**CPSC 319**

**Team 2**



**Dr. Pat Mirenda**

**Software Requirements Specification Document**

## Version: 1.1 Date: (02/10/2006)

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | Wei Lin | Initial Draft | Feb. 6th, 2006 |
| 1.1 | Wei Lin | Final Draft | Feb. 10th, 2006 |
| 1.2 | Wei Lin | Signoff Draft | Feb. 22nd, 2006 |

**Contents**

1. Introduction 4
   1. [Purpose 4](#_TOC_250021)
   2. [Scope 4](#_TOC_250020)
   3. [Definitions, Acronyms, and Abbreviations 5](#_TOC_250019)
   4. [References 6](#_TOC_250018)
   5. [Overview 6](#_TOC_250017)
2. General Description 8
   1. [Product Perspective 8](#_TOC_250016)
   2. [Product Functions 8](#_TOC_250015)
   3. [User Characteristics 10](#_TOC_250014)
   4. [Constraints 11](#_TOC_250013)
   5. [Assumptions and Dependencies 11](#_TOC_250012)
3. Specific Requirements 12
   1. [External Interfaces 12](#_TOC_250011)
      1. [Main Menu 12](#_TOC_250010)
      2. [Registration Subsystem 14](#_TOC_250009)
      3. [Login Subsystem 15](#_TOC_250008)
      4. [Demonstration Subsystem 17](#_TOC_250007)
      5. [Advanced Settings Subsystem 18](#_TOC_250006)
      6. [Game Menu 20](#_TOC_250005)
      7. [View Statistics Subsystem 23](#_TOC_250004)
      8. [Settings Subsystem 25](#_TOC_250003)
      9. [Tutorial Subsystem 27](#_TOC_250002)
      10. [Practice Subsystem 27](#_TOC_250001)
      11. [Game Interface Subsystem 27](#_TOC_250000)
   2. Functions 31
      1. Accounts System 31
      2. Game System 33
      3. Training System 33
      4. User Data System 35
      5. Advanced Settings System 36
   3. Performance Requirement 37
      1. Static numerical requirements 37
      2. Dynamic numerical requirements 37
   4. Logical Database Requirements 37
      1. File Format 37
      2. Accessibility and Security 38
   5. Software System Attributes 38
      1. Reliability 38
      2. Availability 39
      3. Security 39
      4. Maintainability 39
      5. Portability 39
4. Change Management 41
5. Document Approval 42
6. Supporting Information 44
   1. Outline of Section 3.1 44

**1 Introduction**

# Purpose

The purpose of this Systems Requirements Specifications document (henceforth referred to as SRS) is to provide the guidelines necessary to design and implement software that fulfills all the requirements given.

In the case of the Autistic Conversational Skills Software (henceforth simply referred to as ACSS), this document will contain enough information so that in the event that the current team is not able to complete the software, a future team can use this document and only this document to create another version of the software that still fulfills all the requirements.

This SRS will be entirely design-independent, focusing only the end requirements that the final software must achieve. Furthermore, this document can be used by clients and stakeholders as a representation of their requirements for the software. Fulfilling all the requirements of the SRS will be in essence fulfilling the contract with the client.

# Scope

This SRS will cover the software Autistic Conversational Skills Software or ACSS. This name is simply a developmental project name and in no ways reflects the final marketable name for the product.

ACSS will be a tool to help conversationally-impaired users learn and understand the proper protocol and timing of interrupting conversation. These users are children between ages 6-19 who are autistic or have Asperger’s disorder and have difficulty knowing when to interrupt an ongoing conversation

ACSS will provide multimedia examples of real conversations in order to illustrate the two main types of legitimate conversation interruption:

* + 1. An appropriate pause in the conversation. The length of this pause will depend on the rate at which the conversation is taking place (rapid, relaxed, moderate, etc.)
    2. An emergency situation in which immediate attention of one or more of the people involved in the conversation. Although emergencies can be relative, the software will focus on examples of emergencies that are universally recognized (e.g. fire, injury, etc.)

ACSS will provide an interface in which a user must indicate when it is appropriate to interrupt the onscreen conversation with regard to either of the situations presented above. Likewise, ACSS must relay back to the user whether or not the chosen moment of interruption was correct or incorrect, along with a reason and or suggestion.

ACSS will provide a statistical history of correct and incorrect choices along with sub- categories of each. ACSS will provide a profile based on these statistics for at least one main user.

