



4Leaf Co.

REQUIREMENTS

Jordan Quick (PM)

Frances Coronel

Calvin Chambers

Anesha Passalacqua



Table of Contents

1.0 Introduction

1.1 Objectives	4
1.2 Statement of Scope	4
1.3 Software Context	5
1.4 Chosen Approach	5

2.0 Primary Platform

2.1 Platform Benefits	6
-----------------------	---

3.0 Software Interface Description

3.1 External Machine Interfaces	6
3.2 Human Interfaces	7
3.3 Components	7
3.3.1 Data Storage	7
3.3.2 Data Verification	7

4.0 Behavioral Model Description

4.1 Overview	8
--------------	---

5.0 Constraints

5.1 Overview	8
--------------	---

6.0 Product Requirements for Application

6.1 Stakeholder's Requirements	9
--------------------------------	---

7.0 Functional Requirements



7.1 Application Capabilities	11
8.0 Tracking and Control Mechanisms	
8.1 Document Revision History	12



1.0 Introduction

4 Leaf Co. was tasked to create a program that teaches a user the correct way to play Blackjack. A Strategy table is a chart that has different card combinations that could be used in a particular scenario. The game will serve as the dealer and will play the user in a game following a blackjack strategy table. This program will be used as both a learning experience and interactive activity to provide users with a better understanding of this fun game.

1.1 Objectives

4Leaf Co's Blackjack application will provide an intermediate knowledge of the game while providing the user with a strategy table. The application will provide guidance on different hand combinations that the user can take to achieve their card score of "21". The user will be able to view their possibility of winning with a statistical analysis that evaluates their proficiency.

1.2 Statement of Scope

The scope of this project is to develop a desktop application that will provide an interactive program that teaches Blackjack. This desktop application will use a strategy table implemented using the database. The target audience of this desktop application is anyone with a desire for learning how to efficiently play Blackjack.



1.3 Software Context

Blackjack is a well know game that many individuals may not know how to correctly play. 4Leaf Co's application will contain a strategy table and provide the user feedback when a wrong move is made.

1.4 Chosen Approach

4Leaf Co. will be implementing various programming languages in creating the Blackjack application. The programming languages that will be used are SQL, and Java. The database uses SQL and is printed out using Java. The second and third programs will implement statistical algorithms to find the probability of the hand combinations using SQL Code. The second program uses the database and calculates the frequency that a hand appears out of n hands for the dealer's one card and the user's first two cards. The frequency table in the second program forms the basis for the strategy table in program three. The third program calculates the probability of an action according to the user's move for that hand. The final program will use Java, SQL, and the MySQL Workbench to teach the player/user how to play Blackjack based on the strategy table generated in Program 3. All four programs will be using the built-in console as the platform to host the desktop console application. The user will be allowed to make their own moves but if a wrong move is made, the application will show the user the correct move that should have been made.



2.0 Primary Platform


2.1 Platform Benefits

The goal of this application is to provide the user with a desktop experience as many early computer games. Desktop applications are useful because of their stationary nature, uses local storage when running, and independence from the internet. This makes maintenance issues to be fixed quicker, enables ease of use, a decrease in security threats, connectivity or speed by not having to rely on the internet, and cost.

- **Maintenance:** Installation through updates when a new patch is released.
- **Ease of use:** Can be used on any desktop computer without the use of the Internet.
- **Security:** Less likely to be susceptible to cyber threats.
- **Connectivity:** Doesn't depend on connectivity and speed of the Internet.
- **Cost Factor:** Development and maintenance cost will be lower than if it was a web application.

3.0 Software Interface Description

3.1 External Machine Interfaces



The first program will create the database and connect with the second and third programs to produce statistics on both the chances of a user winning with the starting hand and after each of the user's turns.

3.2 Human Interface

The desktop app will be executed through the console application. The console application is written in Java and can be run on any desktop computer with an installed Java SDK that can be downloaded from the Oracle website. The console application does not use the database and it is purely Java code.

3.3 Components


Below are the capabilities of the database storage and how the data will be verified with the user:

3.3.1 Data Storage

The database will compute and store n combinations for statistical analysis. The number of supported decks is 1, 2, 4, 6, and 8. For each supported deck, the database will formulate n -combinations. The two algorithms that will be implemented are one to compute the percentage of the winning hand and the probability of winning due to the remaining combinations.

