***SOFTWARE REQUIREMENTS SPECIFICATION***

Online BlackJack

Nathan Sikon

Keaton Dinger

LiLou DeRiveria

Hitesh Kukreja

**1.0 Introduction**

This section provides an overview of the entire requirement document. This document describes all data, functional and behavioral requirements for software.

**1.1 Goals and objectives**

Overall goals and software objectives are described.

Goal: To create an engaging and user-friendly online blackjack game that replicates the experience of playing blackjack in a casino.

**1.2 Statement of scope**

A description of the software is presented. Major inputs, processing functionality and outputs are described without regard to implementation detail.

* Inputs: User actions(bet placement, stand, hit, double down, etc.), game settings, and user details
* Processing: Card Shuffling, game logic(calculating scores, deteremine winner, etc.)
* Output: Game visuals(cards, chips, scores), win/lose notifications, and user statistics.

**1.3 Software context**

The software is placed in a business or product line context. Strategic issues relevant to context are discussed. The intent is for the reader to understand the 'big picture'.

Our online blackjack game is intended for entertainment purposes and can be used by occasional or skilled players. The software will be developed as a web-based application, and we aim for it to be accessible across multiple browsers. The development process will focus on a consistent and smooth user experience as well as compatibility over various browsers.

**1.4 Major constraints**

Any business or product line constraints that will impact the manner in which the software is to be specified, designed, implemented or tested are noted here.

* Time: The project has a set deadline based on our course schedule.
* Browser Consistency: The game must function similarly and fluidly over various web browsers.
* Performance Demand: The game should have very little lag when processing and assessing user inputted moves.

**2.0 Usage scenario**

This section provides a usage scenario for the software. It organized information collected during requirements elicitation into use-cases.

**2.1 User profiles**

The profiles of all user categories are described here.

* Standard Users: Players wanting to participate in an online blackjack game.
* Developers: Our group members, responsible for testing and maintaining the software.

**2.2 Use-cases**

All use-cases for the software are presented.

* User creates an account/logins/no account needed to access the game
* User places bets, receives cards, and makes decisions based off hand (stand, hit, etc.)
* User can restart game or cash out at a certain point

**2.3 Special usage considerations**

Special requirements associated with the use of the software are presented.

* Users should be given feedback for each move in a game.
* The dealer’s moves should impersonate real casino behavior.
* The system should block incorrect moves.

**3.0 Data Model and Description**

This section describes information domain for the software

**3.1 Data Description**

Data objects that will be managed/manipulated by the software are described in this section.

* Card: Attributes like the card ID, suit, and value
* Game: Game ID, player ID, bet amount, cards dealt, and result.
* User: Login ID, balance, cards used

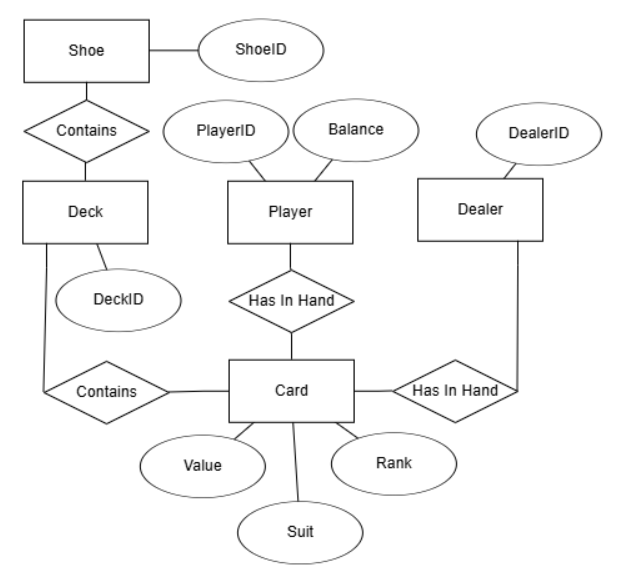
**3.1.1 Data objects**

Data objects and their major attributes are described.

