Software Requirements Specification

Sean Allred
Molly Domino
Josh Kaminsky
Matthan Lee

September 16, 2012

1 Introduction

1.1 Purpose

The goal of our endeavor is to produce an executable file that will create a one user blackjack game. The game should have many functions for the user including (but not limited to) betting functionality, personalization, [1] and a graphical user interface.

1.2 Scope

We will be creating a file that will play a game of blackjack with hit, split, and stand functionality. The scope of our project will be between this baseline and the constraints outlined below:

- No multiplayer functionality
- No other game functionality beyond blackjack
- No animations

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
SMCM	St. Mary's College of Maryland
SME	Subject Matter Expert
FA	Functional Analyst
SA	Solutions Architect
DEV	Developer
QA	Quality Assurance
Player	The user
Dealer	The computer
Hit	The move in blackjack wherein the Player draws another Card from the Deck through the Dealer.
Split	The move in blackjack wherein the Player's hand is split into two, where each of the Player's hands take half of the bet.
Stand	The move in blackjack wherein the Player ceases to draw more Cards from the Deck, ending the Round.

[1] Is this really a requirement?

1.4 References

- Iteration Plans
  None yet
- Development Case
  None yet
- Vision
  None yet
- Glossary
  None yet
- Other
  Thank you to GitHub.com for allowing our team to collaborate online.

1.5 Overview

1.5.1 Project Overview

This *Software Development Plan* provides a description of the purpose, scope, objectives and deliverables pertaining to the project.

1.5.2 Project Organization

Name	SME	FA	SA	DEV	QA
Sean Allred		✓	✓	✓	✓
Molly Domino	✓		✓	✓	✓
Joshua Kaminsky	✓	✓	✓		✓
Matthan Lee	✓	✓		✓	✓

1.5.3 Management Process

This project should cost, essentially, only time, and should take about one semester to complete. The major phases of this project are defined in sections 4.1 and 4.2.

1.5.4 Applicable plans and Guidelines

Software development team will be following an iterative waterfall model with two iterations and GUI prototyping, iterating through the design of the logical structure of the game and the graphical user interface. Tools may include Microsoft Visual Studio for C# development[2] and the computers to run these programming environments. Techniques may include round-table prototyping sessions and mock-up code demonstrations for the client.

[2]There also is consideration of using NetBeans with Java.

2 Functional Description

2.1 Language and Terminology

Planned program features will be described as "necessary", "unnecessary", or "pending"; features will also be described as "intended" or "not intended", to indicate whether or not they will be implemented. Including the following language models in the descriptions of various requirements indicates their category as necessary, unnecessary, planned, unplanned, or pending categorization.

	Intended	**Unintended**
Necessary	will	should
Pending	may	may not
Unnecessary	can	will not

Table 1: Terminology

Must — Indicates a feature that absolutely must be included in the software for the software project to be considered finished, and is intended to be included in the software package.

Should — Indicates a feature that absolutely must be included in the software for the software project to be considered finished, but is not intended to be included in the software package. Concerted effort will be made to avoid ever using this keyword.

May — Indicates a feature that or may not be necessary, pending further insight, but is intended to be included in the software package.

May not — Indicates a feature that or may not be necessary, pending further insight, but is not intended to be included in the software package.

Will — Indicates a feature that is not necessary, but is intended to be included in the software project.

Will not — Indicates a feature that is not necessary, and is not intended to be included in the software project.

2.2 User Interface

The program must implement a graphical user interface (GUI) as the primary mode of interactions with the user.

Title The program's GUI must be titled for the game being played (i.e. 'Blackjack').

Menu The GUI must contain a standardized menu bar situated at the top of the interface with the following items:

File The menu must include the standard 'File' menu option that reveals the following actions:

- **Restart**
  Starts a new game.
- **Statistics**
  Reveals, in a separate window, the Player's gameplay statistics for the current session: number of wins, number of losses, largest win, and greatest loss.
- **Exit**
  Upon confirmation, terminates the program.

Help The menu must include the standard 'Help' menu option that reveals the following actions:

- **About**
  Reveals various information about the software including, but not limited to, the title of the program, the version number, its authors, and its license.

2.3 Graphics

Program must access and display many visual resources, to include Windows graphics packages and external image files to represent objects in-game.

Card Display Graphics must include an image to represent the back of a playing card (i.e., face-down) and the front of a playing card (i.e., face-up).

Deck Display Graphics must include an image to represent more than one card in a stack or deck.

Animation Graphics will not include any animations.

Background Graphics must include an image to represent the background of the game window (i.e., the "table" on which Blackjack is being played.)

Money Display Graphics must include an actively-updated text field in which the Player's current Funds will be displayed.

Player's Name Graphics must include an actively-updated text field in which the Player's Name is displayed.

Hit Button Graphics must include a button for the user to indicate their wish to Hit.

Split Button Graphics must include a button for the user to Split.

Stand Button Graphics must include a button for the user to Stand.

Betting Box Graphics must include a betting box, which will be located in the upper left hand of the screen and will display the active bet for the hand.

Additional effects Program may not include any additional effects.

2.4 Display Screens

Start-up

GUI Initialization The screen must appears without any cards dealt. The player's name and total starting funds ($500.00) must be displayed. Buttons must remain inactive.

Session Start Program must prompt the user for their user name, and display it on the GUI. This prompt will be in the form of a pop-up window or splash screen.

Game over Program must display a pop-up window or splash screen when the game ends. It must display a message telling the user whether they won or lost.

3 System Requirements

Hardware Requirements

- 4GB RAM
- 2.0 GHz CPU (i836)

Software Requirements

- Microsoft .NET Framework 4
- Windows 7

4 Interfaces

4.1 Standalone Program

The program will neither require nor implement a network interaction of any type.

4.2 Use of Windows DLLs

The program will utilize standard and required class libraries (from Windows 7 and .NET 4), but will require no other resources not specified in System Requirements.

4.3 Use of Keyboard and Mouse

The program must take input via a keyboard and mouse.[3]

4.4 Performance

4.5 Standalone Executable

Program will be run from a standalone executable (*.exe).

4.6 System Benchmark

Program will have a minimal footprint and will operate near-instantaneously on any machine that meets the system requirements outlined in section 3.1. For best results, program should be run independently on a dedicated machine by a trained operator wearing static-resistant clothing.

5 Delivery

5.1 CD-ROM

Compiled program files will be delivered on a CD-ROM. Users must be able to run the program by double-clicking or otherwise activating the .exe file on the CD ROM.

5.2 Installation

The program will not contain any installation methods and will not be supported post-release. The user will bear the responsibility for accessing the CD-ROM and activating the executable.

6 Schedule

To be completed.[4]

7 Miscellaneous

7.1 Specificity

Blackjack! v. 1.0 by No Dice has been optimized for functionality on machine 4, lab 165, Physical Sciences Building, St. Marys College.

[3] It wouldn't be hard to add completely keyboard-based functionality to it. It'd be fun.
[4] Gantt chart?

7.2 Safety

Blackjack! v. 1.0 has not been proven to be non-carcinogenic or arsenic-free. Blackjack! v. 1.0 is pending FDA approval. The user takes full responsibility for any damages or injuries incurred by running the program while operating heavy machinery or during late-term pregnancy.