**Advance Augmented Reality Battle-system (AARB)**

**CS 337 Software DesignFunctional Requirements and Design Document**Prepared by

Namba, KazuyaLiang, TonyLee, Hsu PengHsu, Michael

Approved By:

Jose Macias 05/15/2012

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_

Professor Date

**Document Change Log**

|  |  |  |
| --- | --- | --- |
| Update | Date Released | Changes |
| Draft #1 | 05/15/2012 | First Draft |
| Final | 05/30/2012 | Delivery version |

**List of TBD Items**

|  |  |  |  |
| --- | --- | --- | --- |
| Page | Item | Description | Status |
| Section 4 | Detailed Design | Level 2 DFD | TBI in cs437 |
| Section 5 | Implementation | Coding | TBI in cs437 |
| Section 6 | Test Plan | Testing Plan | TBI in cs437 |

**Table of Contents**

[1.0 INTRODUCTION 5](#_Toc477629727)

[1.1 Purpose 5](#_Toc477629728)

[1.2 Scope 5](#_Toc477629729)

[1.2.1 Document Organization 5](#_Toc477629730)

[1.2.2 Relationship to Other Documents 5](#_Toc477629731)

[1.3 AARB Architecture 6](#_Toc477629732)

[1.3.1 Context Diagram (Level 0 Data Flow Diagram) 6](#_Toc477629733)

[1.3.2 Description and Major Functions 6](#_Toc477629734)

[1.3.3 Hardware and Software Considerations 7](#_Toc477629735)

[1.4 Documentation of the Development Process 7](#_Toc477629736)

[1.5 References 7](#_Toc477629737)

[1.5.1 Controlling Documents 8](#_Toc477629738)

[1.5.2 Applicable Documents 8](#_Toc477629739)

[1.5.3 Standards 8](#_Toc477629740)

[2.0 AARB DETAILED FUNCTIONAL DESCRIPTION 8](#_Toc477629741)

[2.1 Detailed AARB Functional Description 8](#_Toc477629742)

[2.1.1 Level 1 Data Flow Diagram (DFD) 8](#_Toc477629743)

[2.1.2 Detailed Functional Description of AARB’s Major Units 9](#_Toc477629744)

[3.0 AARB REQUIREMENTS 13](#_Toc477629745)

[3.1 AARB Functional Requirements 13](#_Toc477629746)

[3.2 AARB Non-Functional Requirements 25](#_Toc477629747)

[3.3 AARB Hardware Requirements 25](#_Toc477629748)

[4.0 AARBDETAILED DESIGN 26](#_Toc477629749)

[4.1 Level 2 DFDs 26](#_Toc477629750)

[5.0 AARBIMPLEMENTATION 26](#_Toc477629751)

[6.0 AARB TEST PLAN 26](#_Toc477629752)

[6.1 Introduction 26](#_Toc477629753)

[6.2 Functional Requirements Validation Matrix 26](#_Toc477629754)

[A. ACRONYMS 41](#_Toc477629755)

[B. DATA DICTIONARY 42](#_Toc477629756)

**List of Figures**

Figure 1-1 Context Diagram (Level 0 Data Flow Diagram)Figure 2-1 Level 1 Data Flow Diagram

**List of Tables**

None.

# 1.0 INTRODUCTION

# 1.1 Purpose

The purpose of this document is four-fold:

1. Define a full set of requirements for AARB. These sections correspond to a Software Requirements Document (SRD).
2. Define the design for AARB. These sections correspond to a Software Design Document (SDD).
3. Define the implementation of AARB. These sections correspond to a Software Implementation Document (SID) and will be partially completed during a subsequent Software Engineering class.
4. Define the Test Plan for AARB. These sections correspond to a Software Test Plan (STP) and will be partially completed during a subsequent Software Engineering class.

The complete definition of all AARB requirements provides the requirements to be used in the subsequent software subsystem documents.

# 1.2 Scope

This documentation was developed as part of a Software Design class, CS337. Implementation will continue in a subsequent Software Engineering class.

The scope of this document includes the following:

1. All functional and non-functional requirements. These requirements are organized by key AARB functional units shown on the Level 1 Data Flow Diagram (DFD) given in section 2.0.
2. A trace matrix, relating all AARB functional requirements to functional subunits as expanded in lower level DFDs. Higher level DFDs will be provided next quarter as part of the design in section 4.0.
3. General descriptions of hardware necessary for implementation of AARB.
4. A matrix of testing methods for each functional requirement.

## 1.2.1 Document Organization

The organization of this document provides a natural 'flow' or allocation of requirements to each succeeding section. Details regarding the overall document structure are discussed in sub-section 1.4.

## 1.2.2 Relationship to Other Documents

The AARB SRD/SDD/SID/STP is a complete self-contained document. Some relationships to other documents in the literature are indicated below in sub-section 1.5.

# 1.3 AARB Architecture

## 1.3.1 Context Diagram (Level 0 Data Flow Diagram)

AARB’s architecture is summarized in the Context Diagram (Level 0 DFD) given below. A more detailed Functional Description is given in section 2 of this document.

**Figure 1-1:** **Level 0 DFD**

**H/W**

**Control**

**Signals**

**Advance Augmented Reality Battle-system (AARB)**

* + **Main Control**
  + **Login**
  + **Sound Recognition**
  + **Motion Recognition**
  + **Battle**
  + **Shop**
  + **Location Service**
  + **User Profile**

**User**

**Input**

**Hardware**

**User**

## 1.3.2 Description and Major Functions

This project is based on the hardware concept (which will be referenced as Google Glasses in this paper) showcased in the Project Glass video from Google (link can be found at the References section).

The basic concept of Project Glass is to create a pair of lightweight augmented reality (AR) glasses that are capable of using various Google services, such as location services (the heart of the project), various Google apps, social media platforms, and so on, that will enhance user experience based on real-time environment around the user. AARB takes advantage of the location based augmented reality functions Google displayed in the video.

AARB is essentially a cloud-based augmented reality online role-playing game. Each user creates his or her character through character customization options. Each character starts with basic attacks and spells that the user can use. When players perform certain real-world tasks, such as physical conditioning and studying, the Google Glasses collects data through a special adapter, allowing players to gain “experience points” that will essentially “level up” their characters. When a character’s level is increased, upgrades to the character’s attack or spells are available. In addition, an online shop is available for purchasing upgrades with gold as currency. Using Google’s location service, players can log into a lobby and search for players nearby that are seeking for a duel. Battles take place in predetermined maps retrieved from the server. The battle commences once players get to the assigned spots. Players will seek each other using a minimap that will provide an overview of the area. Using voice commands and body action, players can perform various attacks or spells. Through the glasses, players can see their commands come true and interact with the environment, such as burning the surroundings with a fireball (in other words, augmented reality). The primary objective is to get the enemy’s health to zero. After the battle, the winner gets experience and gold, which can contribute to leveling the character and purchasing items from the shop respectively.

## 1.3.3 Hardware and Software Considerations

AARB requires AR glasses. Other hardware requirements include installing a server on data centers around the globe. The server will support mass amount of data processing and storage.

# 1.4 Documentation of the Development Process

AARB’s detailed functional description is documented in section 2.0. Section 2.0 is a succinct software description document. The overall detailed functional description is based on higher level DFDs (above level 1). All major functional units are described in detail in this part of the document.

Requirements for AARB are captured in section 3.0 of this document. This section includes both functional and non-functional software requirements, which are supplemented with more detailed information as necessary.

Section 4, AARB’s detailed Design Description Document (SDD), will be completed in a subsequent Software Engineering class.

Section 5, AARB’s Software Implementation Document (SID), will be completed in a subsequent Software engineering class.

Section 6, AARB’s Software Test Plan (STP), will be completed in a subsequent Software engineering class.

# 1.5 References

All references used in the creation of this document are listed below.

"Augmented Reality." *Wikipedia*. Wikimedia Foundation, 06 Feb. 2012. Web. 04 June 2012. <http://en.wikipedia.org/wiki/Augmented\_reality>.

"Cloud Computing." *Wikipedia*. Wikimedia Foundation, 06 Apr. 2012. Web. 04 June 2012. <http://en.wikipedia.org/wiki/Cloud\_computing>.

"Google Begins Testing Its Augmented-Reality Glasses." *Bits Blog*. Web. 04 June 2012. <http://bits.blogs.nytimes.com/2012/04/04/google-begins-testing-its-augmented-reality-glasses/>.

