Department of Computer Science and Engineering
The University of Texas at Arlington

Virtual Slugger
Breaking Bat

Sean Gibeault
Ehidiamen Ojielu
Brandon Auwaerter
Geoff Graham
Roshan Lamicdhane
## Table of Contents

### REVISION HISTORY

5

### LIST OF TABLES

6

### LIST OF FIGURES

7

### 1. PRODUCT CONCEPT

8

1.1 Purpose and Use

8

1.2 Intended Audience

8

### 2. PRODUCT DESCRIPTION AND FUNCTIONAL OVERVIEW

10

2.1 Features and Functions

10

2.2 External Inputs and Outputs

11

2.3 Product Interfaces

11

### 3. CUSTOMER REQUIREMENTS

14

3.1 Switch Hitting Capability

14

3.2 Setting Configuration

14

3.3 Hit Virtual Baseball

15

3.4 Real World Bat

15

3.5 Virtual Reality Batting Environment

16

3.6 Feedback

16

3.7 Real Time Statistics

17

3.8 Learn Fundamentals of Hitting

17

3.9 Switch Pitching

17

### 4. PACKAGING REQUIREMENTS

19

4.1 Virtual Reality Device

19

4.2 Physical Baseball Bat

19

4.3 Xbox Controller

20

4.4 Sensors

20

4.4 Software

20
5. PERFORMANCE REQUIREMENTS

5.1 LATENCY
5.2 GRAPHICS

6. SAFETY REQUIREMENTS

6.1 SIMULATOR SICKNESS
6.2 BASEBALL BAT HAZARD
6.3 MENTAL HEALTH

7. MAINTENANCE AND SUPPORT REQUIREMENTS

7.1. SOURCE CODE DOCUMENTATION
7.2 HARDWARE SUPPORT
7.3 SOFTWARE SUPPORT

8. OTHER REQUIREMENTS

8.1 AMERICAN ENGLISH STANDARD
8.2 USER FRIENDLY USER INTERFACE

9. ACCEPTANCE CRITERIA

9.1 VERIFY THAT VIRTUAL SLUGGER USES VIRTUAL REALITY
9.2 VERIFY THAT VIRTUAL SLUGGER IS INTERACTIVE
9.2.2 Verification Procedure: The user will be able to interact with the system. He/she will be able to switch the batting to be left/right handed, they will also be able to set configuration and most importantly interact with the virtual baseball by hitting it with the real world physical bat.
9.3 VERIFY THAT THE VIRTUAL SLUGGER USES A REAL WORLD BAT
9.4 VERIFY THAT THE VIRTUAL SLUGGER USER HITS OR MISSES THE PITCH
9.5 VERIFY THE VIRTUAL SLUGGER PROVIDES FEEDBACK
9.6 VERIFY THAT THE VIRTUAL SLUGGER IS SAFE

10. USE CASES

10.1 START VIRTUAL SLUGGER
10.2 START PRACTICE
10.3 SELECT PITCH SPEED
10.4 DISPLAY STATISTICS
10.5 SET CONFIGURATION 33
10.6 QUIT PRACTICE MODE 34
10.7 EXIT VIRTUAL SLUGGER 34

11. FEASIBILITY ASSESSMENT 36

11.1 SCOPE ANALYSIS 36
11.2 RESEARCH 37
11.3 TECHNICAL ANALYSIS 37
11.4 COST ANALYSIS 38
11.5 RESOURCE ANALYSIS 39
11.6 SCHEDULE ANALYSIS 39
- EXTERNAL INPUTS (EI) 40
- EXTERNAL OUTPUTS (EO) 40
- EXTERNAL INQUIRY (EI) 40
- INTERNAL LOGICAL FILES (ILF’S) 40
- EXTERNAL INTERFACE FILES (EIF’S) 40