ACSS will NOT provide any examples in which the exceptional clause “excuse me” is required or permitted to interrupt a conversation. This clause is beyond the scope and purpose of the software.

The application of ACSS is as learning software as well as possibly being used in research.

ACSS is targeted towards autistic children of age 6-19 with conversational impairments, specifically difficulties in detecting appropriate ways/times to interrupt an ongoing conversation.

The objective of the software is to simulate a real life conversation and to allow the user the ability to “interrupt” the ongoing conversation. The software will then determine whether or not the interruption choice was appropriate or not and why.

The ultimate goal of the software is to improve the ability of the user to determine how and when to interrupt a real-life conversation based on his/her performance with the software. A secondary goal is to allow an observer to monitor the progress of the user based on statistical data gathered by the software. Any inferences, conclusions, and relevant studies made based on this data will be handled independently and the software is not required to make any opinion nor is it responsible for any that are made.

# Definitions, Acronyms, and Abbreviations

**ACSS** – Autistic Conversational Skills Software, the software that this SRS describes, simulating a real life conversation and allowing users to choose when to interrupt the given conversation

**SRS** – Systems Requirements Software, this document which outlines the requirements that the software must fulfill. Entirely design independent.

**InterruptED** – The temporary internal developmental name for the ACSS, used for the simplicity of labeling windows/title bars, etc. The completed project will be named according to marketing and research needs.

**User** – any person who uses the program, with the general case being children ages 6-19 with autism or having Asperger’s disorder who have conversational difficulties.

**Administrator** – a person who has administrative access to the advanced settings

**GUI** – graphical user interface

**Main Menu Page** – the initial menu to allow users to register/login **Game Menu Page** – the initial menu for users who have logged in **Registration Page** – the page that allows the user to create new accounts

**Login Page** – the page that allows the user to login if they have a valid login

**UML** – Unified Modeling Language. Refer to <http://www.uml.org/>

**OS** – operating system

**SDD** – Software Design Document

**RC** – Release Candidate

**CVS** – Concurrent Versioning System

**Client** – Dr. Pat Mirenda, with possible inclusion of her department and/or research staff.

# References

The PowerPoint on Autism given by Dr. Pat Mirenda (the client) is available at <http://www.umobileco.com/cs319/project/autismintro.06.HO.ppt>

The recorded project description conversation given by Dr. Mirenda is available at [http://www.umobileco.com/cs319/clientmeeting-jan24.mp3.](http://www.umobileco.com/cs319/clientmeeting-jan24.mp3)

The general team website is located at <http://www.umobileco.com/cs319/>

# Overview

Section 2 contains a more in-depth and complete definition of the requirements of the ACSS software. This section is intended for the client and other stakeholders who may or may not be familiar with the actual software engineering process. This section contains little to no technical information regarding the software process necessary to transform this document into an actual piece of software and is instead targeted towards non-software engineering subjects.

Section 3 contains all the technical requirements for the software, including a full list of functional and non-functional requirements. This section is intended to be used by the

software developers in order to create the actual software. The majority of information contained in this section is very technical and field-specific in nature, and thus is not intended to be read or understand by anyone outside of the software engineering field.

**2 General Description**

# Product Perspective

The ACSS is independent and self-contained. An integrated data base will ensure the storage and the retrieval of the user related information and the required video learning session data. It will be the only product on the market that teaches autistic people when to intervene in a conversation. There are other kinds of software for children designed to teach them things like: body parts, animals, spelling and reading, math, and other stuff about the world that they live in. Even though they cover a wide range of interests, none of them are specifically designed for the children with the Asperger’s disorder.