3.3.2 Data Verification

The Data Verification portion of this program will check to see if the user made the correct move. If the user makes an incorrect move, the program



will show the correct (most efficient) move the player could have made by using the strategy table.

4.0 Behavioral Model and Description


4.1 Overview

The user will be able to interact with the game via the console app. The user will be able to view the optimal choice they should have taken in order to learn Blackjack every time they play. The strategy table will be implemented in the game that will give suggestions when a user makes an incorrect (least sufficient) move and will show their overall probability of winning. This application will develop a user's skill in the game of Blackjack so they can become more proficient in this fun game.

5.0 Constraints

5.1 Overview

- Each hand played will have a unit bet of one and the only exception will be if the player doubles down and then the bet will be two.
- Each player will not be able to see upcoming card combinations before a move is made.
- The player will only be allowed to bet one point. The player will only be allowed to bet two points when doubling down.
- Each player will only receive a number of cards they are allowed to



receive per move.

- The player will not be allowed to go back a move after it is played.

6.0 Product Requirements for Applications

6.1 Stakeholder's Requirements

These are the requirements we must abide by to satisfy stakeholders:

- The application must be a desktop-based platform.
- All four programs must implement gaming mechanics. Gaming mechanics are constructs of rules or methods designed for interaction with the game state to form gameplay.
- The first program must have a database.
- The application must be capable of generating the data for a number of deck sizes.
- The second program must calculate the frequency with which that hand appears out of n hands for the dealer's one card and the user's first two cards.
- The application must calculate the overall probability of a user winning for the number decks that were used.
- The application must, for each possible play (Split, Double Down, Hit, or Stand), generate the probability of winning for each possible hand.
- The application must enter a probability of 0.000 if a play is not possible.

- The third program must:
 - Calculate the probability of both splitting and not splitting if a split is possible
 - Calculate the probability of both doubling and not doubling, it is possible to double down
 - Calculate the probability of both hitting and standing, if it is possible that hitting could result in a bust
 - Use the rules that that player may double down on 9, 10, 11 and Ace
- The application must allow a Double Down or Split after a Split
- The fourth program must create a Blackjack Strategy Card calling the data from the second and third program to provide user statistics and feedback.
- The strategy table the game program uses must be editable and the user has to be able to change the strategy table.
- The Strategy table generated by the second program and used in the fourth program must have the following mnemonics:
 - P for split
 - D for double down
 - H for hit
 - S for stand

- Table rows and columns must be fixed and standardized between applications stated in the original requirements
- When 65% of the deck (or shoe) has to be used a reshuffle will occur
- The application must use the original rules that Blackjack Pays 3:2

7.0 Functional Requirements

4Leaf Co is to create a desktop application to teach users the most efficient way to play Blackjack. The application must have the following capabilities:

- Various amounts of hands must be calculated using the Blackjack strategy table for all hands used
- The program must assume the role as the dealer that will play against the user
- If a player makes a mistake, the program must notify the user of the mistake and show the correct move

7.1 Application Capabilities

- The application must generate a MySQL database of n hands
- The application must be capable of generating the data for a number of decks. The number of deck choices is 1, 2, 4, 6, and 8. Each deck size is 52.
- The application must calculate the frequency with which that hand appears out of n hands for the dealer's one card and the user's first two cards

- The application must correctly teach individuals how to play Blackjack

8.0 Tracking and Control Mechanisms

8.1 Document Revision History

Version	Implemented By	Approved By	Date	Reason
1.0	Calvin Chambers Jordan Quick Frances Coronel Anesha Passalacqua	Jordan Quick	1/21/16	Iteration 1
2.0	Calvin Chambers Anesha Passalacqua	Jordan Quick	2/11/16	Iteration 2
3.0	Calvin Chambers Anesha Passalacqua	Jordan Quick	3/3/16	Iteration 3
4.0	Calvin Chambers	Jordan Quick	3/16/2016	Iteration 4
5.0	Calvin Chambers	Jordan Quick	4/26/2016	Iteration 5