12 FUTURE ITEMS 44

12.1 IMPLEMENT PRODUCT IN OTHER SPORTS 44
12.2 CHOICE OF PITCHES 44
12.3 CHOICE OF PITCHERS 44
12.4 IMPLEMENT RUMBLE PACK ON IMPACT 45
12.5 VIRTUAL BATS 45
12.6 SITUATIONAL/SEASON MODE 45
12.7 SOCIAL MEDIA INTEGRATION 45
# Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Revision Date</th>
<th>Description</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>07-03-2014</td>
<td>SRS First Draft</td>
<td>Initial Offering of the SRS</td>
</tr>
<tr>
<td>1.0</td>
<td>7-16-2014</td>
<td>SRS Final Draft</td>
<td>Revised from instructor comments</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table #</th>
<th>Title</th>
<th>Page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-2</td>
<td>External Inputs and Outputs</td>
<td>11</td>
</tr>
<tr>
<td>9-1</td>
<td>Virtual Reality Acceptance</td>
<td>25</td>
</tr>
<tr>
<td>9-2</td>
<td>Interactive Acceptance</td>
<td>26</td>
</tr>
<tr>
<td>9-3</td>
<td>Real Bat Acceptance</td>
<td>27</td>
</tr>
<tr>
<td>9-4</td>
<td>Hit Baseball Acceptance</td>
<td>27</td>
</tr>
<tr>
<td>9-5</td>
<td>Feedback Acceptance</td>
<td>27</td>
</tr>
<tr>
<td>9-6</td>
<td>Safety Acceptance</td>
<td>28</td>
</tr>
<tr>
<td>11-4</td>
<td>Cost Analysis</td>
<td>28</td>
</tr>
<tr>
<td>11-6-1</td>
<td>System Characteristic Table</td>
<td>35</td>
</tr>
<tr>
<td>11-6-2</td>
<td>Schedule Estimation</td>
<td>38</td>
</tr>
<tr>
<td>11-6-3</td>
<td>COCOMO Detailed Report</td>
<td>39</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure #</th>
<th>Title</th>
<th>Page #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Conceptual Drawing</td>
<td>9</td>
</tr>
<tr>
<td>2-1</td>
<td>Virtual Slugger Start Screen</td>
<td>12</td>
</tr>
<tr>
<td>2-2</td>
<td>Virtual Slugger Settings Screen</td>
<td>13</td>
</tr>
<tr>
<td>2-3</td>
<td>Virtual Slugger Batting Example</td>
<td>13</td>
</tr>
<tr>
<td>10-1</td>
<td>Use Case</td>
<td>32</td>
</tr>
</tbody>
</table>
1. Product Concept

This section describes the purpose, use and intended user audience for the Virtual Slugger. The Virtual Slugger is a virtual batting simulator that combines many cutting edge technologies like the Oculus Rift, Kinect, and other sensors to simulate a real experience inside of a virtual world. Users will be able to practice their batting skills by telling the system which type of batter they would like to be, which types of pitches they are working on, left handed or right handed and what difficulty level they would like. The system will simulate a batting experience based on the input they specified and will track the users swing as they try to hit the virtual ball. This data will be analyzed and the system will provide statistics and advice on how to get better.

1.1 Purpose and Use

The Virtual Slugger is designed to simulate a real game situation that baseball players will encounter. This demo be on a computer and be launched from the game engine to the main menu. The user will input which type of batter they would like to be such as a power hitter, lead off, or a combination of both. The user will also be able to choose which type of pitches they would like to work on or choose a random order of pitches. Once the game starts the batter will face a virtual pitcher until they either hit the ball or they strike out. Once there is a break in the game (hit or strikeout) the system will give the user feedback on their swing and advice on how to improve i.e.: try to hit the ball lower or higher depending on which type of hitter they have selected.

1.2 Intended Audience

The Virtual Slugger’s intended users are all levels of baseball players ranging from amateur to pro. The intended consumer would be aspiring baseball players or professionals that want to practice their swing. Additional consumers would be baseball teams or organizations that would like to try new technology to improve batting averages.
Fig 1-1 Conceptual Drawing
2. Product Description and Functional Overview

This section provides the reader with an overview of the Virtual Slugger. The primary operational aspect of this product is to allow the user to practice their batting swing and improve via feedback from the system. Coaches could use some of this data collected on the swing and provide additional feedback as well as use the statistics to rate player performance. Some of the main features found in this product include the use of a VR headset, a simulated batting environment, and the use of a real bat as a controller in the environment.

2.1 Features and Functions

The Virtual Slugger will use a real world bat as their primary interaction with the simulator. Users will be able to swing the bat and try to hit pitches from the virtual reality pitcher. The VR headset will be used as the virtual reality headset to immerse the user in the environment. The Kinect will be used to track the bat and collect extra data on the users form and provide another form of user interaction.

The Oculus Rift will act as both an input and an output device. The VR device will be attached to the computer running the game and display the virtual game environment in 3D. It will also track users head movement so the user is able to look around the virtual environment and enhance their virtual reality experience.

The X-Box Controller will be used to interact with the menus and select user preferences. The target audience is males age 14-40 so we are assuming familiarity with this type of interface.

The Kinect will be used to collect additional data on the users form and technique. In addition it may help to track the swing and the moment it actually makes contact with the virtual ball.

Accelerometer and gyroscope sensors will be attached to the bat handle and send the data instantaneously via Bluetooth to the game engine. The game engine will then take this data and render the bat in the game environment. These sensors will also be used to track statistical data on the user’s swing.

The user will be able to set preferences for the game such as batter height, bat length, bat weight, hitting style and player type. The system will store this information until it is changed.

The user will be able to face a pitcher with a predefined set of pitches or a randomized set of pitches to practice their batting. Their statistics will be stored for the duration of their session and they will get feedback once they are done.
2.2 External Inputs and Outputs

Table 2-2 External Inputs and Outputs

<table>
<thead>
<tr>
<th>Data Flow</th>
<th>Name</th>
<th>Description</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Real World Bat</td>
<td>The bat will be equipped with an accelerometer and gyroscope to collect data on the users swing</td>
<td>The user will use the bat to swing at pitches from the simulated pitcher</td>
</tr>
<tr>
<td>Input</td>
<td>Kinect</td>
<td>The Kinect will capture data about the users form and technique</td>
<td>This data will be gathered and used to provide feedback</td>
</tr>
<tr>
<td>Input</td>
<td>Controller</td>
<td>X-Box controller</td>
<td>Will be used to navigate through user menus</td>
</tr>
<tr>
<td>Output</td>
<td>Virtual Reality</td>
<td>Oculus Rift</td>
<td>Will simulate environment for primary user</td>
</tr>
<tr>
<td>Output</td>
<td>Display</td>
<td>Computer Monitor</td>
<td>Displays what user sees</td>
</tr>
</tbody>
</table>