"Project Glass - Google." *Project Glass - Google*. Web. 04 June 2012. <https://plus.google.com/111626127367496192147>.

"Role-playing Game." *Wikipedia*. Wikimedia Foundation, 06 Jan. 2012. Web. 04 June 2012. <http://en.wikipedia.org/wiki/Role-playing\_game>.

## 1.5.1 Controlling Documents

There is no other document controlling this document.

## 1.5.2 Applicable Documents

No additional applicable document has been used in the production of this document.

## 1.5.3 Standards

No standard has been used in the creation of this document. However, some Standards described in textbooks have been examined as a reference. In particular, the IEEE standard has been briefly discussed in class.

# 2.0 AARB DETAILED FUNCTIONAL DESCRIPTION

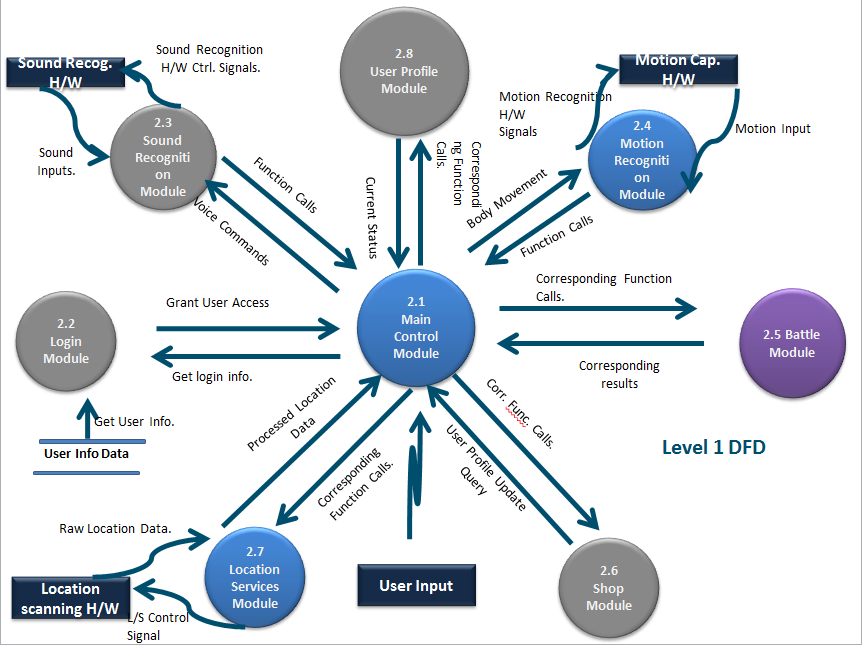
# 2.1 Detailed AARB Functional Description

The major tool used to design AARB is the DFD. The rationale for the selection of DFDs as the preferred design tool is their simplicity and versatility. In the future, additional tools may be used if a stronger correlation from Design to Requirement to Implementation and Testing is required.

## 2.1.1 Level 1 Data Flow Diagram (DFD)

AARB’s major functional subunits are shown in the DFD Level 1 shown below:

**Figure 2.1: Level 1 DFD**



## 2.1.2 Detailed Functional Description of AARB’s Major Units

The description of AARB’s major functional units shown in Figure 2.1 follows. Most of AARB’s operations depend on the Main Control Module’s Information Bus Moduleto transfer data between the modules.

Main Control - Module 2.1

The Main Control Module (MCM), a server side module that is completely cloud-based, provides the main game’s Graphical User Interface (GUI) and communication between other modules in AARB. The MCM contains submodules that fetch and display various user and game information on the main GUI. The MCM also provides internal communication between each module.

The User Interface Submodule (UIS) is instantiated every time a user runs the game. It presents a GUI and gives limited access to the MCM’s submodules: Data Input Controller Submodule (DICS), the Information Bus Submodule (IBS), and the Feedback Submodule (FS).

The Data Input Controller Submodule (DICS) determines the type of input to be taken in based on the current state of the user by using the IBS to fetch data from the User Profile Module’s (UPM) User Statistics Submodule (USTS). There are two states, Not In Battle (NIB) or In Battle (IB), which are specified in the USTS requirements. When the user is NIB, the DICS only queues the usage of the Sound Recognition Module (SRM). When the user is IB, the DICS queues the SRM, the Motion Recognition Module (MRM), and the Location Service Module (LSM). Users can toggle the LSM through the GUI when the user is NIB.

The Information Bus Submodule (IBS) is responsible for processing request, making corresponding function calls, and transferring data between modules, specifying specific submodules if requested.

The Feedback Submodule (FS) is responsible for reporting and handling any potential errors that may occur when running the program.

Log-in / Security function - Module 2.2

The Login module provides standard security features to protect the system from unauthorized access.

Sound Recognition - Module 2.3

The Sound Recognition Module (SRM) matches the user’s voice to built-in functions. Users issue valid commands by first saying the name of the menu (the names are reserved keywords in the software) that is displayed on the GUI and then saying the command desired (e.g., “volume, lower”). On receiving the user’s input, the Command Parsing Submodule (CPS) identifies the reserved keywords and the commands following them, which then goes to corresponding modules through the MCM’s IBM.

The Hearing Submodule (HS) describes hardware interaction with the SRM and error handling.

Motion Recognition - Module 2.4

The Motion Recognition Module (MRM) matches the user’s motions to built-in functions. This module uses the UPM’s USTS to switch between IB and NIB. When NIB, the user issues commands by raising the hand above shoulder level and perform intuitive arm movements to control the menu (e.g., swipe hands to navigate between menus). When IB, all motions are recorded. On receiving the user’s input, the Motion Parsing Submodule (MPS) checks the state of the user, identifies the required position if applicable, and parses the user movements, which then go to corresponding modules through the MCM’s IBM.

The Capturing Submodule (CS) describes hardware interaction with the MRM and error handling.

Battle - Module 2.5

The Battle Module (BM) has four submodules: Battle Interface Submodule (BIS), Matchmaking Submodule (MS), Battle Calculation Submodule (BCS), and Battle Data Submodule (BDS).  
  
The Battle Interface Submodule (BIS) is responsible for main display during the battle. It displays the screens for setting up a new battle, current battle data, and the end battle’s result. During the battle, it is responsible for generating special effects and accessing the User Profile Module to show the current user’s status.

The Matchmaking Submodule (MMS) takes care of setting up the game. It asks the user to choose to be the host player or not. If the user decides to play as the host player, the user customizes the number of players, the time limit, and the battle area. If the user decides to play as a regular player, the user must join a nearby available host player’s battle. While the player is waiting for other players to join the battle, the player can access the Shop Module (SM) to purchase items and access the UPM to change his or her profile or view other players’ profiles. As soon as enough players are gathered, it prompts the player to confirm to begin the battle. Players can leave the battle mid-game or after the battle is over.

The Battle Calculation Submodule (BCS) does all the calculations that occur during the battle. As soon as it gets input from the SRM and the MRM, including attacks, skills, and magic from opponents or the player, the BCS requests the UPM for player data, such as the level, the skill levels, the equipment, and the statuses. By using these data, it does calculations and reflects the changes by passing the values back to the Battle Data Submodule (BDS).

The Battle Data Submodule (BDS) stores all information related to the current battle. It includes the current time, the time remain, the current location, the map, the equipment, the skills, and the status of all players in the battle. BIS uses these resources to display the current situation of the battle.

Shop - Module 2.6

The Shop Module (SM) provides the same function as Amazon.com. Although players can unlock their skills when they level up and they can grab pick-ups while playing, they can also log into the shop to search for items and skills to purchase. These items and skills help players improve their characters.

The Calculation Item Submodule (CIS) stores all information related to the items available from the shop and provides transactions of the items. It provides descriptions of the items and their prices. Items are sorted by categories, such as “potion” (e.g., health, magic, or damage increase) and “equipment” (e.g., weapons, armors, GPS detector, etc.). It also calculates the player’s funds during the process of a transaction.

The Calculation Skills Submodule (CSS) stores all information related to the skills available from the shop and provides transactions of the skills. It provides descriptions of the skills and their prices. The skills are defined by physical skills (e.g., reaction, fast, damage, dexterity, etc.), active skills (e.g., fireball, ice ball, lightning ball, etc.), and passive skills (e.g., abilities, logic, movement, etc.). It also calculates the player’s funds during the process of a transaction.