* Card (Value, Suit, Rank)
* Deck (Assortment of Cards)
* Shoe (Assortment of Decks)
* Player (Hand, Balance)
* Dealer (Hand)

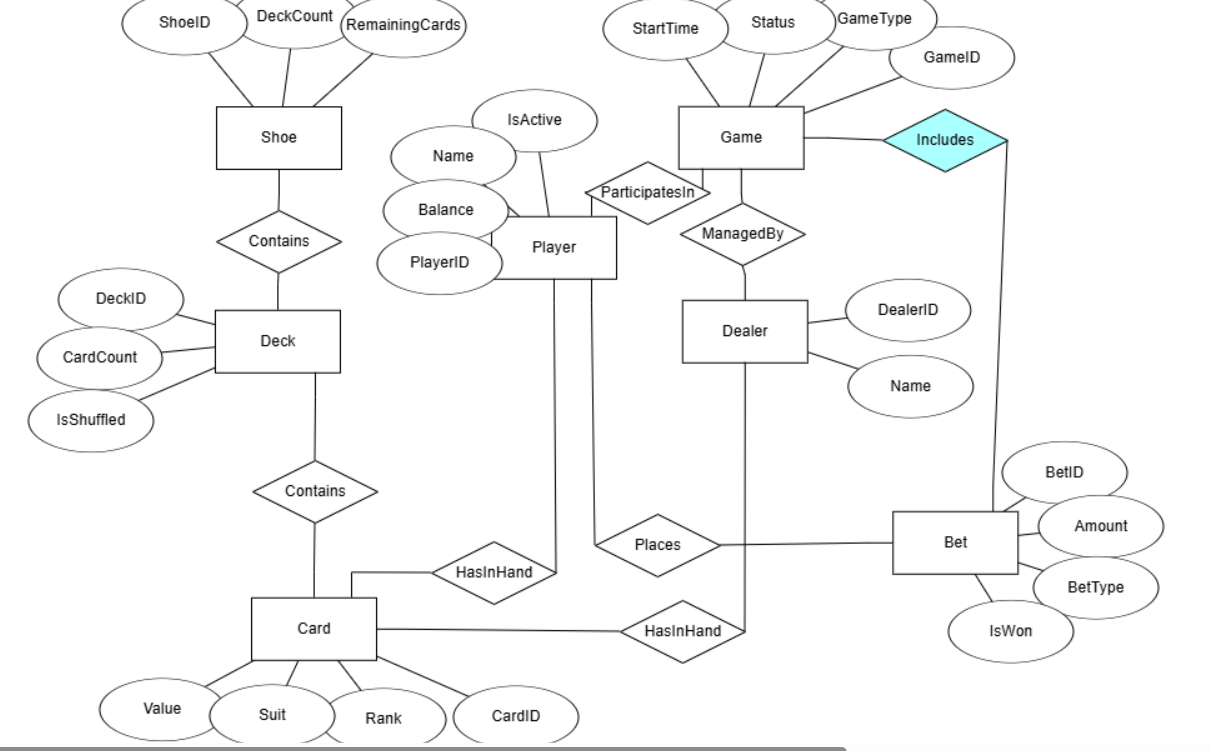
**3.1.2 Relationships**

Relationships among data objects are described using an Entity-Relationship Diagram (ERD) like form. No attempt is made to provide detail at this stage.



**3.1.3 Complete data model**

An ERD for the software is developed



**3.1.4 Data dictionary**

A reference to the data dictionary is provided. The dictionary is maintained in electronic form.