2.3 Product Interfaces

The user will interface with the Breaking Bat System through the initial home screen with options: Practice, Home Run Derby, Random, and Settings. Practice allows you to practice certain objectives, Home Run Derby pitches only straight fastballs and keeps track of how many home runs you hit, Random will randomize the pitches thrown to you, and Settings takes you to a screen to setup the user’s preferences.
**Fig 2-1 Virtual Slugger Start Screen**

![Virtual Slugger Start Screen](image1)

**Fig 2-2 Virtual Slugger Settings Screen**

![Virtual Slugger Settings Screen](image2)
Fig 2-3 Virtual Slugger Batting Example
3. Customer Requirements

This section covers the requirements that have been established by the customer for the Virtual Slugger product. It covers each feature and gives a description to establish the “look and feel” as well as to give an idea of what the end-user should expect the product to do. The requirements have also been given a priority to give a general idea of how important each feature is to the Virtual Slugger product.

3.1 Switch Hitting Capability

3.1.1 Description: The user will have the option of choosing to bat left-handed or right-handed. This will change the view in the Virtual Slugger system to accommodate both batting styles.

3.1.2 Source: Breaking Bat Team.

3.1.3 Constraints: None.

3.1.4 Standards: None.

3.1.5 Priority: Critical.

3.2 Setting Configuration

3.2.1 Description: Virtual Slugger should allow the user before batting to configure their settings such as height of user, speed of pitches, and type of batter. These configurations will personalize settings to determine the user’s strike zone and help give feedback after batting.

3.2.2 Source: Breaking Bat Team.

3.2.3 Constraints: None.

3.2.4 Standards: Strike zone will be calculated from the user’s preferences.

3.2.5 Priority: Critical
3.3 Hit Virtual Baseball

3.3.1 Description: Virtual Slugger should allow the user the ability to see a virtual baseball (pitch) thrown and allow the user to swing a real baseball bat to try to ‘make contact’ with the virtual baseball.

3.3.2 Source: Breaking Bat Team.

3.3.3 Constraints: Interference or latency between the bluetooth signal and the computer might limit our ability to make the sensors communicate with each other.

3.3.4 Standards: None.

3.3.5 Priority: Critical.

3.4 Real World Bat

3.4.1 Description: Virtual Slugger should allow the user to use a real baseball bat in order to make the batting experience as realistic as possible.

3.4.2 Source: Sponsor.

3.4.3 Constraints: Sensor weight and placement may affect physical characteristics of the bat.

3.4.4 Standards: Any standard baseball bat.

3.4.5 Priority: Critical.
3.5 Virtual Reality Batting Environment

3.5.1 **Description:** The Virtual Reality simulator will display a batting environment in order to make the batting experience as realistic as possible.

3.5.2 **Source:** Breaking Bat team.

3.5.3 **Constraints:** Limitations of VR headset in presenting realistic environment.

3.5.4 **Standards:** Pitcher will be on 10 inch mound and be 60.5 feet from the hitter in reference to the user. The release point of the pitch will be 55 feet from home plate.

3.5.5 **Priority:** Moderate.

3.6 Feedback

3.6.1 **Description:** The Virtual Slugger should provide feedback on what to modify in order to improve their batting following a batting session. Different feedback will be delivered based on batting style chosen along with statistics from the session.

3.6.2 **Source:** Breaking Bat and Sponsor.

3.6.3 **Constraints:** The sensors might not be capable to sense the level of detail that we need which may limit our ability to give good feedback to the user.

3.6.4 **Standards:** None.

3.6.5 **Priority:** Critical.
3.7 Real Time Statistics

3.7.1 Description: The Virtual Slugger should gather real-time statistics during the user’s batting session. These statistics will be available for review after each session.

3.7.2 Source: Breaking Bat Team

3.7.3 Constraints: The sensors chosen might not provide fast enough response due to latency.

3.7.4 Standards: None.

3.7.5 Priority: Critical.

3.8 Learn Fundamentals of Hitting

3.8.1 Description: The user will learn the fundamentals of Batting via one of three types of batting style they choose to train for. These three batting styles consist of a “bopper” which trains to hit the ball with power, a “rabbit” which trains to hit the ball on the ground, and a “dirtbag”, who hits the ball for line drives. Based on the batting style chosen, a summary of the batting style will be available along with feedback after batting sessions.

3.8.2 Source: Sponsor

3.8.3 Constraints: User must pick batting style

3.8.4 Standards: None

3.8.5 Priority: Moderate

3.9 Switch Pitching

3.9.1 Description: The Virtual Slugger should provide the user an option to choose whether the pitcher is left handed or right handed.