The User Transaction Submodule (UTS) provides various payment methods for players to choose from when they are purchasing from the shop such as gold, PayPal, prepaid cards, mobile payment, etc.

The User Interface Submodule (UIS) records and updates players’ profiles, items, skills, and gold before and after every transaction.

The Search Categories Submodule (SCS) lets players browse and search for available items and skills in the shop, which are organized, sorted, and categorized (e.g., weapons, armors, consumptions, etc.).

Location Service - Module 2.7

The Location Service Module (LSM) module, implemented in the AR Glass, controls the GPS H/W and includes two submodules, Geographic Scanning Submodule (GSS) and Location Tracking Submodule (LTS). The GSS is responsible for scanning the battlefield chosen by the host player and generating the 2D/3D geographic image. The LTS controls the GPS and tracks the current user’s location. The data for both submodules will be stored in the BM’s BDS.

User Profile - Module 2.8

The User Profile Module (UPM) handles all user profile related information with its six submodules: User Assets Submodule (UAS), User Experience Submodule (UES), User Skills Submodule (USS), User Items Submodule (UIS), User Location Submodule (ULS), and User Statistics Submodule (USTS).

The User Assets Submodule (UAS) processes all player transactions with the shop. It transfers assets between the shop and the player and updates the player’s funds.

The User Experience Submodule (UES) processes all experiences given to the player. It calculates the experience requirements for each level, adds experience gained from battle or training to the player, and increments the player’s level when the requirements are met.

The User Skills Submodule (USS) processes all player skills related information. It provides new players with basic skills. Every player has endurance/stamina and mana. Physical skills consume endurance/stamina while active skills consume mana. The USS also records any changes made to the current skills and adds new skills unlocked by the player.

The User Items Submodule (UITS) processes all player interactions with items. It labels, describes, and displays every single item in the player’s inventory. It updates the player’s inventory when the player adds or removes an item. It imposes the limitations of the player’s items and the restrictions of the player’s equipment. For example, if the player picks up a first aid kit, then the first aid kit can only be used to heal the player. If the player picks up a hat, the player can only wear it in his or her head. The player cannot use it as a weapon, wear it as a shoe, or eat it as food.

The User Location Submodule (ULS) processes the player’s current precise location to the UPM. It displays the full address of the player’s location. It also displays a map view with a character icon as the marker of the player’s location and the direction the player is facing.

The User Statistics Submodule (USTS) processes the player’s statistics to the UPM. It displays the player’s statistics: profile name, location, current game status, level, experience, equipment, appearance, health, mana, endurance, stamina, strengths, weakness, immunities, and skills.

# 3.0 AARB REQUIREMENTS

# 3.1 AARB Functional Requirements

This section lists AARB’s functional requirements. This section includes the complete set of functional requirements, along with explanations for cases in which the statement of the requirement was deemed insufficient or requires additional clarification. All requirements relate to the design modules described in section 2.0. An effort has been made to standardize the correlation between the design modules and the requirements to make access and organization more consistent. For example, requirement number “n” affecting module 2.1 will be labeled 3.1.n.

|  |  |
| --- | --- |
| **Requirements Related to Design Module 2.1: Main Control Module (MCM)** | |
| Requirement No. | Requirement Description |
| **User Interface Submodule (UIS)** | |
| 3.1.1.1 | UIS shall be instantiated every time the user launches the game. |
| 3.1.1.2 | UIS shall be a GUI with scrolling menus. |
| 3.1.1.3 | UIS shall display user controllable options. |
| 3.1.1.4 | UIS shall take user input using the Data Input Controller Submodule. |
| 3.1.1.5 | UIS shall display any significant errors or messages to the user. |
| **Data Input Controller Submodule (DICS)** | |
| 3.1.2.1 | DICS shall determine the state of the user. |
| 3.1.2.2 | DICS shall obtain data from the User Profile Module using the Information Bus Submodule. |
| 3.1.2.3 | DICS shall queue the Motion Recognition Module if necessary. |
| 3.1.2.4 | DICS shall queue the Sound Recognition Module if necessary. |
| **Information Bus Submodule (IBS)** | |
| 3.1.3.1 | IBS shall handle requests from all modules. |
| 3.1.3.2 | IBS shall handle responses from all modules. |
| **Feedback Submodule (FS)** | |
| 3.1.4 | FS shall monitor all modules and report all errors to the MCM. |
| 3.1.4.1 | FS shall receive monitoring input from all modules. |
| 3.1.4.2 | FS shall record all errors into a database. |
| 3.1.4.3 | FS shall retrieve errors from the database. |
| 3.1.4.4 | FS shall ensure atomicity of transactions. |
| 3.1.4.5 | FS shall detect and avoid deadlocks. |
| 3.1.4.6 | FS shall receive requests for any errors from the user. |
| 3.1.4.7 | FS shall report errors to the user and suggest appropriate actions. |
| 3.1.4.7.1 | FS shall communicate appropriate actions to modules to recover from any errors. |

|  |  |
| --- | --- |
| **Requirements Related to Design Module 2.2: Login Module (LM)** | |
| Requirement No. | Requirement Description |
| 3.2 | **Login module shall provide standard access security services.** |

|  |  |
| --- | --- |
| **Requirements Related to Design Module 2.3: Sound Recognition Module (SRM)** | |
| Requirement No. | Requirement Description |
| **Command Parsing Submodule (CPS)** | |
| 3.3.1.1 | CPS shall recognize reserved keywords. |
| 3.3.1.2 | CPS shall recognize commands following the reserved keywords. |
| 3.3.1.3 | CPS shall use natural language recognition to identify words. |
| 3.3.1.4 | CPS shall deliver the command to the corresponding module. |
| 3.3.1.4.1 | CPS shall use the IBS to deliver commands. |
| **Hearing Submodule (HS)** | |
| 3.3.2.1 | HS shall receive a live sound signal from the Hearing hardware every nanosecond. |
| 3.3.2.1.1 | HS shall send the live sound signal to the Sync Module. |
| 3.3.2.2 | HS shall determine if there is a problem with the Hearing hardware. |
| 3.3.2.2.1 | HS shall correct any software problems with the Hearing hardware. |

|  |  |
| --- | --- |
| **Requirements Related to Design Module 2.4: Motion Recognition Module (MRM)** | |
| Requirement No. | Requirement Description |
| **Motion Parsing Submodule (MPS)** | |
| 3.4.1.1 | MPS shall verify that the user’s arm is above shoulder level. |
| 3.4.1.2 | MPS shall recognize motions if the user’s arm is above shoulder level. |
| 3.4.1.3 | MPS shall use pattern recognition technology to recognize motions. |
| 3.4.1.3.1 | MPS shall parse recognized motions into commands. |
| 3.4.1.4 | MPS shall deliver the commands to the corresponding module. |
| 3.4.1.4.1 | MPS shall use the IBS to deliver the commands. |
| **Capturing Submodule (CS)** | |
| 3.4.2.1 | CS shall receive a live signal from the Motion Recognition hardware every nanosecond. |
| 3.4.2.2 | CS shall determine if there is a problem with the Motion Recognition hardware. |
| 3.4.2.2.1 | CS shall correct any software problems with the Motion Recognition hardware. |

