Lab 2 – War of Brawns Product Specification

Team Bronze

Old Dominion University

CS 411W

Professor J. Brunelle

4 April 2020

Final Version 2

Table of Contents

3. Specific Requirements

3.1. Functional Requirements

3.1.1. **User Interface/User Experience.** User Interface/User Experience consists of multiple components that make up the interaction between the gamer and War of Brawns.

3.1.1.1. **Account Creation Page.** The Account Creation page is the display of a page that gamers go to create an account. The Account Creation Page should achieve this objective: (O: April)

As a gamer, I want to see the account creation page.

3.1.1.1.1. **Display Account Creation Page.** Provide a page that displays a sample of an account creation form that meet these requirements:

- The page must be accessible from a “create account” link located on the Login Page.
- The page must display a blank and non-interactive form that asks for: Name, Username, Birthday, Email, and Password.
- The page must have a way to go back to the Login Page.

3.1.1.2. **Login Page.** The Login page is the page where gamers use their username and password to login to their personal account. The Login Page should achieve the following objective: (O: April)

As a gamer, I need to be able to login to my account.

3.1.1.2.1. **Allow Gamer Login.** Provide a page that allows login options for gamers that are already registered. The Login Page should meet these requirements:

- Must display a form asking for username and password.
- Must display a “Create Account” link that accesses the Account Creation Page.
- Must display a “forgot password” option.
- Must not allow Login if username and password are incorrect.
- Must display notification if the username and password are incorrect.

3.1.1.3. **Navigation.** Provide a way for the gamer to navigate throughout the application. (O: April)

As a gamer I need to be able to navigate to different pages within the application.

3.1.1.3.1. **Navigation Bar.** The Navigation Bar is a static bar on the bottom of every page that provides access to the Home, Options, Challenges, Weight Progression, Diet Journal, and Game Play pages

- Must be displayed at the bottom of every page except the gameplay page.
- Must have links to the Home, Options, Challenges, Weight Progression, Diet Journal, and Game Play pages.

3.1.1.4. **Home Page.** The Home page is the page that is seen when the gamer logs in.(O: April)

As a gamer, I need to be able to access a homepage to navigate to different pages within the application.

As a gamer, I need to be able to access the home page.

As a gamer, I need to see my avatar’s STATS.

3.1.1.4.1. **Home Page Display.** Provide a page that displays selected information about the gamer's account. The Home Page should meet these requirements:

- Must display Username
- Must display Avatar STATS
- Must display the Default Avatar

3.1.1.5. **Options Page.** The Options Page is the page that allows the gamer to customize information within their account. (O: April)

As a gamer, I need to be able to view the options page and update current weight.

As a gamer, I want to see the default avatar on the avatar creation page.

3.1.1.5.1. **Options Page Display.** Provide a page that allows the gamer to view an example of a settings page. The Options Page should achieve these following requirements:

- Must display functional links to the following child pages: Customize Avatar, Edit Profile, and Goals.
- Must display non-functional links to Account Management, and External Devices.

3.1.1.5.2. **Edit Profile.** The Edit Profile child page is where the gamer can view their profile information. The Edit Profile page should meet these requirements:

- Must display Username, gender, date of birth, and units.

3.1.1.5.3. **Goals.** The Goals child page is where the gamer can view and change their goals. The Goals page should meet these requirements:

- Must display Starting weight, current weight, goal weight and weekly goal weight.
- Must provide a way to change the gamer's current weight.

3.1.1.5.4. **Avatar Creation.** The Avatar Creation page is the page that displays the default avatar. The Avatar Creation page should meet the these requirements: (O: April)

- Must display default avatar with the default parameters that were initially setup with the prototype.
- Must display a functional Navigation Bar to provide access to additional pages within the application.

3.1.1.6. **Challenges Page.** The Challenges Page is the page where the gamer can search, track and complete workout challenges. (O: Rodolfo, M1: April)

As a gamer, I need to be able to access the Challenges page.

As a gamer, I need to be able to select specific and compiled workout challenges.