3.9.2 Source: Sponsor

3.9.3 Constraints: User can choose to face left or right handed pitcher
3.9.4 Standards: None

3.9.5 Priority: Low
4. Packaging Requirements

This section provides information on how the Virtual Slugger will be packaged for the end user.

4.1 Virtual Reality Device

4.1.1 Description: The product will include a virtual reality headset.

4.1.2 Source: Breaking Bat Team

4.1.3 Constraints: None.

4.1.4 Standards: None.

4.1.5 Priority: Critical

4.2 Physical Baseball Bat

4.2.1 Description: A baseball bat will be provided for the user to interact with the virtual environment.

4.2.2 Source: Breaking Bat Team

4.2.3 Constraints: Sensor weight and placement may affect physical characteristics of the bat.

4.2.4 Standards: None.

4.2.5 Priority: High.
4.3 Xbox Controller

4.3.1 Description: An Xbox controller will be provided for users to interact with the menus and select preferences.

4.3.2 Source: Breaking Bat Team

4.3.3 Constraints: Installing necessary drivers for the controller.

4.3.4 Standards: None.

4.3.5 Priority: Moderate.

4.4 Sensors

4.4.1 Description: Sensors will be provided with the product and used with the bat to monitor its motion.

4.4.2 Source: Breaking Bat Team.

4.4.3 Constraints: Weight and size, and therefore function of the sensors may be limited in order to maintain real world characteristics of the bat.

4.4.4 Standards: None.

4.4.5 Priority: Critical.

4.5 Software

4.5.1 Description: Software will be included either as an installable USB or CD/DVD.

4.5.2 Source: Breaking Bat Team.

4.5.3 Constraints: Supported operating systems and platforms by the game engine for packaging.

4.5.4 Standards: None.

4.5.5 Priority: High.
5. Performance Requirements

This section highlights the technical performance requirements of the product.

5.1 Latency

5.1.1 Description: Latency is currently an issue with virtual reality devices. However, the user should experience as little latency as possible to ensure accuracy when hitting pitches.

5.1.2 Source: Breaking Bat Team.

5.1.3 Constraints: Latency is one of the biggest challenges facing virtual reality devices as a whole and has not been solved yet.

5.1.4 Standards: None.

5.1.5 Priority: High

5.2 Graphics

5.2.1 Description: The user should experience smooth gameplay, animations and interactions when using the Virtual Slugger.

5.2.2 Source: Breaking Bat Team.

5.2.3 Constraints: Limited by the computer’s graphics card and resolution.

5.2.4 Standards: None.

5.2.5 Priority: High.
6. Safety Requirements

This section highlights the safety issues associated with the product with detailed description.

6.1 Simulator sickness

6.1.1 Description: This is similar to motion sickness and may cause nausea, sweating or vomiting. The product will help mitigate this by using correct camera calibration and distortion. Also, a warning label will printed on the product to address this requirement.

6.1.2 Source: Virtual Reality Device.

6.1.3 Constraints: Different users react differently to VR environments so each user must assess their susceptibility.

6.1.4 Standards: None.

6.1.5 Priority: Critical

6.2 Baseball Bat Hazard

6.2.1 Description: A warning label will placed on the product advising users to provide adequate room when in use due to the hazards of using a real baseball bat.

6.2.2 Source: Baseball bat.

6.2.3 Constraints: The system cannot ensure adequacy of space.

6.2.4 Standards: None.

6.2.5 Priority: Critical
6.3 Mental Health

6.3.1 **Description:** A label that warns against excessive use will be affixed to the Virtual Slugger as excessive use could affect the health of the user.

6.3.2 **Source:** Virtual Reality Device.

6.3.3 **Constraints:** The system cannot enforce proper use.

6.3.4 **Standards:** None.

6.3.5 **Priority:** Moderate.
7. Maintenance and Support Requirements

This section covers maintenance and support for end users after delivery of product.

7.1. Source Code Documentation

7.1.1 Description: Detailed and clean code must be provided for future reference, updates and technical support.

7.1.2 Source: Breaking Bat Team.

7.1.3 Constraints: None.

7.1.4 Standards: None.

7.1.5 Priority: High.

7.2 Hardware Support

7.2.1 Description: Provide replaceable components for users if issue is within the scope of hardware built by the team.

7.2.2 Source: Breaking Bat Team.

7.2.3 Constraints: Not feasible for third-party components.

7.2.4 Standards: None.

7.2.5 Priority: Moderate.
7.3 Software Support

7.3.1 **Description:** Bugs reported by users will be handled by the software team and marked with the appropriate priority level.

7.3.2 **Source:** Breaking Bat Team.

7.3.3 **Constraints:** None.

7.3.4 **Standards:** None.

7.3.5 **Priority:** High.

7.4 Troubleshooting Guide

7.4.1 **Description:** A troubleshooting guide will be provided with the Virtual Slugger.

7.4.2 **Source:** Breaking Bat Team.

7.4.3 **Constraints:** None.

7.4.4 **Standards:** None.

7.4.5 **Priority:** High.
8. Other Requirements

This section describes extra requirements for the Virtual Slugger to be complete that are not previously listed.

