

Software Requirements Specification

Grand Slam Baseball All-Stars 2: Electric Boogaloo

COP4331C, Spring, 2018

Team Name: Team MLB Card Game 2

Team Members:

- Daniel Carman
- John DeSimone
- Malik Henriquez
- Ronald Marrero
- Brian Wengier
- Kaleb Yangson

Contents of this Document

Introduction

- Software to be Produced
- Applicable Standards

Definition, Acronyms, and Abbreviations

Product Overview

- Assumptions
- Stakeholders
- Event Table
- Use Case Diagram
- Use Case Descriptions

Specific Requirements

- Functional Requirements
- Interface Requirements
- Physical Environment Requirements
- Users and Human Factors Requirements
- Documentation Requirements

- Data Requirements
- Resource Requirements
- Security Requirements
- Quality Assurance Requirements (AKA nonfunctional, Quality Requirements)

Supporting Material

Section 1: Introduction

Software to be Produced:

There are two main software products that are to be produced as part of this project. More information on each deliverable is available in the rest of this document.

- Core game application - The primary deliverable is the game application itself. This application is a mobile game that will contain all of the necessary assets to run the game, except for the user data, which will be stored on the user database.
- User database - The secondary deliverable is the database that will store user data for all of the users that have created an account for the game. This data will be stored on the server to prevent players from altering their user data in malicious ways, such as editing their card collection or game statistics (such as their win/loss ratio). This data will be fetched from the server when the user logs in and will be updated to the server after each card pack acquisition, after each change to the player's teams, and after each game concludes.

Applicable Standards

- All requirements shall be listed in the same format. Each requirement will be given a unique number, description,

Definitions, Acronyms, and Abbreviations

- User - any player with an account stored on the user database.
- Active user - the user that is currently playing the game.
- User data - the data that is associated with each user. This data consists of the user's current teams, the user's win/loss ratio, the win/loss ratio for each of their currently built teams, the user's card collection, and the user's current amount of in-game currency.
- Database - the user database that houses user data for each user.

Section 2: Product Overview

Assumptions:

- Users are assumed to have an android device with at least a 1 GHz processor and 1 GB of RAM.
- Users are assumed to have an internet connection capable of the minimal transfers required to communicate with the database.
- Users are assumed to have basic 'smartphone literacy', aka the ability to operate standard smartphone applications.
- The hosting service being used for the user database is assumed to provide us with at least 99% uptime.

Stakeholders:

- Users - This class of stakeholder consists of the players that shall use the system to play the described game. The user's main interest is in the core functionality of the game, as well as its responsiveness and visual appeal.
- Admins - This class of stakeholder consists of the administrators that will oversee application and database maintenance. Their primary concerns is ease of maintainability, as well as the software's immunity to errors.
- Customer - This stakeholder is the customer that proposed the development of the application. Their main concerns are the functionality, robustness, immunity to errors, and visual appeal of the final system.

Event Table:

Event Name	External Stimuli	External Responses	Internal data and state
Startup	User launches the application	App shows Unity logo and starts up, then prompts the user to login	No internal data yet, system state goes to login page
Login	User enters valid login information, user data sent from server	Prompts the user with a welcome message, then loads the main menu	User login info sent to database, user data fetched from database, state change to main menu
Failed Login	User enters invalid login information, Null reference sent from server	Prompts the user with a message saying "Invalid email or password."	Invalid login info sent to database, no user data fetched, no state change
Main Menu	User makes a selection	User is sent to submenu depending on selection (session search if Play game is selected, scripted tutorial	User data, current state is main menu, state change depending on user selection

		game if tutorial is selected, store if card shop is selected, options menu if options is selected)	
Game Search	N/A	User is sent to a multiplayer lobby where the user will be paired to another player seeking a game (if available)	User data, opposing user data, state change to active game
Card Collection Menu	User input (scrolling, card selection)	User is shown their current card collection and can sort based on drop-down criteria	User data, state change to selected card (if applicable)
Options Menu	User makes a selection (Sound settings, network settings, help screen)	User is shown a list of options where they can change volume level, brightness, and graphic quality	User is shown options to change based on role (administrator or user)
Shop Menu	User makes a selection (which kind of card pack to purchase)	User is shown a variety of card packs and a price associated depending on the pack type	Card pack data, state change to selected pack
Start Turn	Active user presses start turn button	"TURN START!" message displayed on screen	Active user data, opposing user data, game results initialized, state change to main step
Main Step	N/A	Simulated inning will play out on screen with card graphics and animations, turn results screen will be displayed after completion, along with "Next turn" button	Active user data, opposing user data, state change to game results based on turn results
Game End	N/A	After 9 innings if score is tied, simulated audience will choose which team is the victor (based on things like showboating and home field advantage). Otherwise the team with the highest score is shown the "ALL-STAR WINNER" screen, and the other player is show the "ALL-STAR LOSER" screen	Active user data, opposing user data, state change to game results based on current game results, active user data and opposing user data updated with completed game results