|  |  |
| --- | --- |
| **Requirements Related to Design Module 2.5: Battle Module (BM)** | |
| Requirement No. | Requirement Description |
| **Battle Interface Submodule (BIS)** | |
| 3.5.1.1 | BIS shall provide the GUI. |
| 3.5.1.2 | BIS shall operate by sound or motion commands detected by the SRM or the MRM. |
| 3.5.1.3 | BIS shall display the “Choose Host/Player” screen. |
| 3.5.1.4 | BIS shall display the “Setup Battle Environment” screen for the host player. |
| 3.5.1.5 | BIS shall display the “Search Host” screen for the regular player. |
| 3.5.1.5.1 | BIS shall display the “List of Players” screen showing the list of players hosting battles including participating players’ name and level. |
| 3.5.1.6 | BIS shall display the “Battle” screen during actual battle. |
| 3.5.1.7 | BIS shall obtain the current battle data by accessing the Battle Data Submodule. |
| 3.5.1.7.1 | BIS shall display the current time, the time limit, and the time remaining. |
| 3.5.1.7.2 | BIS shall display the current location and the map of the battlefield. |
| 3.5.1.7.3 | BIS shall display network connection status. |
| 3.5.1.7.4 | BIS shall display the current user status. |
| 3.5.1.7.5 | BIS shall display other players’ status. |
| 3.5.1.8 | BIS shall display special effects according to the player’s motion or sound. |
| 3.5.1.9 | BIS shall display the “Show Result” screen. |
| **Matchmaking Submodule (MMS)** | |
| 3.5.2.1 | MMS shall give the user a choice to play as the Host Player or a Regular player. |
| 3.5.2.2 | MMS shall enable the Host Player to choose the number of players and the area of the battlefield. |
| 3.5.2.3 | MMS shall enable a Regular Player to choose a player hosting a battle. |
| 3.5.2.4 | MMS shall access every players’ User Profile Module that joined the same battle. |
| 3.5.2.4.1 | MMS shall enable the user access to the User Profile Module to change or update his or her user profile. |
| 3.5.2.4.2 | MMS shall enable the user to view other players’ name and level. |
| 3.5.2.5 | MMS shall enable the user to access the Shop Module to buy or sell items. |
| 3.5.2.6 | MMS shall force players to wait until enough players have joined in the battle. |
| 3.5.2.7 | MMS shall ask the user to confirm starting the battle. |
| 3.5.2.8 | MMS shall give battle-setting data to the Battle Data Submodule as the battle begins. |
| 3.5.2.9 | MMS shall enable the user to exit from battle. |
| 3.5.2.9.1 | MMS shall let the user choose to exit or continue the battle. |
| **Battle Calculation Submodule (BCS)** | |
| 3.5.3.1 | BCS shall operate by sound or motion detection by the SRM or the MRM. |
| 3.5.3.1.1 | BCS shall get user action as input from the SRM or the MRM. |
| 3.5.3.2 | BCS shall get required user data for calculation from the Battle Data Submodule. |
| 3.5.3.3 | BCS shall do calculations using user action and user profile data. |
| 3.5.3.3.1 | BCS shall return the result to the Battle Data Submodule. |
| **Battle Data Submodule (BDS)** | |
| 3.5.4 | BDS shall store current battle data. |
| 3.5.4.1 | BDS shall store the current time, the time limit, and the time remaining. |
| 3.5.4.2 | BDS shall store the current location, the area of the battlefield, and the 2D/3D map generated by Location Service Module. |
| 3.5.4.3 | BDS shall store current network connection status. |
| 3.5.4.4 | BDS shall synchronize with the user’s User Profile Module and store the information including the status. |
| 3.5.4.5 | BDS shall synchronize other players’ User Profile Module in the same battle and store their information including their status. |

|  |  |
| --- | --- |
| **Requirements Related to Design Module** **2.6: Shop Module (SM)** | |
| Requirement No. | Requirement Description |
| **Calculation Item Submodule (CIS)** | |
| 3.6.1 | CIS shall provide transaction of items to the SM. |
| 3.6.1.1 | CIS shall store items in the SM. |
| 3.6.1.2 | CIS shall display the full price of items to the SM. |
| 3.6.1.3 | CIS shall calculate the user’s funds. |
| 3.6.1.3.1 | CIS shall add funds to the SM when the user purchased item(s). |
| 3.6.1.3.2 | CIS shall subtract funds from the SM when the user sold item(s). |
| 3.6.1.4 | CIS shall display any error messages to the SM when the connection is dead. |
| **Calculation Skills Submodule (CSS)** | |
| 3.6.2 | CSS shall provide transaction of skills to the SM. |
| 3.6.2.1 | CSS shall store skills in the SM. |
| 3.6.2.2 | CSS shall display the full price of skills to the SM. |
| 3.6.2.3 | CSS shall calculate the user’s funds. |
| 3.6.2.3.1 | CSS shall add funds to the SM when the user purchased skill(s). |
| 3.6.2.3.2 | CSS shall subtract funds from the SM when the user sold skill(s). |
| 3.6.2.4 | CSS shall display any error messages to the SM when the connection is dead. |
| **User Transaction Submodule (UTS)** | |
| 3.6.3 | UTS shall provide several purchase methods to the SM when the user tries to get extra gold. |
| 3.6.3.1 | UTS shall have pay card method. |
| 3.6.3.2 | UTS shall have PayPal method. |
| 3.6.3.3 | UTS shall have a mobile payment method. |
| 3.6.3.4 | UTS shall have prepaid card method. |
| 3.6.3.5 | UTS shall have Paysafecard method. |
| 3.6.3.6 | UTS shall have a game gold method. |
| **User Interface Submodule (UIS)** | |
| 3.6.4 | UIS shall display the user’s profile. |
| 3.6.4.1 | UIS shall display the user’s current items. |
| 3.6.4.1.1 | UIS shall display the user’s items after he or she finishes shopping. |
| 3.6.4.2 | UIS shall display the user’s current skills. |
| 3.6.4.2.1 | UIS shall display the user’s skills after he or she finishes shopping. |
| 3.6.4.3 | UIS shall display the user’s current gold. |
| 3.6.4.3.1 | UIS shall display the user’s gold after he or she finishes shopping. |
| **Search Categories Submodule (SCS)** | |
| 3.6.5 | SCS shall organize all items and skills. |
| 3.6.5.1 | SCS shall provide searching function. |
| 3.6.5.2 | SCS shall sort items by grade. |
| 3.6.5.2.1 | SCS shall split items into weapons, armors, and potions. |
| 3.6.5.2.1.1 | SCS shall split weapons into swords, guns, spears, and magic wands. |
| 3.6.5.2.1.2 | SCS shall split armors into helmets, shields, shoes, gloves, and hats. |
| 3.6.5.2.1.3 | SCS shall split potions into health, magic, endurance, and cleanses. |
| 3.6.5.3 | SCS shall sort skills by grade. |
| 3.6.5.3.1 | SCS shall split skills into a fire, ice, and lightning ball. |
| 3.6.5.3.1.1 | SCS shall split fireball into level 1 through level 10. |
| 3.6.5.3.1.2 | SCS shall split ice ball into level 1 through level 10. |
| 3.6.5.3.1.3 | SCS shall split lightning ball into level 1 through level 10. |

|  |  |
| --- | --- |
| **Requirements Related to Design Module 2.7: Location Service Module (LSM)** | |
| Requirement No. | Requirement Description |
| **Geographic Scanning Submodule (GSS)** | |
| 3.7.1.1 | GSS shall access the BDS and get the area of the battlefield. |
| 3.7.1.2 | GSS shall scan the battlefield. |
| 3.7.1.2.1 | GSS shall generate the 2D map. |
| 3.7.1.2.2 | GSS shall generate the 3D map. |
| **Location Tracking Submodule (LTS)** | |
| 3.7.2 | LTS shall track the current user’s location. |