8.1 American English Standard

8.1.1 Description: The Virtual Slugger must use the American English language as the default for any text.

8.1.2 Source: Breaking Bat Team.

8.1.3 Constraints: None.

8.1.4 Standards: None.

8.1.5 Priority: High

8.2 User Friendly User Interface

8.2.1 Description: The Virtual Slugger must have an intuitive and easy to use user interface.

8.2.2 Source: Breaking Bat Team.

8.2.3 Constraints: None.

8.2.4 Standards: None.

8.2.5 Priority: High
9. Acceptance Criteria

In this section of System Requirements Specification we will discuss on what minimum criteria are needed to be satisfied for the system to be acceptable. Here we will be discussing some important aspects that have to be covered by the system for it to meet customer requirements.

9.1 Verify that Virtual Slugger Uses Virtual Reality

9.1.1 Requirement(s) Addressed:

<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5</td>
<td>VR Batting Environment</td>
</tr>
</tbody>
</table>

*Table 9-1 Virtual Reality Acceptance*

9.1.2 Verification Procedure: The user will be able to wear the VR headset and will be interacting with the virtual reality environment, in our case, the baseball stadium and the pitch thrown at him.
9.2 Verify that Virtual Slugger is Interactive

9.2.1 Requirement(s) Addressed:

<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Switch Hitting Capability</td>
</tr>
<tr>
<td>3.2</td>
<td>Set Configuration</td>
</tr>
<tr>
<td>3.3</td>
<td>Hit Baseball</td>
</tr>
</tbody>
</table>

*Table 9-2 Interactive Acceptance*

9.2.2 Verification Procedure: The user will be able to interact with the system. He/she will be able to switch the batting to be left/right handed, they will also be able to set configuration and most importantly interact with the virtual baseball by hitting it with the real world physical bat.
9.3 Verify that the Virtual Slugger Uses a Real World Bat

9.3.1 Requirement(s) Addressed:

<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4</td>
<td>Real World Bat</td>
</tr>
<tr>
<td>3.3</td>
<td>Hit Baseball</td>
</tr>
</tbody>
</table>

Table 9-3 Real Bat Acceptance

9.3.2 Verification Procedure: The user will be able to use a real world physical baseball bat which is connected to the virtual environment and hit or miss the pitch thrown at the user in virtual environment by swinging the bat in real world.

9.4 Verify that the Virtual Slugger User Hits or Misses the Pitch

9.4.1 Requirement(s) Addressed:

<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>Hit Baseball</td>
</tr>
</tbody>
</table>

Table 9-4 Hit Baseball Acceptance
9.4.2 **Verification Procedure:** The user will be able to use a real world physical baseball bat which is connected to the virtual environment and hit or miss the pitch thrown at the user in virtual environment by swinging the bat in real world.

9.5 **Verify the Virtual Slugger Provides Feedback**

9.5.1 **Requirement(s) Addressed:**

<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.6</td>
<td>Feedback</td>
</tr>
</tbody>
</table>

*Table 9-5 Feedback Acceptance*

9.5.2 **Verification Procedure:** After a brief session of training the user will be able to view the statistics of his/her batting and the system will provide advice to the user on how to improve.

9.6 **Verify that the Virtual Slugger is Safe**

9.6.1 **Requirement(s) Addressed:**

<table>
<thead>
<tr>
<th>Requirement Number</th>
<th>Requirement Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>Simulator Sickness</td>
</tr>
<tr>
<td>6.2</td>
<td>Baseball Bat Hazard</td>
</tr>
<tr>
<td>6.3</td>
<td>Mental Health</td>
</tr>
</tbody>
</table>

*Table 9-6 Safety Acceptance*
9.6.2 Verification Procedure: Multiple users will be tested to make sure that the system is safe to use. There will be an emphasis to use in moderation so that the user does not experience dizziness from prolonged exposure to the VR headset. Batting will be performed in the solitary environment for example a batting cage and it will be attached to the player’s wrist. Somebody with previous history of seizures will have to consult the doctor before using the Virtual Slugger.
10. Use Cases

In this section, we will be listing and describing our use cases. The uses cases are the business processes that the system will perform. A brief description of how the user performs the task and how the system reacts will be outlined in the following use cases.

10.1 Start Virtual Slugger

10.1.1 Scenario: The user starts the program by clicking Virtual Slugger Icon. Then the program will load and the user will be able to start practice or set configurations based on the user’s needs.

10.1.2 Actor(s): User.

10.1.3 TUCBW: The user clicks Virtual Slugger Icon.

10.1.4 TUCEW: The user sees the program loading.

10.2 Start Practice

10.2.1 Scenario: The user starts to practice when the user selects the option from the main menu. The user will be able to set configuration before starting practice mode on the program.

10.2.2 Actor(s): User.

10.2.3 TUCBW: The user clicks “Start Practice” button.

10.2.4 TUCEW: The user sees the visual display of practice mode initiation.
10.3 Select Pitch Speed

10.3.1 Scenario: The users opens the Virtual Slugger and then clicks on “Select Speed” button and selects one of the radio buttons to choose a desired speed of ball he wants to bat against.

