Software Requirements Specification

for

Charity Blackjack

Version 1.0 approved

Prepared by Kareem Person, Colin Kissane and Eric Leyder

Ferris Delta

March 19, 2017

Table of Contents

Table of Contents **2**

Revision History **2**

1. Introduction **3**
1.1 Purpose 3
1.2 Project Scope 3
1.3 References 3

2. Overall Description **3**
2.1 Product Perspective 3
2.2 Product Features 3
2.3 User Classes and Characteristics 4
2.4 Operating Environment 4
2.5 Design, Implementation Constraints, Assumptions and Dependencies 4
2.6 User Documentation 5

3. System Features **5**
3.1 Single Player Game 5
3.2 Player Profile Linked to GooglePlay User ID 6
3.3 Twitter Feeds of Charitable Organizations Display on Screen 6

4. External Interface Requirements **7**
4.1 User Interfaces 7
4.2 Software Interfaces 7
4.3 Communications Interfaces 8

5. Other Nonfunctional Requirements **8**
5.1 Performance Requirements 8
5.2 Safety Requirements 8
5.3 Security Requirements 8
5.4 Software Quality Attributes 9

Revision History

Name	Date	Reason For Changes	Version
Charity Blackjack SRS	03.15.17	Draft	0.1
Charity Blackjack SRS	03.19.17	Revisions and Finalization	1.0

1. Introduction

1.1 Purpose

This software requirement specification (SRS) describes the software functional and nonfunctional requirements for release 1.0 of the Charity Blackjack application. This document is intended to be used by the members of development team that will implement and verify the correct functioning of the system. Requirements as specified in section *2.1 Product Perspective* are high priority and committed for release version 1.0. All others will be implemented in subsequent releases.

1.2 Project Scope

Charity Blackjack is an Android based app. Users will be able to play BlackJack on their mobile devices while simultaneously staying up to date with selected charities. The purpose of this project is to push the abilities of the development team, while providing experience in application development. More information regarding the visions and business objectives can be found in the Charity Blackjack Vision and Scope document associated with this application.

1.3 References

REF-1: Ferris Delta, Charity Blackjack Vision and Scope; *attached with this document*
REF-2: Ferris Delta, Charity Blackjack Use Cases; *attached with this document*

2. Overall Description

2.1 Product Perspective

Blackjack charity is a first release, stand alone application for Android devices. Product features one through three, listed below in section 2.2 will be implemented in initial release of application. For more detailed information on the implementation schedule see section *3.1 Scope of Initial and Subsequent Releases* in the vision and scope document.

2.2 Product Features

FE-1: Single player offline gameplay
FE-2: Player profile links to GooglePlay profile
FE-3: Twitter feeds of charitable organizations displayed on screen
FE-4: In-game purchases that support selected charities

FE-5: Ability to link player profile to Facebook and Twitter accounts
FE-6: Leaderboards for most charitable players
FE-7: Online multiplayer; table limit of five
FE-8: Ability to create and host tournament events
FE-9: Improved newsfeeds

2.3 User Classes and Characteristics

The primary user class will be the player user. The player user will consist of individuals playing the game under typical conditions.

Implemented in future releases will be subsets of the player user, the user host and tournament host. User hosts will be individuals who are hosting online multiplayer games. Tournament host will be individuals who are hosting tournament events.

All sets of users will only need the expertise level of being able to navigate the user interface. All users will have access to all features, though restrictions will apply relating to a user host, tournament host, and the specific event they initiated. Both tournament and user hosts will need the ability to invite players to their games. Tournament hosts will have unique options setting time frames, number of players, number of rounds, etc.

2.4 Operating Environment

OE-1: The Charity Blackjack game application shall operate with mobile devices using the Android operating system versions 6.0.1 or higher.
OE-2: The Charity Blackjack server shall operate running the current versions of Ubuntu Server and Apache Web Server.
OE-3: Charity Blackjack shall use the Paypal e-commerce payment system.

2.5 Design, Implementation Constraints, Assumptions, and Dependencies

CO-1: Software shall be developed using Unity game engine.
CO-2: Software shall be written in C#.
CO-3: Software is being developed for Android version 6.0.1 and higher devices.
AS-1: User will have internet access while using the application.
AS-2: Charitable organizations will not object to being associated with a gambling themed application.
DE-1: Newsfeed and social media links will require internet access to function.
DE-2: In-app purchases are dependent on user having a paypal account.
DE-3: Use of organization logos is dependent on permission being granted.

2.6 User Documentation

UD-1: In-app tutorial explaining application features will be available from the main menu.
UD-2: In-app tutorial explaining how to play blackjack will be available from both the main menu and while playing the game.

3. System Features

3.1 Single Player Game

3.1.1 Description and Priority

Player user has the ability to play single player blackjack game against computer dealer. Up to four computer controlled players may accompany player user at table.