|  |  |
| --- | --- |
| **Requirements Related to Design Module 2.8: User Profile Module (UPM)** | |
| Requirement No. | Requirement Description |
| **User Assets Submodule (UAS)** | |
| 3.8.1 | UAS shall process all user transactions with the SM to the UPM. |
| 3.8.1.1 | UAS shall update the UPM when the transaction was successful. |
| 3.8.1.2 | UAS shall cancel the transaction when errors occurred. |
| 3.8.1.2.1 | UAS shall not let the transaction go through if the user does not have sufficient funds. |
| 3.8.1.3 | UAS shall subtract the user’s funds when the user bought asset(s). |
| 3.8.1.3.1 | UAS shall transfer the correct asset(s) to the user when the user bought the asset(s). |
| 3.8.1.4 | UAS shall add funds to the user when the user sold his or her asset(s) to the shop. |
| 3.8.1.4.1 | UAS shall take the correct asset(s) from the user when the user sold the asset(s). |
| **User Experience Submodule (UES)** | |
| 3.8.2 | UES shall process all experiences given to the user to the UPM. |
| 3.8.2.1 | UES shall have every new user’s experience start at zero. |
| 3.8.2.2 | UES shall calculate the user’s experience required to advance to the next level. |
| 3.8.2.3 | UES shall add experience calculated from the BCS to the user when the user won a battle. |
| 3.8.2.4 | UES shall add experience to the user when the user does any physical training or exercise. |
| 3.8.2.5 | UES shall not ever take away the user’s experience under any circumstances. |
| 3.8.2.6 | UES shall increment the user’s level when he or she has the required experience. |
| 3.8.2.6.1 | UES shall not reset the user’s experience back to zero every time the user advanced a level. |
| **User Skills Submodule (USS)** | |
| 3.8.3 | USS shall process all user skills related information to the UPM. |
| 3.8.3.1 | USS shall provide the user with basic skills. |
| 3.8.3.2 | USS shall have three types of skills. |
| 3.8.3.2.1 | USS shall have physical skills. |
| 3.8.3.2.1.1 | USS shall drain the user’s endurance or stamina when using physical skills. |
| 3.8.3.2.2 | USS shall have active skills. |
| 3.8.3.2.2.1 | USS shall drain the user’s mana when using active skills. |
| 3.8.3.2.3 | USS shall have passive skills. |
| 3.8.3.2.3.1 | USS shall not drain anything from the user when using passive skills. |
| 3.8.3.3 | USS shall unlock new skills for the user when he or she bought them from the shop. |
| 3.8.3.4 | USS shall unlock new skills for the user when he or she meets the level requirement. |
| 3.8.3.5 | USS shall increment a skill’s level when the user bought an upgrade from the shop. |
| 3.8.3.6 | USS shall make all user changes to his or her skill(s) to be permanent. |
| 3.8.3.6.1 | USS shall not allow the user’s skills to be transferable. |
| 3.8.3.6.2 | USS shall not allow the user’s skills to be refundable. |
| 3.8.3.6.3 | USS shall not allow the user’s skills to be tradable. |
| **User Items Submodule (UITS)** | |
| 3.8.4 | UITS shall process all user interactions with items to the UPM. |
| 3.8.4.1 | UITS shall set the same maximum number of inventory slots that every user has. |
| 3.8.4.2 | UITS shall show name labels next to every item the user has. |
| 3.8.4.3 | UITS shall show a brief description of every item the user has. |
| 3.8.4.4 | UITS shall add the item to the user’s inventory when the user picked it up. |
| 3.8.4.5 | UITS shall add the correct asset(s) to the user’s inventory when the user bought them from the shop. |
| 3.8.4.6 | UITS shall remove the asset(s) from the user’s inventory when the user sold them to the shop. |
| 3.8.4.7 | UITS shall categorize every equipment the user has to the appropriate body part. |
| 3.8.4.7.1 | UITS shall label every equipment the user has with the appropriate body part. |
| 3.8.4.7.2 | UITS shall not allow the user to equip the equipment to the wrong body part. |
| 3.8.4.7.2.1 | UITS shall only allow one equipment to be equipped at a time for each part of the user’s body. |
| 3.8.4.7.2.1.1 | UITS shall only allow one equipment to be equipped at a time as the user’s headgear. |
| 3.8.4.7.2.1.2 | UITS shall only allow one equipment to be equipped at a time as the user’s eyewear. |
| 3.8.4.7.2.1.3 | UITS shall only allow one equipment to be equipped at a time as the user’s nose wear. |
| 3.8.4.7.2.1.4 | UITS shall only allow one equipment to be equipped at a time as the user’s mouth gear. |
| 3.8.4.7.2.1.5 | UITS shall only allow one equipment to be equipped at a time as the user’s ear wear. |
| 3.8.4.7.2.1.6 | UITS shall only allow one equipment to be equipped at a time as the user’s neckwear. |
| 3.8.4.7.2.1.7 | UITS shall only allow one equipment to be equipped at a time as the user’s arm wear for each arm. |
| 3.8.4.7.2.1.8 | UITS shall only allow one equipment to be equipped at a time as the user’s wrist wear for each wrist. |
| 3.8.4.7.2.1.9 | UITS shall only allow one equipment to be equipped at a time as the user’s finger wear for each finger of each hand of the user. |
| 3.8.4.7.2.1.10 | UITS shall only allow one equipment to be equipped at a time as the user’s torso gear. |
| 3.8.4.7.2.1.11 | UITS shall only allow one equipment to be equipped at a time as the user’s back wear. |
| 3.8.4.7.2.1.12 | UITS shall only allow one equipment to be equipped at a time as the user’s legwear. |
| 3.8.4.7.2.1.13 | UITS shall only allow one equipment to be equipped at a time as the user’s footwear for each foot. |
| 3.8.4.7.2.1.14 | UITS shall only allow at most two weapons to be equipped at a time. |
| 3.8.4.7.2.1.14.1 | UITS shall only allow two weapons to be equipped at a time if the two weapons are lightweight. |
| 3.8.4.7.2.1.14.2 | UITS shall only allow at most one weapon to be equipped at a time if the weapon is heavyweight. |
| 3.8.4.7.2.1.14.3 | UITS shall allow one weapon and one shield to be equipped at the same time. |
| **User Location Submodule (ULS)** | |
| 3.8.5 | ULS shall process the user’s current location from the LSM to the UPM. |
| 3.8.5.1 | ULS shall record the user’s exact location. |
| 3.8.5.2 | ULS shall translate the user’s exact location to the full street address. |
| 3.8.5.2.1 | ULS shall display the user’s full street address. |
| 3.8.5.3 | ULS shall display a map view with a character icon at the user’s current location. |
| 3.8.5.3.1 | ULS shall display the direction the user is currently facing. |
| **User Statistics Submodule (USTS)** | |
| 3.8.6 | USTS shall process the user’s statistics to the UPM. |
| 3.8.6.1 | USTS shall display the user’s profile name. |
| 3.8.6.2 | USTS shall display the user’s location. |
| 3.8.6.3 | USTS shall display the user’s current game status. |
| 3.8.6.3.1 | USTS shall display the current game status as In Battle (IB) if the user is in battle. |
| 3.8.6.3.2 | USTS shall display the current game status as Not In Battle (NIB) if the user is not in battle. |
| 3.8.6.4 | USTS shall display the user’s level. |
| 3.8.6.5 | USTS shall display the user’s experience. |
| 3.8.6.6 | USTS shall display the current equipment the user is wearing. |
| 3.8.6.6.1 | USTS shall change the user’s appearance when the user equips any of his or her equipment. |
| 3.8.6.6.2 | USTS shall change the user’s appearance when the user removes or changes any of his or her equipment. |
| 3.8.6.7 | USTS shall display the user’s current health. |
| 3.8.6.8 | USTS shall display the user’s current mana. |
| 3.8.6.9 | USTS shall display the user’s current endurance. |
| 3.8.6.10 | USTS shall display the user’s current stamina. |
| 3.8.6.11 | USTS shall display the user’s strengths. |
| 3.8.6.12 | USTS shall display the user’s weaknesses. |
| 3.8.6.13 | USTS shall display the user’s immunities. |
| 3.8.6.14 | USTS shall display the user’s skills. |

# 3.2 AARB Non-Functional Requirements

This section collects all the AARB Non-Functional Requirements. All non-functional requirements are numbered “NF – n” where “n” indicates the nth requirement.

NF – 1 AARB requires sufficient data storage to store a library of commands, maps, and users account information.

NF – 2 AARB must execute and transfer data as rapidly as possible.

NF – 3 AARB must be operable by players who have little technical training.

# 3.3 AARB Hardware Requirements

This section collects all AARB’s electronic hardware requirements. All hardware requirements are numbered “H – n” where “n” indicates the nth requirement.

H – 1 AARB will run on standard microcomputer hardware.

H – 2 AARB will require Patron display devices for supertitle display. Each device must include a touchscreen suitable for simple text input and a screen capable of clearly displaying parallel libretto text in two languages.

# 4.0 AARBDETAILED DESIGN

In this section, AARB will be designed in detail including several higher level DFDs. Each major module detailed design is included in correspondence with the design sections defined in section 2.0 and responding to the requirements listed in its correlated sub-section in section 3.0.

# 4.1 Level 2 DFDs

The detailed design of each of the modules discussed in section 2.0 with requirements presented in section 3.0 is presented in the Figures below.

**THIS WILL BE DESIGNED IN CS 437.**

# 5.0 AARBIMPLEMENTATION

In this section, (some of) the modules designed in section 4.0 with requirements listed in section 3.0 will be implemented initially at the level of pseudo code. Actual code will be provided as time permits. Each module is implemented in correspondence with the design sections defined in section 2.0 and responding to the requirements listed in its correlated sub-section in section 3.0.

**THIS WILL BE IMPLEMENTED IN CS 437.**

# 6.0 AARB TEST PLAN

# 6.1 Introduction

In this section, the testing methodology to be used to verify and validate (V&V) each of the requirements listed in section 3.0 has been identified. At some points, additional testing may be required. They shall be documented as an attachment to this document.