10.3.2 Actor(s): User.

10.3.3 TUCBW: The user clicks the “Select Speed” button and chooses the speed of the fast ball from the radio button.

10.3.4 TUCEW: The user sees the confirmation of selected ball speed.

10.4 Display Statistics

10.4.1 Scenario: After finishing batting practice, the user will be able to view the statistics of how far the ball travelled and the speed of the bat.

10.4.2 Actor(s): User.

10.4.3 TUCBW: The user clicks the “View Results” button.

10.4.4 TUCEW: The user sees the ball travelled distance in feet.

10.5 Set Configuration

10.5.1 Scenario: The user will be able to set configurations such as left/right handed batter and height of the batter.

10.5.2 Actor(s): User.

10.5.3 TUCBW: The user clicks the “Set Configuration” button.

10.5.4 TUCEW: The user sees the confirmation of chosen configuration.
10.6 Quit Practice Mode

10.6.1 Scenario: The user will be able to quit the practice mode after completing the practice session.

10.6.2 Actor(s): User.

10.6.3 TUCBW: The user clicks “Quit Practice” button.

10.6.4 TUCEW: The user sees the main Virtual Slugger window.

10.7 Exit Virtual Slugger

10.7.1 Scenario: The user will be able to exit the program from the main window of the Virtual Slugger.

10.7.2 Actor(s): User.

10.7.3 TUCBW: The user clicks “Exit Program” button.

10.7.4 TUCEW: The user sees the home screen.
Fig 10-1 Use Case
11. Feasibility Assessment

This section will analyze the feasibility of the Virtual Reality Batting Simulator. This section includes the scope analysis, research, technical analysis, cost analysis, resource analysis and schedule analysis.

11.1 Scope Analysis

After analyzing all the requirements, this project is a moderate to difficult project for our team given the time frame we have available and the amount of skills we need to learn. Prototyping this project should be feasible if we keep our advanced system interface requirements low. This analysis is based on the previous Senior Design project similar to ours which also uses the VR headset. We will not only be learning to work with the rift and game engine but also creating a controller out of a real world bat. We expect that 3 requirements will comprise the bulk of the work load. These are Requirements are 3.4 Real World Bat, 3.5 VR Batting Environment, and 3.3 Hit Baseball.

Requirement 3.4 Real World Bat is our highest priority requirement. This is the basis of our project and we need to put a lot of our time into making sure this requirement is functional. We will most likely use a combination of accelerometer and gyroscopes in addition to the Kinect to track the bat in the real world. Requirement 3.3 Hit Baseball will be our next highest priority requirement but goes hand in hand with the first one mentioned. The bat must be tracked and be able to hit the baseball as it’s being pitched for this to work and it must be extremely accurate if we want the prototype to have real world training application. Requirement 3.5 VR Batting Environment is less important than the previous two but still a vital part of our user’s experience. This type of thing has been done before and we know that it is possible, but we lack time to learn how to create the environment and player models and we may end up using templates from Unreal/Unity to get us started.

Finishing Requirements 3.3, 3.4 and 3.5 give us a good bases for Virtual Slugger and will be the first requirements we will have to work on. Once we have these in place adding the additional requirements like 3.1 switch hitting, 3.2 setting configuration, 3.6 feedback, 3.7 real-time statistics shouldn’t be too hard to implement. If these are all done correctly requirement 3.8 will take care of itself as the user will be able to learn the fundamentals of batting through our application.
11.2 Research

This section includes the research that needs to be accomplished in order to better understand the feasibility of the product. This section includes two parts; the research that has been done and the research that needs to be done.

The research that has been completed is as follows:

- The Oculus Rift has two different versions. The second version has better resolution lenses and it also has a head tracker. The higher resolution might be a necessity in order for the user to see the baseball being pitched at him/her.

The research that needs to be completed is as follows:

- There are two game engines that will work with the Oculus Rift; Unity and Unreal. The team needs to research what will work best with the other equipment and our amateur experience level.
- The game engines will require 3D models and the research needs to be done to figure out what tools would work in order to create the models.
- The team needs to research a way to track the location and the speed of the bat in real time. This data then needs to be passed back to the game engine for further processing.
- The team will need to research how to physically integrate the sensors on the baseball bat.
- Research needs to be done to see if the Kinect 2.0 will be able to track user movements and the bat.
- Research will need to be done on what development methodology will best suit the scope of the project.

11.3 Technical Analysis

The project will be a virtual reality hitting simulator that will track hitting stats and offer tips to improve hitting performance. The hardware will consist of an virtual reality headset for the virtual reality environment, a combination of sensors which will track bat speed and bat location, and Unreal gaming engine to develop the game environment. None of the team members have experience with any of this technology. We will need to research a way to send the bat information collected by the sensors to the Unreal game engine. This process will need to be done seamlessly in order to accurately capture bat location. We will also need a system to configure settings.
11.4 Cost Analysis

The team is budgeted $800 for all the equipment needed to build our prototype. After all of our research we found that we would be within budget. The Virtual Slugger is a simulator where the user will experience hitting a baseball virtually by swinging a real baseball bat. The majority of our cost is the equipment needed for this simulator. We have identified the major cost being the Oculus Rift at $350 and the Kinect at $300. We determined that we will not need to purchase any other hardware or software. Table 11-1 gives a breakdown of the cost analysis.