Pause Menu	User presses the pause button	User is shown options menu, with additional option for "Quit game"	Active user data, opposing user data, state change depending on user input
Team Creation Menu	User selection	User is shown controls and options for creating a team based on available cards	Active user data, state change depending on user input
Concede	User selects "Quit Game" from pause menu	User is shown a confirmation page to verify requires to leave	Active user data, opposing user data, Game state changes depending on user selection, active user data and opposing user data updated to database
After Match Report	User selection	User is shown match stats and options for "Main Menu" and "New Game"(game search)	Active user data, game report, state change depending on user selection, active user data and opposing user data updated to database
Exit Application	User closes app through native phone controls	Database connections are closed, default view is set to main menu for next application use	User data updated to database, application closes

[illegible]

- Select from main menu - This use case describes the user's ability to select an option from the main menu. These options include "collection", "options", "store", "find game", and "exit".
- View collection - This use case describes the user's ability to view their current card collection. Within their collection, they will also have the option to manage their teams.
- Manage teams - This use case describes the user's ability to create and manage new teams using their currently acquired cards.
- Exit app - This use case describes the user's ability to safely exit the application from the main menu.
- Select from options menu - This use case describes the user's ability to select one of several options available in the games options menu, which is accessible from the main menu and from the game's pause screen.

- Pause game - This use case describes the active user's ability to pause the game at any point during play. Pausing the game will display the options menu.
- Mute game - This use case describes the user's ability to mute all game audio.
- View help - This use case describes the user's ability to view the provided in-game help documentation, which is written to provide new users with a basic understanding of the game's mechanics.
- View store - This use case describes the user's ability to view the card pack shop, which will show the user a list of the types of packs currently available for purchase with their in-game currency.
- Purchase packs - This use case extends the functionality of view store, allowing users to select an available card pack and purchase it with their in-game currency.
- Create account - This use case describes the system's ability to allow a new user to create an account.
- Login - This use case describes the user's ability to enter their username and password in order to log in to the game.
- Login failure - This use case extends the functionality of the login use case, allowing the system to handle situations where the user provides invalid login credentials.
- Search for game - This use case describes the user's ability to search the user database for an opposing team.
- Generate team if none can be found - This use case extends the functionality of search for game, allowing the system to handle situations where no opposing team can be found.
- Play game - This use case describes the user's ability to play through a full simulated game of baseball with their team. (need to add quit game use case)
- Show end results - This use case describes the system's ability to provide the user with an end of game statistics screen, as well as update the user database with these statistics.
- NOTE: The role of the Admin actor is to oversee maintenance of the system and the database.

Section 3: Specific Requirements

3.1 Functional Requirements

No: 1

Statement: The system shall allow the user to open the application and login with their username and password
Source: Events Table
Dependency: Requirement 1.1, 2
Conflicts: None
Supporting Materials: Events Table, UML Use Case diagram (above)
Evaluation Method: A successful login will return user data and allow the user to enter the game's main menu
Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 1.1
Statement: The system shall prompt the user to login again if a login attempt fails to fetch user data
Source: Events Table
Dependency: None
Conflicts: None
Supporting Materials: UML Use Case Diagram 1 (above)
Evaluation Method: If a login attempt fails, the user will be prompted with a message informing them of the failure and asking them to try again
Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 2
Statement: The system shall allow a user to make a new account on the login page
Source: Requirement 1

Dependency: None
Conflicts: Requirement 1 (if the chosen account information is already taken)
Supporting Materials: UML Use Case diagram
Evaluation Method: If the account is successfully created, its information will be available on the user database
Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 3
Statement: The system shall allow the user to search for a new game
Source: Events Table, UML Use Case Diagram (above)
Dependency: Requirements 5
Conflicts: Requirement 3.1
Supporting Materials: None
Evaluation Method: This requirement shall be met if the game search function successfully places the active user into a game
Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 3.1
Statement: If a game cannot be found, the system shall generate a team for the active user to play against
Source: Requirement 3
Dependency: None
Conflicts: None