As a gamer, I need to be able to cancel the workout challenge while in progress.

As a gamer, I need to receive experience points in certain stats after completing workout

3.1.1.6.1. **Challenges Page Display.** Provide a page that allows the gamer to add exercises to their challenges.

- Must allow gamers to add exercises to their challenge lists.
- Must display all available workout challenges.

3.1.1.6.2. **Gamer's Challenges List.** The gamer's challenges list is the list the gamer makes by adding exercises.

- Must display a compiled list of exercises the gamer selected.
- Must allow gamers to add exercises to a challenge.
- Must allow gamers to select exercises to perform.
- Must display the status of when exercises are completed within a selected challenge.
- Must convert completion of challenges into experience points based on workout challenges.
- Must keep total amount of experience points per stat updated in database as challenges are completed

3.1.1.6.3. **Performing an Exercise Interface.** The performing an exercise interface is what is displayed when the gamer is performing a selected exercise.

- Must provide capability to start and stop tracking an exercise.
- Must display the rewarded amount of experience points for stats after completion of an exercise.

3.1.1.7. **Weight Progression.** The Weight Progression Page is the page where the gamer's weight loss is measured and displayed through a line graph. (O: Rodolfo, M1: April)

As a gamer, I need to be able to see a line graph showing my weight progression.

3.1.1.7.1. **Weight Progression Page.** Provides a display of the gamer's weight progression.

- Must display data of weight progression for all weight measurements received.
- Data must be displayed through a line graph.

- Must display summary of data in text below graph.

3.1.1.8. **Diet Journal Page.** The Diet Journal Page is a page that displays suggested daily caloric intake and allows gamers to input their calorie intake in a meal log. (O: Rodolfo, M1: April, M2: Rodolfo)

As a gamer, I need to be able to track my caloric intake so that I can manage my dietary habits.

As a gamer, I need to be able to access the diet journal data.

As a gamer, I need to be able to create a custom meal entry.

3.1.1.8.1. **Input Display.** Provides capability to track my specific caloric intake through inputting meals in the meal log

- Display suggested daily caloric intake based on input data from the gamer.
- Display the remaining amount of calories for each day.
- Provide capability for gamer's to log multiple meals.
- Must allow gamers to log a food item by manually entering in a custom entry.

3.1.1.8.2. **Simulated Nutritional Database.** Provides nutritional information on selected food items.

- Must allow gamers to log a food item by searching the simulate nutritional database.

3.1.1.9. **In-Game Menu.** The In-Game Menu is the displayed menu during gameplay. The menu displays the inventory, weaponry, armory and avatar of the gamer account.. (O: Rodolfo, M1: April)

As a gamer, I need to be able to see the gaming HUD.

3.1.1.9.1. **Inventory.** The inventory is accessed through the Game Menu that shows all of the gamer's items.

- Must display items in inventory.

3.1.1.9.2. **Weaponry.** The weaponry is accessed through the Game Menu that shows all of the gamer's weapons.

- Displays the avatar holding the current weaponry.
- Displays all weaponry from the gamer's inventory.

3.1.1.9.3. **Armory.** The Armory is accessed through the Game Menu that shows all of the gamer's armory.

- Displays the avatar wearing the current armor.
- Displays all armor from the gamer's inventory.

3.1.1.9.4. **Gaming Head-Up Display (HUD).** The Gaming HUD is the gaming interface that shows the avatar's vital statistics and movement controls.

- Must display avatar's health
- Must update the avatar's health as it engages in gameplay.
- Must display control to attack and defend.
- Must allow character movement.

3.1.2. **Gameplay.** Gameplay is where the gamer controls their avatar to enter dungeons and battle enemies.

3.1.2.1. **Combat System.** The combat system is a real-time system where the gamer battles enemies using their avatar in dungeons. (O: Joa)

As a developer, I need to design an interactive RPG gameplay system.

As a developer, I need to develop game avatar animations.

As a developer, I need to develop non-gamer animations.

As a gamer, I need to be able to progress in-game through my character.