<table>
<thead>
<tr>
<th>Description</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oculus Rift</td>
<td>$350</td>
</tr>
<tr>
<td>Sensors (Accelerometer, Gyroscope)</td>
<td>$39.86</td>
</tr>
<tr>
<td>Baseball Bat</td>
<td>Free</td>
</tr>
<tr>
<td>X-Box Controller</td>
<td>Free</td>
</tr>
<tr>
<td>Unreal License</td>
<td>$20 a month</td>
</tr>
<tr>
<td>Kinect SDK</td>
<td>$300</td>
</tr>
<tr>
<td>Total</td>
<td>$709.86</td>
</tr>
</tbody>
</table>

*Table 11-4 Cost Analysis*
11.5 Resource Analysis

Team Breaking Bat is made up of three computer scientists, a computer engineer, and a software engineer. Our three computer scientists have extensive knowledge and experience in a variety of languages including C, C++, C#, Java, and .Net Entity Framework which is great since unreal uses C++ and Unity uses C#. Our Computer Engineer has experience with a wide variety of hardware and sensors in addition to work experience at Cisco. Our Software Engineer will be able to keep us on track and organized due to his experience with MS Project and management skills.

Our team weakness are a lack of experience with integrating hardware and software, game design and 3D modeling. To compensate for this we have already spoken to a few people with game design experience for their advice and help us learn the processes of game development. In addition, we will be testing out how the Oculus Rift works with the gaming engines in order to get an idea of what we are developing. As for the hardware integration, we are doing research into accelerometers and gyroscopes that could be used on the bat and how to integrate that into the gaming engines and physically integrate them onto the baseball bat. We have spoken to someone that is very interested in hardware integration from the Mobi club and will use his advice on hardware.

All of our members have their individual strengths and weaknesses but our greatest strength is teamwork. Everyone has bought into the idea proposed and is self motivated to put out a working prototype by the end of the year.

11.6 Schedule Analysis

The First Method we use for schedule estimation is Jones First Order Estimation because we want to get a rough estimate of the total time it might take to complete a project of this size. Below we categorize inputs and outputs into 5 different categories and rank them by difficulty as either high, medium or low based on the number of File Types Referenced (FTR) and Data Element Types (DET). Using the International Function Point User Group (IFPUG) tables we derived how many function points for each component. Results are displayed below:
System Requirements Specification

Breaking Bat

■ External Inputs (EI)

▪ Main Menu Low 3
▪ Settings Low 3
▪ Practice Settings Low 3
▪ Oculus Rift Low 3
▪ Accelerometer Low 3
▪ Gyroscope Low 3
▪ Kinect High 6
▪ Controller Medium 4

■ External Outputs (EO)

▪ Oculus Rift High 7
▪ Computer Monitor Low 4

■ External Inquiry (EI)

▪ Feedback High 6

■ Internal Logical Files (ILF’s)

▪ Statistics tracking Low 7
▪ User Preferences Low 7

■ External Interface Files (EIF’s)

▪ none

Total Unadjusted Function Points: 59
After getting our unadjusted total we must calculate our Value Adjustment Factor (VAF) based on 14 different System Characteristics. The degree of influence is ranked from 0-5 for each characteristic.

<table>
<thead>
<tr>
<th>General System Characteristic</th>
<th>Influence (0-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Communications</td>
<td>5</td>
</tr>
<tr>
<td>Distributed Data Processing</td>
<td>0</td>
</tr>
<tr>
<td>Performance</td>
<td>5</td>
</tr>
<tr>
<td>Heavily Used Configuration</td>
<td>3</td>
</tr>
<tr>
<td>Transaction Rate</td>
<td>0</td>
</tr>
<tr>
<td>Online Data Entry</td>
<td>0</td>
</tr>
<tr>
<td>End-User Efficiency</td>
<td>5</td>
</tr>
<tr>
<td>Online Update</td>
<td>0</td>
</tr>
<tr>
<td>Complex Processing</td>
<td>4</td>
</tr>
<tr>
<td>Re-usability</td>
<td>2</td>
</tr>
<tr>
<td>Installation Ease</td>
<td>2</td>
</tr>
<tr>
<td>Operational Ease</td>
<td>1</td>
</tr>
<tr>
<td>Multiple Sites</td>
<td>0</td>
</tr>
<tr>
<td>Facilitate Change</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>29</td>
</tr>
<tr>
<td>Influence Factor (.65 + Sc/100)</td>
<td>.94</td>
</tr>
</tbody>
</table>

*Table 11-6-1 System Characteristic Table*

**Total Adjusted Function Points:** 55.46
Using Jones First Order Estimation Practice we are able to estimate our schedule based on a Business Project as seen below:

<table>
<thead>
<tr>
<th>Business Application</th>
<th>Best Case</th>
<th>Avg Case</th>
<th>Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exponent</td>
<td>.41</td>
<td>.43</td>
<td>.46</td>
</tr>
<tr>
<td>Estimated Schedule(Months)</td>
<td>55.46^.41 = 5.19</td>
<td>55.46^.43 = 5.62</td>
<td>55.46^.46 = 6.34</td>
</tr>
</tbody>
</table>

*Table 11-6-2 Schedule Estimation*

We also ran some COCOMO Estimation software on our project requirements and got a somewhat similar estimate, below is our detailed report from the program:

<table>
<thead>
<tr>
<th>Estimate Name: Virtual Slugger</th>
<th>Model Name: COCOMO II 2000</th>
<th>Process Model: COCOMO II Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimate ID:</td>
<td>Model ID: 2000</td>
<td>Phases: Waterfall</td>
</tr>
<tr>
<td>Component Name: Component1</td>
<td>Increment: 1</td>
<td>Level: 1</td>
</tr>
<tr>
<td>Developed Size: 2,939</td>
<td>EAF: 0.7050</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phase</th>
<th>Effort (Person-Months)</th>
<th>Cost (K$)</th>
<th>Duration (Months)</th>
<th>Staffing</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ -- Requirements</td>
<td>0.5</td>
<td>0.0</td>
<td>1.1</td>
<td>0.4</td>
</tr>
<tr>
<td>PD -- Product Design</td>
<td>1.1</td>
<td>0.0</td>
<td>1.6</td>
<td>0.7</td>
</tr>
<tr>
<td>DD -- Detailed Design</td>
<td>1.7</td>
<td>0.0</td>
<td>1.6</td>
<td>1.1</td>
</tr>
<tr>
<td>CT -- Code &amp; Unit Test</td>
<td>2.4</td>
<td>0.0</td>
<td>2.1</td>
<td>1.1</td>
</tr>
<tr>
<td>IT -- Integration &amp; Test</td>
<td>1.3</td>
<td>0.0</td>
<td>1.3</td>
<td>0.9</td>
</tr>
<tr>
<td>Development (PD+DD+CT+IT)</td>
<td>6.5</td>
<td>0.0</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td>Totals (RQ+PD+DD+CT+IT)</td>
<td>7.0</td>
<td>0.0</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>MN -- Maintenance (per year)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
</tbody>
</table>

*Table 11-6-3 COCOMO Detailed Report*
As of 7/3/2014 we have roughly 5-5.5 months to finish our project and deliver a prototype. Our rough estimate puts us at 5.19 - 6.34 month till completion which would fit an optimistic schedule. The COCOMO estimation software estimated that it would take 6.6 months for development and 1.1 month for requirement, which is what we have just completed so we would be about 1.1 month over our schedule at this point. We will have to keep looking at this throughout the development process and try to reduce requirements or speed up the schedule in some other way. In conclusion, the project is feasible, however given the time and resource constraints Team Breaking Bat will use an incremental development cycle focusing on the critical and high priority requirements.
12 Future Items

12.1 Implement Product in Other Sports

12.1.1 Description: For the scope of this project we will be focusing on giving the user the full baseball experience, but we have thought about broadening the product to other sports such as soccer and basketball.

12.1.2 Constraint: Schedule - Due to lack of time to develop the complex product in five months period we will not be able to implement them in other sports.

12.2 Choice of Pitches

12.2.1 Description: We will try to implement different kinds of pitches that can be selected by a batsman. We will implement curve-ball, change-up and breaking balls in the future development.

12.2.2 Constraint: Schedule - Limited time to develop the product. We also lack the understanding of the mechanism of curve-ball, changing up and breaking balls work for now.

12.3 Choice of Pitchers

12.3.1 Description: We will try to simulate real life pitchers in the application. Users will be able to select from the different Major League Baseball pitchers and bat against those pitchers.

12.3.2 Constraint: Schedule and License - Limited time to implement real life pitchers. We will also not be able to use the real life players’ information without getting permission from Major League Baseball.
12.4 Implement Rumble Pack on Impact

12.4.1 Description: We will try to implement the rumbling of the bat when the batter hits the baseball.

12.4.2 Constraint: Schedule and Budget - Limited time to implement rumbling and we also don’t have sufficient budget to spend on the rumbling sensors.

12.5 Virtual Bats

12.5.1 Description: The players will be able to choose different kinds of virtual bats ranging from metal bats, composite bats, wooden bats and hybrid bats.

12.5.2 Constraint: Schedule - Limited time to implement the choices of different bats. We won’t have enough time to work around more than one bat and we have chosen to work on metal bat as a standard.

12.6 Situational/Season Mode

12.6.1 Description: We will try to implement different modes in our system. We have thought about broadening the product from training mode to situational mode. In situational mode the batsman will have to hit against certain types of pitches to make certain runs.

12.6.2 Constraint: Schedule - Limited time to implement different types of modes. Training will be our standard mode in the present context.

12.7 Social Media Integration

12.7.1 Description: We will try to make the stats shareable in social media sources such as Facebook and Twitter to make it recreational as well as social and competitive.

12.7.2 Constraint: Schedule - Limited time to integrate social media features in our product because we will be spending time to develop the working prototype of the product.