Supporting Materials: None
Evaluation Method: A game will be searched for when the database has no user data besides that of the active user to see if a team is properly generated
Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 4
Statement: The system shall use the active user's team stats and the other user's team stats to simulate a game of baseball
Source: Events Table
Dependency: Requirements 3, 3.1, 4.1, 4.2
Conflicts: None
Supporting Materials: UML Use Case Diagram
Evaluation Method: This requirement is met if the completed system can play through an entire game from the starting turn until the game end screen is shown
Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 4.1
Statement: The system shall allow the active user to control the flow of the game by letting them start each turn (inning)
Source: Events table
Dependency: None
Conflicts: None
Supporting Materials: UML Use Case Diagram

Evaluation Method: This requirement is met if the active user is able to make the turns go by at their leisure through use of the next turn button

Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 4.2

Statement: The system shall end the game and tally up the game statistics after the 9th turn (inning)

Source: Events Table

Dependency: None

Conflicts: None

Supporting Materials: Data requirements

Evaluation Method: This requirement will be satisfied if the end game statistics screen matches up with manual, by hand calculations done for a single game

Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 4.3

Statement: The system shall allow the active user to concede a game they are currently playing, resulting in a loss for them and a perfect win for the other use (a win as if the other user had no opposition)

Source: Events Table

Dependency: None

Conflicts: None

Supporting Materials: Data requirements, UML Use Case diagram

Evaluation Method: This requirement will be met if the active user's data is updated to result a total defeat in the user database and the other user's data is updated to reflect a perfect win in the user database

Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 5

Statement: The system shall allow the user to view their current card collection
--

Source: Events Table, Project Management Plan

Dependency: None

Conflicts: None

Supporting Materials: UML Use Case diagram
--

Evaluation Method: This requirement shall be met if the displayed collection matches up perfectly with the user's card collection data in the user database

Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 6

Statement: The system shall allow users to create and manage their teams with their currently collected cards

Source: <source of the requirement>

Dependency: Requirement 5

Conflicts: None

Supporting Materials: UML Use Case diagram, UML Class diagram

Evaluation Method: This requirement shall be met if users can create, edit, and delete their teams and have these changes show up in the user database
--

Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 7
Statement: The system shall allow users to view packs available for purchase and to purchase said card packs using their acquired in game currency
Source: Events Table
Dependency: None
Conflicts: None
Supporting Materials: UML Use Case diagram, UML Class diagram
Evaluation Method: This requirement shall be met if users are able to purchase a pack and have the newly acquired cards show up in the user database
Revision History: Daniel Carman, 2/24/2018, ver. 01

No: 8
Statement: The system shall allow the user to exit the application via an option on the main menu
Source: Events Table
Dependency: None
Conflicts: None
Supporting Materials: UML Use Case diagram
Evaluation Method: This requirement shall be met if
Revision History: Daniel Carman, 2/24/2018, ver. 01

3.2 Interface Requirements

No: 9
Statement: The system shall retrieve user information with synchronous calls.

Source: Events table, UML Use Case Diagram
Dependency: Requirement 1
Conflicts: None
Supporting Materials: Data requirements
Evaluation Method: The system will output correct user information.
Revision History: Ronald Marrero, 2/24/2018, ver. 01

No: 9.1
Statement: The system shall retrieve team information with synchronous calls.
Source: Events table, UML Use Case Diagram
Dependency: Requirement 4
Conflicts: None
Supporting Materials: Data requirements
Evaluation Method: The system will output correct and current team stats within a range of [0%, 100%].
Revision History: Ronald Marrero, 2/24/2018, ver. 01

No: 9.2
Statement: The system shall retrieve player scores with asynchronous calls every 30 seconds.
Source: Events table, UML Use Case Diagram
Dependency: Requirement 4
Conflicts: None
Supporting Materials: Data requirements

Evaluation Method: The system will reflect successful updates to all player scores within a positive range of integers.

Revision History: Ronald Marrero, 2/24/2018, ver. 01

No: 9.3

Statement: The system shall retrieve store information with synchronous calls.

Source: Events table, UML Use Case Diagram

Dependency: Requirement 7

Conflicts: None

Supporting Materials: Data requirements

Evaluation Method: The system will output all current store items and will accept successful input from the user on purchases. The price will also be within a positive range of floats where accuracy is crucial up to the hundredths place.

