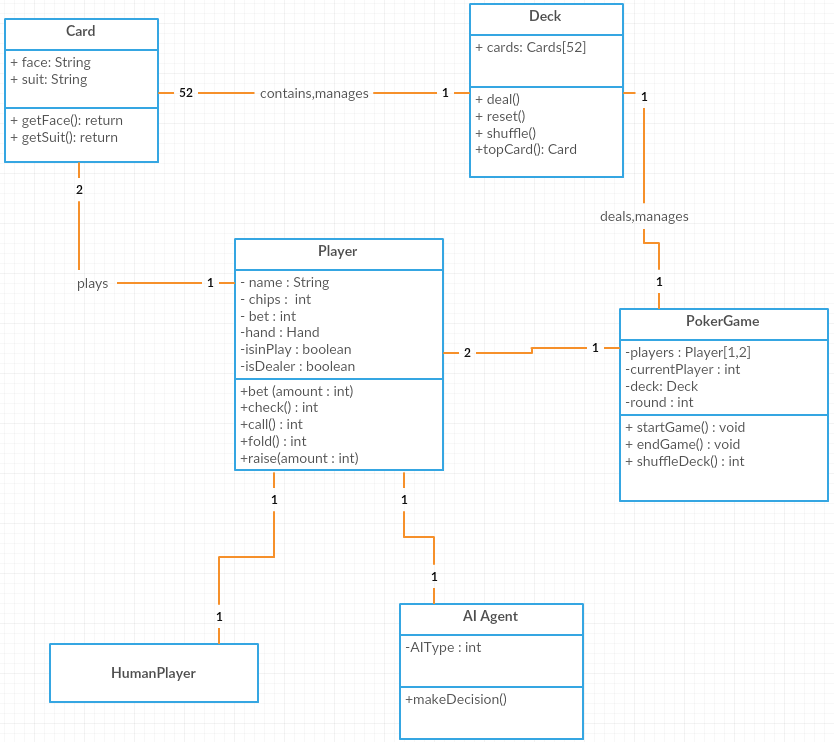
**Play Poker Menu**

****

## Application Programming Interfaces (API)

Explain which interfaces your system offers or which are used by your system. Examples include Google maps and Weka.

# System Architecture



# System Evolution

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.