The methodologies and testing strategies identified at this point include three major approaches with various variations to adapt to the AARB project:

1. Testing using additional ad-hoc created software including a correlation-testing unit.
2. Demonstration of the specified capability.
3. Inspection of the software code, possibly using additional inspection techniques.

# 6.2 Functional Requirements Validation Matrix

The AARB Functional Requirements Validation Matrix is given below.

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module 2.1: Main Control Module (MCM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| **User Interface Submodule (UIS)** | | |
| 3.1.1.1 | UIS shall be instantiated every time the user launches the game. | Demonstration |
| 3.1.1.2 | UIS shall be a GUI with scrolling menus. | Demonstration |
| 3.1.1.3 | UIS shall display user controllable options. | Demonstration |
| 3.1.1.4 | UIS shall take user input using the Data Input Controller Submodule. | Demonstration |
| 3.1.1.5 | UIS shall display any significant errors or messages to the user. | Demonstration |
| **Data Input Controller Submodule (DICS)** | | |
| 3.1.2.1 | DICS shall determine the state of the user. | Demonstration |
| 3.1.2.2 | DICS shall obtain data from the User Profile Module using the Information Bus Submodule. | Demonstration |
| 3.1.2.3 | DICS shall queue the Motion Recognition Module if necessary. | Demonstration |
| 3.1.2.4 | DICS shall queue the Sound Recognition Module if necessary. | Demonstration |
| **Information Bus Submodule (IBS)** | | |
| 3.1.3.1 | IBS shall handle requests from all modules. | Demonstration |
| 3.1.3.2 | IBS shall handle responses from all modules. | Demonstration |
| **Feedback Submodule (FS)** | | |
| 3.1.4 | FS shall monitor all modules and report all errors to the MCM. | Demonstration |
| 3.1.4.1 | FS shall receive monitoring input from all modules. | Demonstration |
| 3.1.4.2 | FS shall record all errors into a database. | Demonstration |
| 3.1.4.3 | FS shall retrieve errors from the database. | Demonstration |
| 3.1.4.4 | FS shall ensure atomicity of transactions. | Demonstration |
| 3.1.4.5 | FS shall detect and avoid deadlocks. | Demonstration |
| 3.1.4.6 | FS shall receive requests for any errors from the user. | Demonstration |
| 3.1.4.7 | FS shall report errors to the user and suggest appropriate actions. | Demonstration |
| 3.1.4.7.1 | FS shall communicate appropriate actions to modules to recover from any errors. | Demonstration |

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module 2.2: Login Module (LM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| 3.2 | **Login module shall provide standard access security services.** | Demonstration |

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module 2.3: Sound Recognition Module (SRM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| **Command Parsing Submodule (CPS)** | | |
| 3.3.1.1 | CPS shall recognize reserved keywords. | Demonstration |
| 3.3.1.2 | CPS shall recognize commands following the reserved keywords. | Demonstration |
| 3.3.1.3 | CPS shall use natural language recognition to identify words. | Demonstration |
| 3.3.1.4 | CPS shall deliver the command to the corresponding module. | Demonstration |
| 3.3.1.4.1 | CPS shall use the IBS to deliver commands. | Demonstration |
| **Hearing Submodule (HS)** | | |
| 3.3.2.1 | HS shall receive a live sound signal from the Hearing hardware every nanosecond. | Demonstration |
| 3.3.2.1.1 | HS shall send the live sound signal to the Sync Module. | Demonstration |
| 3.3.2.2 | HS shall determine if there is a problem with the Hearing hardware. | Demonstration |
| 3.3.2.2.1 | HS shall correct any software problems with the Hearing hardware. | Demonstration |

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module 2.4: Motion Recognition Module (MRM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| **Motion Parsing Submodule (MPS)** | | |
| 3.4.1.1 | MPS shall verify that the user’s arm is above shoulder level. | Demonstration |
| 3.4.1.2 | MPS shall recognize motions if the user’s arm is above shoulder level. | Demonstration |
| 3.4.1.3 | MPS shall use pattern recognition technology to recognize motions. | Demonstration |
| 3.4.1.3.1 | MPS shall parse recognized motions into commands. | Demonstration |
| 3.4.1.4 | MPS shall deliver the commands to the corresponding module. | Demonstration |
| 3.4.1.4.1 | MPS shall use the IBS to deliver the commands. | Demonstration |
| **Capturing Submodule (CS)** | | |
| 3.4.2.1 | CS shall receive a live signal from the Motion Recognition hardware every nanosecond. | Demonstration |
| 3.4.2.2 | CS shall determine if there is a problem with the Motion Recognition hardware. | Demonstration |
| 3.4.2.2.1 | CS shall correct any software problems with the Motion Recognition hardware. | Demonstration |

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module 2.5: Battle Module (BM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| **Battle Interface Submodule (BIS)** | | |
| 3.5.1.1 | BIS shall provide the GUI. | Demonstration |
| 3.5.1.2 | BIS shall operate by sound or motion commands detected by the SRM or the MRM. | Demonstration |
| 3.5.1.3 | BIS shall display the “Choose Host/Player” screen. | Demonstration |
| 3.5.1.4 | BIS shall display the “Setup Battle Environment” screen for the host player. | Demonstration |
| 3.5.1.5 | BIS shall display the “Search Host” screen for the regular player. | Demonstration |
| 3.5.1.5.1 | BIS shall display the “List of Players” screen showing the list of players hosting battles including participating players’ name and level. | Demonstration |
| 3.5.1.6 | BIS shall display the “Battle” screen during actual battle. | Demonstration |
| 3.5.1.7 | BIS shall obtain the current battle data by accessing the Battle Data Submodule. | Demonstration |
| 3.5.1.7.1 | BIS shall display the current time, the time limit, and the time remaining. | Demonstration |
| 3.5.1.7.2 | BIS shall display the current location and the map of the battlefield. | Demonstration |
| 3.5.1.7.3 | BIS shall display network connection status. | Demonstration |
| 3.5.1.7.4 | BIS shall display the current user status. | Demonstration |
| 3.5.1.7.5 | BIS shall display other players’ status. | Demonstration |
| 3.5.1.8 | BIS shall display special effects according to the player’s motion or sound. | Demonstration |
| 3.5.1.9 | BIS shall display the “Show Result” screen. | Demonstration |
| **Matchmaking Submodule (MMS)** | | |
| 3.5.2.1 | MMS shall give the user a choice to play as the Host Player or a Regular player. | Demonstration |
| 3.5.2.2 | MMS shall enable the Host Player to choose the number of players and the area of the battlefield. | Demonstration |
| 3.5.2.3 | MMS shall enable a Regular Player to choose a player hosting a battle. | Demonstration |
| 3.5.2.4 | MMS shall access every players’ User Profile Module that joined the same battle. | Demonstration |
| 3.5.2.4.1 | MMS shall enable the user access to the User Profile Module to change or update his or her user profile. | Demonstration |
| 3.5.2.4.2 | MMS shall enable the user to view other players’ name and level. | Demonstration |
| 3.5.2.5 | MMS shall enable the user to access the Shop Module to buy or sell items. | Demonstration |
| 3.5.2.6 | MMS shall force players to wait until enough players have joined in the battle. | Demonstration |
| 3.5.2.7 | MMS shall ask the user to confirm starting the battle. | Demonstration |
| 3.5.2.8 | MMS shall give battle-setting data to the Battle Data Submodule as the battle begins. | Demonstration |
| 3.5.2.9 | MMS shall enable the user to exit from battle. | Demonstration |
| 3.5.2.9.1 | MMS shall let the user choose to exit or continue the battle. | Demonstration |
| **Battle Calculation Submodule (BCS)** | | |
| 3.5.3.1 | BCS shall operate by sound or motion detection by the SRM or the MRM. | Demonstration |
| 3.5.3.1.1 | BCS shall get user action as input from the SRM or the MRM. | Demonstration |
| 3.5.3.2 | BCS shall get required user data for calculation from the Battle Data Submodule. | Demonstration |
| 3.5.3.3 | BCS shall do calculations using user action and user profile data. | Demonstration |
| 3.5.3.3.1 | BCS shall return the result to the Battle Data Submodule. | Demonstration |
| **Battle Data Submodule (BDS)** | | |
| 3.5.4 | BDS shall store current battle data. | Demonstration |
| 3.5.4.1 | BDS shall store the current time, the time limit, and the time remaining. | Demonstration |
| 3.5.4.2 | BDS shall store the current location, the area of the battlefield, and the 2D/3D map generated by Location Service Module. | Demonstration |
| 3.5.4.3 | BDS shall store current network connection status. | Demonstration |
| 3.5.4.4 | BDS shall synchronize with the user’s User Profile Module and store the information including the status. | Demonstration |
| 3.5.4.5 | BDS shall synchronize other players’ User Profile Module in the same battle and store their information including their status. | Demonstration |