3.1.2.1.1. **Dungeons.** Dungeons are the core gameplay scene where combat takes place. Dungeons are 2D environments.

- Dungeons must be initiated by the gamer when selecting “tower” from the gameplay menu.
- Dungeons must contain the gamer avatar and several default enemies when initialized.
- The dungeon must fail and end when the avatar’s health reaches 0.
- The dungeon must complete and end when every enemy’s health reaches 0.
- Dungeon must fail and end when the gamer exits the dungeon via menu.

3.1.2.1.2. **Combat.** The gamer avatar and all enemies have a unique resource called health which is initialized at a predetermined value based on the character’s health level.

- Health must be decreased when struck by an attack.
- Once a character’s health reaches 0, the character must be unable to perform actions for the remainder of the dungeon instance.

3.1.2.1.3. **Progression.** In-game progression occurs when the gamer avatar becomes more effective in combat due to increased stats. Stats are increased by completing workout challenges.

- Stats are strength, endurance, and health.
    - Strength increases damage dealt by the avatar's attacks.
    - Endurance increases the amount of total energy for the avatar.
    - Health increases the total amount of health for the avatar.

3.1.2.2. **Gaming Controls.** Gaming controls are on-screen interactable controls that enable the gamer avatar to move and perform actions while in a dungeon. (O: Joa)

As a gamer, I need to interact with gameplay through on screen controls.

3.1.2.2.1. **Movement.** The gamer avatar must be able to move horizontally and jump.

- Game must display on-screen controls that enable movement.

3.1.2.2.2. **Actions.** The gamer avatar must be able to perform the following actions:

- Attack in the direction the avatar is facing.
- Dodge/Block.

3.1.2.3. **Enemy Artificial Intelligence (AI).** The Enemy AI determines enemy behavior in dungeons. (O: Joa)

As a developer, I want to design a simple RPG enemy AI to demonstrate combat.

3.1.2.3.1. **Movement.** An enemy must continually move horizontally toward the gamer avatar until in range to attack.

- If the enemy is not facing the avatar, the enemy must face the avatar.
- If the avatar is not within a predetermined range, the enemy must move toward the gamer.

3.1.2.3.2. **Attack.** The enemy will initiate attack actions facing the gamer avatar.

- If the avatar is within a predetermined attack range from the enemy, the enemy must attack.

3.1.2.3.3. **Defeat.** Enemy is defeated when enemy health reaches zero.

- When struck by a gamer avatar's attack, enemy health must be decreased by a predetermined value.
- If enemy health falls to zero, the enemy must cease all action.

3.1.2.4. **Sound and Music.** Allows the playback of sound and music. (O: Joa)

As a developer, I need to include sound effects and music to augment the gameplay.

3.1.2.4.1. **Scene music.** Music is played depending on the active scene.

- Background music must play during menu navigation and gameplay.

3.1.2.4.2. **Sound effects.** Allows for sound effects when certain actions are made.

- Sound effects must play during corresponding animation triggers.

3.1.3. **Diet Tracking.** Diet tracking consists of multiple components that help the gamer to track their diet to reach their goal weight.

3.1.3.1. **Daily Caloric Intake.** The daily caloric intake is a component of the diet journal page. It uses the daily caloric intake algorithm to calculate the daily suggested caloric intake for the gamer to meet their goal weight. The component should achieve the following objectives: (O: Kevin)

As a developer, I need to be able to use the gamer's gender, current weight, height, age, activity level, and goal weight to calculate the recommended daily caloric intake.

3.1.3.1.1. **Daily Caloric Intake Algorithm.** This is an algorithm that calculates a daily suggested caloric intake for the gamer. This will use the gamer's personal information from the database. The diet journal will use this algorithm to show the game whether they are meeting the suggested caloric intake. The following functional requirements must be met:

- Must provide the capability to allow the gamer to see their current caloric intake versus the suggested caloric intake.
- The algorithm must query the database to retrieve the following information: gender, current weight, height, age, activity level, and goal weight.
- An error message should be displayed if access to the database fails. The error message should tell the gamer to wait and try later.
- An error message should be displayed if the gamer has not entered the required information to calculate the caloric intake. The error message should tell the gamer to go back and input any of the following missing personal information: gender, current weight, height, age, activity level, and goal weight.
- The algorithm must use the gamer's gender, current weight, height, age, activity level, and goal weight to calculate the suggested caloric intake and store it onto the database.
- The algorithm result should be retrieved from the database to be displayed in the diet journal.