3.1.2 Stimulus/Response Sequences

Stimulus: Player chooses to 'hit'
Response: Card is added to user's hand
Stimulus: Player chooses to 'stand'
Response: Game moves to dealer's turn
Stimulus: Player busts
Response: Game moves to settlement phase

3.1.3 Functional Requirements

Game.Deck:	New deck list object in which to store a collection of card objects is instantiated.
Game.Deck.Shuffle:	Deck is populated with collection of card objects.
Game.Deck.Remove:	Card object is randomly selected from deck list and removed from collection.
Game.Hand:	New hand list object in which to store a collection of card objects is instantiated.
Game.Hand.Add:	Card object from Game.Deck.Remove function is added to added to hand list.
Game.Hand.Empty:	All card objects contained in hand list are removed.
Game.Hand.Remove:	Card object is selected from hand list and removed from collection.
Game.Determine.Value:	Value of each card object in each hand is summed, giving the total value per hand.
Game.Determine.Winner:	Hand values from Game.Determine.Value are compared. Winners are assigned by highest point value under 22.

Game.Purse:	New purse variable is instantiated. Purse represents total money in player's possession.
Game.Purse.Add:	Purse value is increased.
Game.Purse.Remove:	Purse value is decreased.
Game.Pot:	New pot variable is instantiated. Pot represents sum of player's bets.
Game.Pot.Add:	Pot value is increased.
Game.Purse.Remove:	Pot value is decreased.

3.2 Player Profile Linked to GooglePlay User ID

3.2.1 Description and Priority

GooglePlay account User ID is associated with player profile.

3.2.2 Stimulus/Response Sequences

Stimulus: First time application is initiated
Response: Application requests permission to use GooglePlay User ID
Stimulus: User accepts request
Response: GooglePlay account links to player profile
Stimulus: User denies request
Response: Application proceeds without user ID

3.2.3 Functional Requirements

Game.Profile.Request:	Application displays message asking user for permission to use GooglePlay user ID.
Game.Profile.Request.OK:	Application attempts to assign GooglePlay user ID to player profile.
Game.Profile.Request.Neg:	Application proceeds with no user ID.
Game.Profile.Link.Success:	Profile link succeeds. Confirmation message displayed on screen.
Game.Profile.Link.Fail:	Profile link fail. Failure message is displayed on screen. User is given option to attempt again or proceed without linking profile.

3.3 Twitter Feeds of Charitable Organizations Display On Screen

3.3.1 Description and Priority

Application displays Tweets from charitable organizations on screen.

3.3.2 Stimulus/Response Sequence

Stimulus: Twitter feed bar is empty
Response: Application seeks Tweets and displays on screen
Stimulus: Twitter feed bar has been static for sixty seconds
Response: Application seeks new Tweet and displays on screen

3.3.3 Function Requirements

Game.News:	Twitter feed display box is displayed on screen.
Game.News.Time:	Variable that stamps the time display was last updated.
Game.News.TimePass:	Sets time at one minute from Game.News.Time timestamp. Once new time is reached, new Tweet is sought to display on screen.
Game.News.Find:	Application seeks Tweets to display on screen. Seeks newest Tweets from selected charities that are not contained in the Game.News.Store list. If no Tweets newer than Tweets in Game.News.Store list are available, oldest Tweet in Game.News.Store list will be displayed.
Game.News.Display:	Found Tweet is displayed in the Game.News display box.
Game.News.Store:	Tracks the last one hundred Tweets displayed on screen to prevent over repetition.

4. External Interface Requirements

4.1 User Interfaces

UI-1: User interface will be entirely through touchscreen interface.
UI-2: Each screen will display a help button that displays relevant information for the feature of that particular screen.

4.2 Software Interfaces

SI-1: Payment System
SI-1.1: All payments shall transmit through external Paypal e-commerce system.
SI-1.2: Once payment has been verified player profile will reflect in game purchase.

SI-2: Newsfeed
SI-2.1: Newsfeed ticker bar will retrieve information from associated charity's Twitter feed. Information will be displayed unobtrusively at the bottom of the screen.

SI-3: Social Media Links
SI-3.1: Player profile can be linked to Facebook and Twitter accounts.
SI-3.2: After donations player is given option to post to either linked accounts.
SI-3.3: Automatically generated message can be used, but option to personalize is included.
SI-3.4: Players will be presented with option to send charity awareness messages through either Facebook or Twitter.

4.3 Communications Interfaces

CI-1: The system shall send an e-mail message to the player confirming the donation acceptance by thanking them along with the receipt.
CI-2: Players may post their donations and awareness messages on Facebook and Twitter.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

PE-1: Application shall be available at all times, with the exclusion of when an update is installing.
PE-2: Application shall be in ready to play state in no more than 20 seconds from initializing application.
PE-3: Loading screens shall be no longer than four seconds.
PE-4: Downloading newsfeeds shall not hinder gameplay.
PE-5: Uploading to social media shall not hinder gameplay.

5.2 Safety Requirements

SREQ-1: Username policy in place to avoid harmful inappropriate usernames. If one is reported/caught the user is forced to name change or banned. Inappropriate username list in the system to deny players from using such names.
SREQ-2: In-game emotes only, such as “Well Played”, to avoid harassment or inappropriate communication.

5.3 Security Requirements

SEREQ-1: Use of external payment system, PayPal, to protect user’s financial information.
SEREQ-2: All transactions regarding financial or personal information are protected.
SEREQ-3: Offsite database to store backup user information in case of a system failure.

5.4 Software Quality Attributes

Security-1: Any compromise of user’s personal information would greatly affect the willingness of consumers to use this application.
Robustness-1: System failure will create an undesirable experience for the user and is therefore unacceptable in this application.
Efficiency-1: Poor application efficiency can create a negative user experience on both the conscious and subconscious level. Efficiency must conform to the performance requirements set forth in section 5.1.