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module** **2.6: Shop Module (SM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| **Calculation Item Submodule (CIS)** | | |
| 3.6.1 | CIS shall provide transaction of items to the SM. | Demonstration |
| 3.6.1.1 | CIS shall store items in the SM. | Demonstration |
| 3.6.1.2 | CIS shall display the full price of items to the SM. | Demonstration |
| 3.6.1.3 | CIS shall calculate the user’s funds. | Demonstration |
| 3.6.1.3.1 | CIS shall add funds to the SM when the user purchased item(s). | Demonstration |
| 3.6.1.3.2 | CIS shall subtract funds from the SM when the user sold item(s). | Demonstration |
| 3.6.1.4 | CIS shall display any error messages to the SM when the connection is dead. | Demonstration |
| **Calculation Skills Submodule (CSS)** | | |
| 3.6.2 | CSS shall provide transaction of skills to the SM. | Demonstration |
| 3.6.2.1 | CSS shall store skills in the SM. | Demonstration |
| 3.6.2.2 | CSS shall display the full price of skills to the SM. | Demonstration |
| 3.6.2.3 | CSS shall calculate the user’s funds. | Demonstration |
| 3.6.2.3.1 | CSS shall add funds to the SM when the user purchased skill(s). | Demonstration |
| 3.6.2.3.2 | CSS shall subtract funds from the SM when the user sold skill(s). | Demonstration |
| 3.6.2.4 | CSS shall display any error messages to the SM when the connection is dead. | Demonstration |
| **User Transaction Submodule (UTS)** | | |
| 3.6.3 | UTS shall provide several purchase methods to the SM when the user tries to get extra gold. | Demonstration |
| 3.6.3.1 | UTS shall have pay card method. | Demonstration |
| 3.6.3.2 | UTS shall have PayPal method. | Demonstration |
| 3.6.3.3 | UTS shall have a mobile payment method. | Demonstration |
| 3.6.3.4 | UTS shall have prepaid card method. | Demonstration |
| 3.6.3.5 | UTS shall have Paysafecard method. | Demonstration |
| 3.6.3.6 | UTS shall have a game gold method. | Demonstration |
| **User Interface Submodule (UIS)** | | |
| 3.6.4 | UIS shall display the user’s profile. | Demonstration |
| 3.6.4.1 | UIS shall display the user’s current items. | Demonstration |
| 3.6.4.1.1 | UIS shall display the user’s items after he or she finishes shopping. | Demonstration |
| 3.6.4.2. | UIS shall display the user’s current skills. | Demonstration |
| 3.6.4.2.1 | UIS shall display the user’s skills after he or she finishes shopping. | Demonstration |
| 3.6.4.3 | UIS shall display the user’s current gold. | Demonstration |
| 3.6.4.3.1 | UIS shall display the user’s gold after he or she finishes shopping. | Demonstration |
| **Search Categories Submodule (SCS)** | | |
| 3.6.5 | SCS shall organize all items and skills. | Demonstration |
| 3.6.5.1 | SCS shall provide searching function. | Demonstration |
| 3.6.5.2 | SCS shall sort items by grade. | Demonstration |
| 3.6.5.2.1 | SCS shall split items into weapons, armors, and potions. | Demonstration |
| 3.6.5.2.1.1 | SCS shall split weapons into swords, guns, spears, and magic wands. | Demonstration |
| 3.6.5.2.1.2 | SCS shall split armors into helmets, shields, shoes, gloves, and hats. | Demonstration |
| 3.6.5.2.1.3 | SCS shall split potions into health, magic, endurance, and cleanses. | Demonstration |
| 3.6.5.3 | SCS shall sort skills by grade. | Demonstration |
| 3.6.5.3.1 | SCS shall split skills into a fire, ice, and lightning ball. | Demonstration |
| 3.6.5.3.1.1 | SCS shall split fireball into level 1 through level 10. | Demonstration |
| 3.6.5.3.1.2 | SCS shall split ice ball into level 1 through level 10. | Demonstration |
| 3.6.5.3.1.3 | SCS shall split lightning ball into level 1 through level 10. | Demonstration |

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module 2.7: Location Service Module (LSM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| **Geographic Scanning Submodule (GSS)** | | |
| 3.7.1.1 | GSS shall access the BDS and get the area of the battlefield. | Demonstration |
| 3.7.1.2 | GSS shall scan the battlefield. | Demonstration |
| 3.7.1.2.1 | GSS shall generate the 2D map. | Demonstration |
| 3.7.1.2.2 | GSS shall generate the 3D map. | Demonstration |
| **Location Tracking Submodule (LTS)** | | |
| 3.7.2 | LTS shall track the current user’s location. | Demonstration |