3.1.3.2. **Custom meal entry.** Caloric intake entry form that will store the name and the calorie amount of the custom meals in the database. (O:Roshan)

As a gamer, I need to be able to create a custom meal entry.

3.1.3.2.1. **Custom meal entry form.** Form where the gamer can create a meal and save it in the database. The custom meal creation form should meet these requirements:

- Gamer's must provide a name for the meal.
- Gamer's must provide the total calories of the meal.

3.1.3.3. **Tracking diet journal data.** Tracks the intake of the gamer for all meals for each day in the diet journal page and database. (O:Roshan)

As a gamer, I need to be able to access and track the diet journal data.

3.1.3.3.1. **Tracking diet journal data.** Diet tracking will create a new database record for daily calorie tracking and will be fetched on the diet journal page. These requirements should be met:

- Calories submitted in the meal log on the diet journal page must be incremented on the database through an update query.
- Daily total calories must be fetched from the database for the gamer.

3.1.3.4. **Usage of nutritional database for food data in diet journal.** Nutritional information of keyword searched food items will be displayed from the nutritional database on the diet journal page. The nutritional information will be displayed from this database to the diet journal page. (O:Roshan)

As a developer, I need the gamer to be able to search or select food from the fake data in the database so that they can have nutritional information about their meals.

3.1.3.4.1. **Nutritional information of food data.** Nutrition data will be retrieved from the nutritional database for the selected food. These requirements should be met:

- The gamer is able to search for meals on the emulation of the external database by clicking on the "Search" button while logging their meals on the diet journal page.
- Gamers must be able to enter the meal name, meal names matched with the gamer's entry must be retrieved from the database.
- The matched meal names with their calorie data must be shown as a list which a gamer can select to add on to their meal log.
- Food names that are not found in the database must return an alert "Item not found, Please add a custom meal".

3.1.4. **Workout tracking.** Workout tracking consists of multiple components that track the progress and results of the exercises that the gamer completes.

3.1.4.1. **Verification Interpretation.** The verification interpretation component allows the game to verify physical activity is being done. The component should achieve the following objectives: (O: Kevin)

As a developer, I need to verify the active performance of the simulated gamer with the simulated ECG data.

3.1.4.1.1. **Verification Interpretation Algorithm.** This is an algorithm that verifies physical activity is being done from the simulated ECG data. The ECG data is used to track the gamer's heart rate while the workout challenge is being done. The workout challenges page will use this algorithm to ensure workout challenges are being actively performed. The verification result of the workout will be returned to the same page. The following functional requirements must be met:

- Must provide the capability to allow the gamer to gain experience points for individual gamer profile stats after completion of the workout challenge.
- Must provide the capability to allow the gamer to pause the process of the verification interpretation.
- Must provide the capability to allow the gamer to stop the process of the verification interpretation.
- An error message should be displayed if access to the simulated data fails while in progress. The error message should tell the gamer progress will be saved until the connection is restored.
- The algorithm must query the ECG simulated data from the ECG file to retrieve the heart rate data.
- An error message should be displayed if initial access to the ECG simulated data fails. The error message must ask the gamer to check the network connection.

- An error message should be displayed if there is no ECG simulated data. The error message should tell the gamer they can continue the challenge with a penalty against the accumulation of experience points.
- The algorithm must calculate the average of heart rates per second from the ECG simulated data from the ECG file against the minimum heart rate required for the physical activity of the specified workout.
- The algorithm should display the heart rate of the gamer on workout challenges.