Revision History: Ronald Marrero, 2/24/2018, ver. 01

No: 9.4

Statement: The system shall retrieve active game information with asynchronous calls at 5 second intervals in an active game until game completion.

Source: Events table, UML Use Case Diagram

Dependency: Requirement 4

Conflicts: None

Supporting Materials: Data requirements

Evaluation Method: A success message from the database call will indicate that this requirement is satisfied.

Revision History: Ronald Marrero, 2/24/2018, ver. 01
--

3.3 Physical Environment Requirements

No: 10

Statement: The system shall only run on android devices running Android 4.4 and above.
--

Source: Startup

Dependency: None

Conflicts: None

Supporting Materials: Events Table

Evaluation Method: A successful installation of the application will show that the application succeeded on supported hardware.

Revision History: Ronald Marrero, 2/24/2018, ver. 01
--

No: 11

Statement: The system shall only run on devices with an active data connection.

Source: Startup

Dependency: Requirement 10

Conflicts: None

Supporting Materials: Events Table

Evaluation Method: The application will switch between menus successfully if there is an active data connection.
--

Revision History: Ronald Marrero, 2/24/2018, ver. 01
--

3.4 User and Human Factors Requirements

No: 12
Statement: Any user shall be able to adjust the text size in their option menu
Source: Options Menu
Dependency: None
Conflicts: None
Supporting Materials: Project Management Plan
Evaluation Method: During the testing phase, we will have a user try and adjust the text size in the option menu.
Revision History: Malik Henriquez, 2/24/2018, ver.01

No: 13
Statement: Any user shall be able to retrieve a forgotten username and/or password
Source: Failed Login event
Dependency: None
Conflicts: None
Supporting Materials:
Evaluation Method: During the testing phase, we will have a user try and retrieve their username and password
Revision History: Malik Henriquez, 2/24/2018, ver.01

3.5 Documentation Requirements

No: 14

Statement: The users shall be able to view a manual containing the game features and rules
Source: Events Table
Dependency: None
Conflicts: None
Supporting Materials: Project Management Plan
Evaluation Method: An inexperienced user will evaluate the manual for aiding in learning gameplay
Revision History: Malik Henriquez, 2/24/2018, ver. 01

No: 15
Statement: Future software developers shall be able to view information in regards to managing and updating the system.
Source: PMP
Dependency: None
Conflicts: None
Supporting Materials: PMP
Evaluation Method: Software Developers will review document and evaluate on understanding and detail. It is assumed that they know the basics of software development.
Revision History: Malik Henriquez, 2/24/2018, ver. 01

No: 16
Statement: All documentation shall be available to future developers via a central documentation repository
Source: PMP

Dependency: None
Conflicts: None
Supporting Materials: PMP
Evaluation Method: The existence of this repository is sufficient for this requirement evaluation
Revision History: Malik Henriquez, 2/24/2018, ver. 01

3.6 Data Requirements

No: 17
Statement: The user shall be able to start a new multiplayer game where the winner will be determined by calculations.
Source: Game Search Event
Dependency: Requirement 4
Conflicts: None
Supporting Materials: UML Use Case Diagram
Evaluation Method: A successful pairing will occur and the active user will be entered into a game with another user.
Revision History: Ronald Marrero, 2/24/2018, ver. 01

No: 17.1
Statement: The system shall gather a list of each player's stats for each team.
Source: Start Turn event
Dependency: Requirement 17
Conflicts: None

Supporting Materials: Events Table, UML Use Case diagram
Evaluation Method: A success message from the database call will indicate that this requirement is satisfied.
Revision History: Ronald Marrero, 2/24/2018, ver. 01

No: 17.2
Statement: The system shall make data calculations to determine successful actions.
Source: Start Turn event
Dependency: Requirement 17.1
Conflicts: None
Supporting Materials: UML Use Case Diagram
Evaluation Method: A successful change of game state by both players will indicate success.
Revision History: Ronald Marrero, 2/24/2018, ver. 01

No: 17.3
Statement: In the event of a tie, the system shall determine the winner based on a calculation.
Source: Events Table
Dependency: Requirement 17
Conflicts: Requirement 18
Supporting Materials: UML Use Case Diagram
Evaluation Method: A valid winner in the case of a tied game after 9 turns (innings) will indicate success.

