

# SRS DOCUMENT

Project: Python Casino Web Application (Blackjack)

Sprint Goal: Meet basic MVP requirements, including user login/logout, casino dashboard, deposit funds, and a playable blackjack game via Flask web app.

## Part 1 - Introduction

### 1.1 Purpose

The purpose of this project is to create a Python Casino Web Application that allows users to register, log in, and play the Blackjack game through an interactive Flask-based interface. The system will manage user accounts, track balances, and store data in a JSON file to ensure persistence across all sections.

### 1.2 Scope

This software provides a complete, browser-based casino game for users. Users can

- Register and log in with a unique username and password
  - View their username and current balance on a dashboard
  - Deposit funds and track their balance
  - Play a blackjack game against an automated dealer
  - Have all progress (balance, wins, losses) saved in a JSON file.
- 

## Part 2 - Overall Description:

### 2.1 User Characteristics

The target users are both professional and casual gamblers who want a simple and reliable way to check their game performance and balance through a virtual casino.

### 2.2 General Constraints

- The system must be implemented in Python using Flask
  - All persistent data must be stored in a single JSON file (database.json)
  - The app must run locally on a Flask server
  - Each module ([user.py](#), [funds.py](#), [blackjack.py](#), [main.py](#)) must remain modular and independent
  - Passwords must remain private and not be displayed in the main text
  - The interface must show clear instructions and simple navigation
-

## **Part 3 - Specific Requirement**

### **Feature 1: User Registration, Login, & Logout System**

**User story 1:** As a forgetful gambler, I want a secure login method to store my information so that I will not lose track of my gambling history.

**Acceptance criteria:**

- When launching the app, users are prompted to create or enter a username and password.
- If credentials do not match, an error message appears, and login is denied.
- On logout, the system saves player balances and game statistics (wins/losses) to the JSON file.

### **Feature 2: Casino Lobby & Dashboard Display**

**User story 2:** As a professional and competitive gambler, I want to see a dashboard of my current balance so that I can track my progress

**Acceptance criteria:**

- The dashboard must display the gambler's username and current balance.
- The user must be able to navigate back to the Blackjack game or the deposit funds page.
- The display must automatically update when the balance changes.

### **Feature 3: Fund Deposits Tracker/Bank**

**User story 3:** As a frequent gambler, I want to be able to track my wins (and losses) digitally and automatically so that I can eventually use this money for personal purchases.

**Acceptance criteria:**

- Users can fill out a deposit form and add more money to their balance.
- Deposits must reject negative or invalid amounts.
- When a user wins, their winnings are added to the balance. And when they lose, the losses are subtracted.
- Balance is displayed and saved to a JSON file after each update.

### **Feature 4: Blackjack game**

**User story 4:**

As an avid gambler, I want to be able to run the game smoothly so that I can have the best gambling experience.

**Acceptance criteria:**

- The system must include a playable blackjack game with standard rules and an automated dealer.
- Players must be prompted to enter a valid bet amount before starting each round.
- Wins, losses, and pushes must automatically update before starting each round.
- Game results and user progress (wins & losses) are updated and persisted in the JSON file.