3.1.4.1.2. **Simulated Smartwatch (heart rate)ECG data.** This is the ECG data that will be inputted into the verification interpretation algorithm. The workout challenges page and verification interpretation algorithm will use this data. The following functional requirements must be met:

- Must provide the capability to allow the gamer verify physical activity is being done.
- Must input the data in ECG file into the verification interpretation algorithm to verify physical activity.

3.1.4.2. **Post-Exercise Experience Points.** The post-exercise experience points component allows the game to reward gamers with experience points upon completion of workout challenges. The component should achieve the following objectives: (O: Kevin)

As a gamer, I need to receive experience points in certain stats after completing workout challenges to progress my character.

3.1.4.2.1. **Post-Exercise Experience Points Distribution.** This is a distribution that rewards experience points upon completion of workout challenges. The experience points are used to increase gamer stats such as strength. The workout challenge page will use this distribution to reward gamers. The distribution result can be seen on the workout challenge page. The following functional requirements must be met: (O: Kevin)

- Must provide the capability to allow the gamer to see the amount of experience points they are rewarded for each stat.
- The distribution must query the database to add the rewarded experience points to the gamers stat total.
- An error message should be displayed if access to the database fails initially or in progress. The error message should tell the gamer that they must restart the workout challenge.
- The increase of experience points will be based on a hardcoded distribution for each stat based on the exercises they complete.
- The distribution result should be displayed in the workout challenge page.

3.1.5. **Database.** A database is a necessary functional requirement for the application, as all information regarding Gamer accounts, in-game progress, inventory, diet tracking, and workout tracking will be stored in a Firebase API.

3.1.5.1. **Information Storage.**

3.1.5.1.1. **Gamer Credentials.** Credentials specific to the Gamer must be saved to the Firebase database in order for them to be authenticated upon logging in. The following information will be included:(O: Mohammad, M1: Tim)

- Username - Provides an identification tag of the gamer's choice to specify what the application will call the gamer.
- Email - The gamer's email address that they use to sign in to the application using Google authentication in most cases.
- Phone - The phone number linked to the gamer which is registered to their smartphone. Must provide proof that the gamer's exercise is linked to a valid device.

3.1.5.1.2. **Personal Information.** Personal information about the gamer must be saved to the Firebase database. These details are used to calculate the daily suggested caloric intake. This information will already be made into two accounts using preset data. The following information is included:(O: Mohammad, M1: Tim, M2: Kevin)

- Age - The gamer's age is used in calculating daily caloric intake suggestions.
- Gender - The gamer's gender used in calculating daily caloric intake suggestions.
- Current Weight - The gamer's weight used in calculating daily caloric intake suggestions. Must also be used as a tracking mechanism for the gamer's progress toward their goal weight.

- Height - The gamer's height used in calculating daily caloric intake suggestions.
- Goal Weight - must be used as the indicator of the gamer's progress as the end result of the gamer using the application.
- Activity Level - The gamer's activity level used in calculating daily caloric intake suggestions. This will be used as a multiplier based on the activity level.

3.1.5.1.3. **Avatar Stats.** The gameplay statistics of a gamer's avatar as they progress in-game must be saved to the Firebase database. This will include the actual number for the stat and the experience point total per stat. The avatar stats will increase based on the progress of the experience point total using milestones. The increase of experience points will be based on which exercises the gamer completed in workout challenges. The following information is included:(O: Mohammad, M1: Tim, M2: Kevin)

- Strength - A non-zero number that must be used to determine the quantity of damage the gamer does to in-game enemies.
- Defense - A non-zero number that must be used to determine the damage reduction coefficient applied to the gamer while engaged in combat.
    - The damage reduction coefficient will be the percentage amount of enemy base damage taken to the gamer's health stat. This is determined based on the amount of defensive stats acquired.

- Endurance - A number that must be used to determine how often the gamer can attack enemies while engaged in combat.
- Health - A non-zero number that must be used to determine how many health points (HP) the gamer has allocated to their avatar while engaged in combat.
- Level - A non-zero number which is an indicator of how far the gamer has progressed in-game. Determined from the value of the gamer's stats and in-game experience points.
- Experience Points - A number that is used to reflect the gamer's progression upon completing challenges. These points will be allocated to the strength, defense, endurance, or health stats based on the exercises the gamer completes.