Revision History: Kaleb Yangson, 02/24/2018, ver. 01
--

No: 18

Statement: The system shall declare a winner and update both players' win/loss ratio in the database.

Source: Events Table

Dependency: Requirement 17

Conflicts: Requirement 17.3

Supporting Materials: Events Table, UML Use Case Diagram
--

Evaluation Method: A success message from the database call will indicate that this requirement is satisfied.

Revision History: Kaleb Yangson, 02/24/2018, ver. 01
--

3.7 Resource Requirements

No: 19

Statement: The user database shall be built by 6 individuals with some database experience
--

Source: PMP

Dependency: None

Conflicts: None

Supporting Materials: PMP

Evaluation Method: The functioning user database will be sufficient for the evaluation of this requirement
--

Revision History: Brian Wengier, 02/24/2018, ver. 01
--

No: 20
Statement: The core game application shall be built by 6 individuals using the Unity engine, C#, and Visual Studios
Source: PMP
Dependency: None
Conflicts: None
Supporting Materials: PMP
Evaluation Method: The functioning game will be sufficient for the evaluation of this requirement
Revision History: Brian Wengier, 02/24/2018, ver. 01

No: 21
Statement: Product testing shall be performed on Android emulators and Android phones
Source: PMP
Dependency: None
Conflicts: None
Supporting Materials:PMP
Evaluation Method: During the testing phase, logs will be kept to prove testing was done on Android emulators and phones
Revision History: Brian Wengier, 02/24/2018, ver. 01

3.8 Security Requirements

No: 22

Statement: The system shall be accessed only by authenticated users.
Source: Events Table
Dependency: Access to the rest of the system beyond the login page
Conflicts: None
Supporting Materials: UML Use Case diagram
Evaluation Method: The system satisfies this requirement when a user cannot access the system until they enter their username and password
Revision History: Brian Wengier, 02/24/2018, ver. 01

No: 23
Statement: The system shall use cryptographic protocols such as SSL and HTTPS for network communications
Source: PMP
Dependency: None
Conflicts: None
Supporting Materials: UML Use Case Diagram
Evaluation Method: The use of these protocols is sufficient for the evaluation of this requirement
Revision History: Brian Wengier, 02/24/2018, ver. 01

No: 24
Statement: The system shall end the session automatically when an open session is not used for 10 minutes
Source: Login event

Dependency: None
Conflicts: None
Supporting Materials: UML Use Case Diagram
Evaluation Method: The system satisfies this requirement when the user is logged out after being inactive for 10 minutes
Revision History: Brian Wengier, 02/24/2018, ver. 01

3.9 Quality Assurance Requirements (AKA nonfunctional, Quality Requirements)

No: 25
Statement: The system shall have an uptime of 99%
Source: User's needs
Dependency: None
Conflicts: 28
Supporting Materials: PMP
Evaluation Method: Logs of uptime/downtime history
Revision History: Kaleb Yangson, 02/24/2018

No: 26
Statement: The system shall present information to the user with a visually pleasing interface
Source: User's needs
Dependency: None
Conflicts: None
Supporting Materials: PMP
Evaluation Method: Game testers, customer feedback
Revision History: Kaleb Yangson, 02/24/2018

No: 27
Statement: The system shall be available to both customers and admins 24/7
Source: Requirement 25

Dependency: 25
Conflicts: 28
Supporting Materials: PMP
Evaluation Method: Logs of uptime/downtime history
Revision History: Kaleb Yangson, 02/24/2018

No: 28
Statement: The system shall be restarted within 5 minutes of system failure
Source: Requirement 25
Dependency: None
Conflicts: Requirements 25, 27
Supporting Materials: PMP
Evaluation Method: Logs of uptime/downtime history
Revision History: Kaleb Yangson, 02/24/2018

No: 29
Statement: The system shall alert an admin via email when the system has a failure.
Source: Requirement 28
Dependency: Requirement 28
Conflicts: None
Supporting Materials: PMP
Evaluation Method: Testing the system by simulating a failure and checking for an email.

Revision History: Kaleb Yangson, 02/24/2018

No: 30

Statement: The system shall protect customer's information using SSP and HTTPS
--

Source: Requirement 23

Dependency: Requirement 23

Conflicts: None

Supporting Materials: PMP, UML Use Case Diagram

Evaluation Method: The use of these protocols is sufficient for the evaluation of this requirement
--

Revision History: Kaleb Yangson, 02/24/2018

Section 4: Supporting Material