|  |  |  |
| --- | --- | --- |
| **Requirements Related to Design Module 2.8: User Profile Module (UPM)** | | |
| Requirement No. | Requirement Description | Testing Method |
| **User Assets Submodule (UAS)** | | |
| 3.8.1 | UAS shall process all user transactions with the SM to the UPM. | Demonstration |
| 3.8.1.1 | UAS shall update the UPM when the transaction was successful. | Demonstration |
| 3.8.1.2 | UAS shall cancel the transaction when errors occurred. | Demonstration |
| 3.8.1.2.1 | UAS shall not let the transaction go through if the user does not have sufficient funds. | Demonstration |
| 3.8.1.3 | UAS shall subtract the user’s funds when the user bought asset(s). | Demonstration |
| 3.8.1.3.1 | UAS shall transfer the correct asset(s) to the user when the user bought the asset(s). | Demonstration |
| 3.8.1.4. | UAS shall add funds to the user when the user sold his or her asset(s) to the shop. | Demonstration |
| 3.8.1.4.1 | UAS shall take the correct asset(s) from the user when the user sold the asset(s). | Demonstration |
| **User Experience Submodule (UES)** | | |
| 3.8.2 | UES shall process all experiences given to the user to the UPM. | Demonstration |
| 3.8.2.1 | UES shall have every new user’s experience start at zero. | Demonstration |
| 3.8.2.2 | UES shall calculate the user’s experience required to advance to the next level. | Demonstration |
| 3.8.2.3 | UES shall add experience calculated from the BCS to the user when the user won a battle. | Demonstration |
| 3.8.2.4 | UES shall add experience to the user when the user does any physical training or exercise. | Demonstration |
| 3.8.2.5 | UES shall not ever take away the user’s experience under any circumstances. | Demonstration |
| 3.8.2.6 | UES shall increment the user’s level when he or she has the required experience. | Demonstration |
| 3.8.2.6.1 | UES shall not reset the user’s experience back to zero every time the user advanced a level. | Demonstration |
| **User Skills Submodule (USS)** | | |
| 3.8.3 | USS shall process all user skills related information to the UPM. | Demonstration |
| 3.8.3.1 | USS shall provide the user with basic skills. | Demonstration |
| 3.8.3.2 | USS shall have three types of skills. | Demonstration |
| 3.8.3.2.1 | USS shall have physical skills. | Demonstration |
| 3.8.3.2.1.1 | USS shall drain the user’s endurance or stamina when using physical skills. | Demonstration |
| 3.8.3.2.2 | USS shall have active skills. | Demonstration |
| 3.8.3.2.2.1 | USS shall drain the user’s mana when using active skills. | Demonstration |
| 3.8.3.2.3 | USS shall have passive skills. | Demonstration |
| 3.8.3.2.3.1 | USS shall not drain anything from the user when using passive skills. | Demonstration |
| 3.8.3.3 | USS shall unlock new skills for the user when he or she bought them from the shop. | Demonstration |
| 3.8.3.4 | USS shall unlock new skills for the user when he or she meets the level requirement. | Demonstration |
| 3.8.3.5 | USS shall increment a skill’s level when the user bought an upgrade from the shop. | Demonstration |
| 3.8.3.6 | USS shall make all user changes to his or her skill(s) to be permanent. | Demonstration |
| 3.8.3.6.1 | USS shall not allow the user’s skills to be transferable. | Demonstration |
| 3.8.3.6.2 | USS shall not allow the user’s skills to be refundable. | Demonstration |
| 3.8.3.6.3 | USS shall not allow the user’s skills to be tradable. | Demonstration |
| **User Items Submodule (UITS)** | | |
| 3.8.4 | UITS shall process all user interactions with items to the UPM. | Demonstration |
| 3.8.4.1 | UITS shall set the same maximum number of inventory slots that every user has. | Demonstration |
| 3.8.4.2 | UITS shall show name labels next to every item the user has. | Demonstration |
| 3.8.4.3 | UITS shall show a brief description of every item the user has. | Demonstration |
| 3.8.4.4 | UITS shall add the item to the user’s inventory when the user picked it up. | Demonstration |
| 3.8.4.5 | UITS shall add the correct asset(s) to the user’s inventory when the user bought them from the shop. | Demonstration |
| 3.8.4.6 | UITS shall remove the asset(s) from the user’s inventory when the user sold them to the shop. | Demonstration |
| 3.8.4.7 | UITS shall categorize every equipment the user has to the appropriate body part. | Demonstration |
| 3.8.4.7.1 | UITS shall label every equipment the user has with the appropriate body part. | Demonstration |
| 3.8.4.7.2 | UITS shall not allow the user to equip the equipment to the wrong body part. | Demonstration |
| 3.8.4.7.2.1 | UITS shall only allow one equipment to be equipped at a time for each part of the user’s body. | Demonstration |
| 3.8.4.7.2.1.1 | UITS shall only allow one equipment to be equipped at a time as the user’s headgear. | Demonstration |
| 3.8.4.7.2.1.2 | UITS shall only allow one equipment to be equipped at a time as the user’s eyewear. | Demonstration |
| 3.8.4.7.2.1.3 | UITS shall only allow one equipment to be equipped at a time as the user’s nose wear. | Demonstration |
| 3.8.4.7.2.1.4 | UITS shall only allow one equipment to be equipped at a time as the user’s mouth gear. | Demonstration |
| 3.8.4.7.2.1.5 | UITS shall only allow one equipment to be equipped at a time as the user’s ear wear. | Demonstration |
| 3.8.4.7.2.1.6 | UITS shall only allow one equipment to be equipped at a time as the user’s neckwear. | Demonstration |
| 3.8.4.7.2.1.7 | UITS shall only allow one equipment to be equipped at a time as the user’s arm wear for each arm. | Demonstration |
| 3.8.4.7.2.1.8 | UITS shall only allow one equipment to be equipped at a time as the user’s wrist wear for each wrist. | Demonstration |
| 3.8.4.7.2.1.9 | UITS shall only allow one equipment to be equipped at a time as the user’s finger wear for each finger of each hand of the user. | Demonstration |
| 3.8.4.7.2.1.10 | UITS shall only allow one equipment to be equipped at a time as the user’s torso gear. | Demonstration |
| 3.8.4.7.2.1.11 | UITS shall only allow one equipment to be equipped at a time as the user’s back wear. | Demonstration |
| 3.8.4.7.2.1.12 | UITS shall only allow one equipment to be equipped at a time as the user’s legwear. | Demonstration |
| 3.8.4.7.2.1.13 | UITS shall only allow one equipment to be equipped at a time as the user’s footwear for each foot. | Demonstration |
| 3.8.4.7.2.1.14 | UITS shall only allow at most two weapons to be equipped at a time. | Demonstration |
| 3.8.4.7.2.1.14.1 | UITS shall only allow two weapons to be equipped at a time if the two weapons are lightweight. | Demonstration |
| 3.8.4.7.2.1.14.2 | UITS shall only allow at most one weapon to be equipped at a time if the weapon is heavyweight. | Demonstration |
| 3.8.4.7.2.1.14.3 | UITS shall allow one weapon and one shield to be equipped at the same time. | Demonstration |
| **User Location Submodule (ULS)** | | |
| 3.8.5 | ULS shall process the user’s current location from the LSM to the UPM. | Demonstration |
| 3.8.5.1 | ULS shall record the user’s exact location. | Demonstration |
| 3.8.5.2 | ULS shall translate the user’s exact location to the full street address. | Demonstration |
| 3.8.5.2.1 | ULS shall display the user’s full street address. | Demonstration |
| 3.8.5.3 | ULS shall display a map view with a character icon at the user’s current location. | Demonstration |
| 3.8.5.3.1 | ULS shall display the direction the user is currently facing. | Demonstration |
| **User Statistics Submodule (USTS)** | | |
| 3.8.6 | USTS shall process the user’s statistics to the UPM. | Demonstration |
| 3.8.6.1 | USTS shall display the user’s profile name. | Demonstration |
| 3.8.6.2 | USTS shall display the user’s location. | Demonstration |
| 3.8.6.3 | USTS shall display the user’s current game status. | Demonstration |
| 3.8.6.3.1 | USTS shall display the current game status as In Battle (IB) if the user is in battle. | Demonstration |
| 3.8.6.3.2 | USTS shall display the current game status as Not In Battle (NIB) if the user is not in battle. | Demonstration |
| 3.8.6.4 | USTS shall display the user’s level. | Demonstration |
| 3.8.6.5 | USTS shall display the user’s experience. | Demonstration |
| 3.8.6.6 | USTS shall display the current equipment the user is wearing. | Demonstration |
| 3.8.6.6.1 | USTS shall change the user’s appearance when the user equips any of his or her equipment. | Demonstration |
| 3.8.6.6.2 | USTS shall change the user’s appearance when the user removes or changes any of his or her equipment. | Demonstration |
| 3.8.6.7 | USTS shall display the user’s current health. | Demonstration |
| 3.8.6.8 | USTS shall display the user’s current mana. | Demonstration |
| 3.8.6.9 | USTS shall display the user’s current endurance. | Demonstration |
| 3.8.6.10 | USTS shall display the user’s current stamina. | Demonstration |
| 3.8.6.11 | USTS shall display the user’s strengths. | Demonstration |
| 3.8.6.12 | USTS shall display the user’s weaknesses. | Demonstration |
| 3.8.6.13 | USTS shall display the user’s immunities. | Demonstration |
| 3.8.6.14 | USTS shall display the user’s skills. | Demonstration |

# A. ACRONYMS

**AARB** Advanced Augmented Reality Battle-system  
**AR** Augmented Reality  
**BCS** Battle Calculation Submodule  
**BDS** Battle Data Submodule  
**BIS** Battle Interface Submodule

**BM** Battle Module  
**CPS** Command Parsing Submodule

**DFD** Data Flow Diagram  
**DICS** Data Input Controller Submodule  
**FS** Feedback Submodule  
**GSS** Geographic Scanning Submodule  
**GUI** Graphical User Interface  
**H/W** Hardware  
**HS** Hearing Submodule  
**IB** In-Battle  
**IBS** Information Bus Submodule  
**LSM** Location Service Module  
**LTS** Location Tracking Submodule  
**MCM** Main Control Module  
**MMS** Matchmaking Submodule  
**MPS** Motion Parsing Submodule  
**MRM** Motion Recognition Module  
**NIB** Not In-Battle  
**SM** Shop Module  
**SRD** Software Requirements Document

**SRM** Sound Recognition Module

**SDD** Software Design Document

**SID** Software Implementation Document

**STP** Software Test Plan

**TBD** To Be Done

**TBI** To Be Implemented  
**UAS** User Assets Submodule  
**UES** User Experience Submodule  
**UIS** User Interface Submodule  
**UITS** User Items Submodule  
**ULS** User Location Submodule  
**UPM** User Profile Module  
**USS** User Skills Submodule  
**USTS** User Statistics Submodule

**V&V** Verification and Validation

# B. DATA DICTIONARY

**Augmented Reality**

A live, direct or indirect, view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics, or GPS data.

**Cloud Computing**

Cloud computing refers to the delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients. The name comes from the use of clouds as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts services with a user's data, software, and computation over a network. It has considerable overlap with software as a service (SaaS).

**Role-Playing Game**

A game in which players assume the roles of characters in a fictional setting. Players take responsibility for acting out these roles within a narrative, either through literal acting, or through a process of structured decision-making or character development. Actions taken within many games succeed or fail according to a formal system of rules and guidelines.