3.1.5.2. **Progress Tracking.** Progress tracking will update database records of each Gamer for the following components: weight and caloric intake. (O: Mohammad, M1: Tim)

- Must provide information to display to the gamer to visualize their weight change while using the application.
- Values must be updated in the database each time the gamer updates their weight within the application.

3.1.5.3. **Nutritional Database.** A Firebase database will be created based on the EHSA Nutrition Database API, consisting of nutritional information of food items. (O: Mohammad, M1: Tim)

- Must include a mock nutritional database following the same format as the ESHA Nutritional Database.
- A group of arbitrary food items and their nutritional data stored for use within the Diet Journal functionality.
- Nutritional information - The serving size of the item and the caloric intake per serving size must be stored to the database.

3.2. Performance Requirements

3.2.1. Storage/Query Limits

The database interactions are limited to 1 GB of stored data, 50,000 reads per day, and 20,000 writes per day.

3.3. Assumptions and Constraints

Condition	Type	Effect on Requirements
Only valid input would be provided.	**Assumption**	**Allows for minimal error checking for prototype development**
ECG data will be simulated and configured for database	**Constraint**	**Allows for less complicated algorithms to check activity**
Prototype will be run on Android OS.	**Dependency**	**Allows for focus of prototype development on one mobile OS.**

3.4. Non-Functional Requirements

3.4.1. Security

- Encryption methods must be up to date.
- Password’s can only be reset or delivered to the email address used to make the gamer's account.

3.4.2. Maintainability

- Android OS compatibility
- API to cloud storage for Firebase must be up to date.
- Patch updates to keep the application up to date.

3.4.3. Reliability

- The database must be operational at all times.

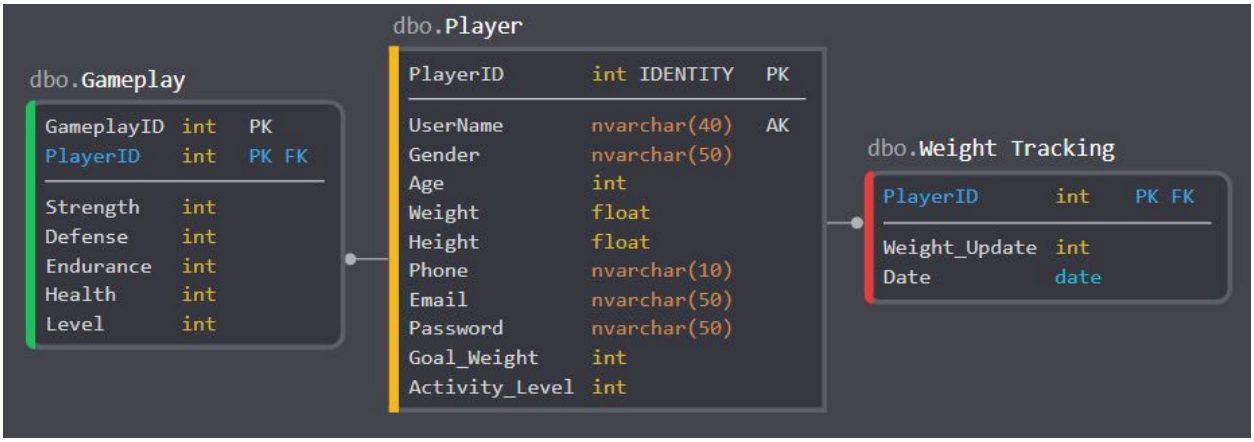
3.4.4. Modifiability

- Application must be able to be modified to add new features
- Application must be able to be modified to update existing features

3.4.5. Firebase

- The application shall utilize the Firebase API for data storage and operations.
- Sensitive information (e.g. usernames, passwords) must be stored securely through Firebase authentication.

Appendix A: Prototype Database Schema (ERD)



Appendix B: Prototype GUI Site Map