**CARD**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | TYPE | DESC. | CONSTRAINTS |
| cardID | int | Unique card identifier | Primary key |
| Value | str | Values of numbered and face cards | Not null |
| Suit | str | Hearts, spades, clubs, diamonds | Not null |
| rank | str | The rank/numerical value of the card | Not null |

**DECK**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUTES** | **TYPE** | **DESC.** | **CONSTRAINTS** |
| **DeckID** | **int** | **Unique deck identifier** | **Primary key** |
| **CardCount** | **int** | **Number of cards in the deck** | **Default = 52** |
| **Is Shuffled** | **boolean** | **Indicates if deck was shuffled** | **Default = false** |

**SHOE**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | TYPE | DESC. | CONSTRAINTS |
| ShoeID | int | Unique shoe identifier | Primary key |
| DeckCount | int | Number of decks in shoe | Default = 1 |
| RemainingCards | int | Count of cards remaining in shoe | Calculated field |

**PLAYER**

|  |  |  |  |
| --- | --- | --- | --- |
| ATTRIBUTE | TYPE | DESC. | CONSTRAINTS |
| PlayerID | int | Unique player identifier | Primary key |
| Name | str | Player’s name | Not null |
| Balance | float | Player’s monetary balance | Default = 0.00 |
| IsActive | boolean | Indicates if palyer is active in current game | Default = false |

**DEALER**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUTE** | **TYPE** | **DESC.** | **CONSTRAINTS** |
| **DealerID** | **int** | **Unique dealer identifier** | **Primary key** |
| **Name** | **str** | **Dealer’s Name** | **Not null** |

**GAME**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUTE** | **TYPE** | **DESC.** | **CONSTRAINTS** |
| **GameID** | **int** | **Unique game identifier** | **Primary key** |
| **GameType** | **str** | **Type of card game being player** | **Not null (choose rules, like casino style BJ)** |
| **status** | **str** | **Currents status of the game** | **Values = pending, active, completed** |
| **StartTime** | **datatime** | **When the game session starts** | **Default: current timestamp** |

**BET**

|  |  |  |  |
| --- | --- | --- | --- |
| **ATTRIBUTE** | **TYPE** | **DESC.** | **CONSTRAINTS** |
| **betid** | **INT** | **Unique bet identifier** | **Primary key** |
| **amount** | **float** | **Monetary amount of the bet** | **Not null, > 0** |
| **betType** | **str** | **Type of bet placed** | **Double down, ante, split** |
| **isWon** | **boolean** | **Indicates if the bet was won** | **Default = null** |

**4.0 Functional Model and Description**

A description of each major software function and software interface is presented.

**4.1.** **Description of Major Functions**

Each requirement is uniquely identified.

**4.1.1 Requirement 1: Card Handling**

* Shuffle and deal cards at random.

**4.1.2 Requirement 2: Game Mechanics**

* Execute blackjack rules (hitting, splitting, staying, doubling down).

**4.1.3 Requirement 3: Betting System**

* Allow user to place bets and adjust balances.

**4.1.4 Requirement 4: Dealer Artificial Intelligence**

* Ensure dealer’s actions match real casino behavior, following set blackjack rules.

**4.1.5 Requirement 5: User Interface**

* Display visual representations of cards, bets, and outcomes for users to view/control.

**4.2 Software Interface Description**

The software interface(s) to the outside world is(are) described.

**4.2.1 External machine interfaces**

Interfaces to other machines (computers or devices) are described.

Integration with (fake) payment gateways for deposit and withdrawing.

**4.2.2 External system interfaces**

Interfaces to other systems, products or networks are described.

N/A

**4.2.3 Human interface**

An overview of any human interfaces to be designed for the software is presented.

Intuitive UI with buttons for game actions (hit, stand) and a display for cards and scores

**5.0 Restrictions, Limitations, and Constraints**

Special issues which impact the specification, design, or implementation of the software are noted here.

* Legal: Compliance with state/government gambling laws in specific regions
* Technical: Limited to web platforms (no application for desktop or mobile devices)
* Performance: Game must handle concurrent users without lag and issues
* Security: Protect users' data and currency. Detect cheating if happening and ban users for it.