# Product Functions

* + - Register – The user must register once in order to get access to the application; after registering, a unique user name and password will be provided and will allow the logging in of the registered user.
    - Log-in – A registered user has to enter his/her unique user name/password combination in order access the application.
    - Demo – An option available to all users (registered or not) introducing the application and highlighting the main feature of the product.
    - Tutorial – An option available to the registered users allowing to familiarize with the features of the application; the tutorial session will provide for a mock session to be performed by the user (without the score being stored in the record). Pop-up messages will help the user familiarize with the program.
    - Practice – An option available to the registered users allowing practice for a video learning session that will not record the score.
    - Recorded score review – An option allowing the registered user to review his/her score from the video learning sessions. The score will be displayed for maximum the last seven sessions. Each user can only see his/her score.
    - Settings – An option allowing the user to modify the settings for the video learning session such as the level of difficulty and the video category, volume of the video and keyboard settings.
    - Video category and difficulty –The user is able to choose a level of difficulty by selecting the number of seconds in which he/she can interrupt the recorded conversation. Also the user can select the category of the video he/she is going to watch. Depending on what age group the user is in, he/she will have different choices. If the user did not choose anything, a video will be chosen by default.
    - Volume and keyboard settings – The user is able to select the desired volume and assign the key that he/she has to press during the video learning session and the key that is to be pressed for emergency situations (explained below).
    - Video learning session – The user watches a video of an actual conversation between two or more people. The user will be required to press a button on the keyboard when he/she thinks it is all right to intervene in the conversation. To score this correctly, the user has to press the button only when there is pause in the conversation (the duration of the pause will be determined later and it will not necessarily be constant for every video). A warning message will appear on the screen in order to inform the user if the button was pressed at the right time or at the wrong time (right time meaning when there is a pause in the conversation, wrong time meaning when the people from video are still talking). If the pause in the conversation passes without the user pressing the button, some sort of warning will appear on the screen informing the user that he/she could have pressed the button.
    - Bonus learning session (emergency situation) – It is the same as the video learning session, except that from time to time a pop up warning will appear on the screen informing the user that there is an emergency situation and the conversation needs to be interrupted even if the persons are still talking. If the user presses the designated button within 2 seconds from when the warning appears, then the user scores.
    - Reward – The user is able to see the reward after a video learning session is over. A reward video clip (chosen from the settings) will appear on the screen if the user scored 80% or more on the video session. If that is not the case, a message will inform the user that a better score is needed in order to see the reward.
    - Score output per current session – After the user gets his/her reward, a score output will de displayed, including a table (with the number of right answers, wrong answers, and the number of times when he/she should have pressed the button but did not, and depending on the case, the right and wrong times for the emergency situations) and a chart representing the data from the table.
    - Log-out – User logs out in order to let another user to register or log-in.
    - Exit – User exits the program.

The following image shows the different functions described above and their relationships:

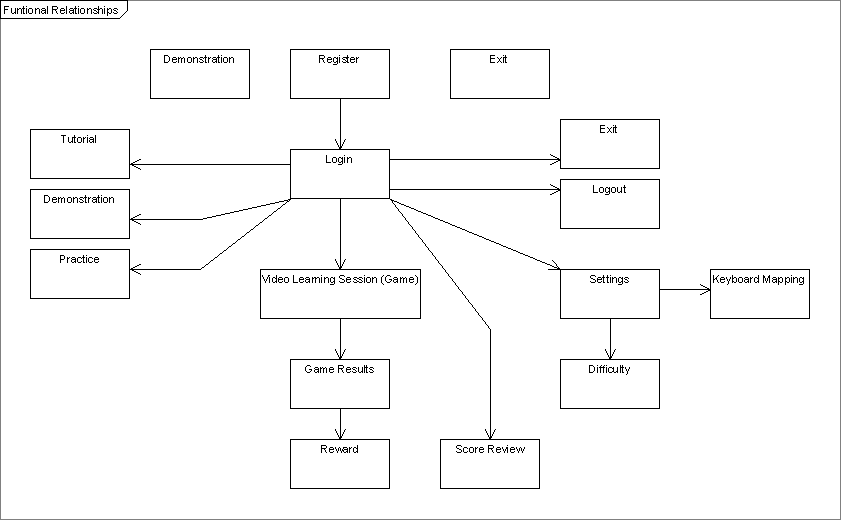


Figure 2.2.1: Functional Relationships The different functions and their relationships

# User Characteristics

* + - Types of users:
      * Persons suffering from the Asperger’s disorder
      * users in the 6-19 age group
    - Technical expertise:
      * All users needs to have minimal technical expertise (need to know how to use the mouse and keyboard)
    - Impact of design by the potential user:
      * For the users in the 6-19 age groups the reward videos need to consist of content of interest.

# Constraints

* + - Platform:
      * must work on Macintosh and PC
    - Operating system:
      * for Macintosh: must have OS X
      * for PC: must have at least Windows 98 or above
    - Hardware:
      * the computer must have an audio output system and speakers or headphones
      * at least 128 MB of RAM
      * minimum 800x600 screen resolution with 256 colors

# Assumptions and Dependencies

* + - Hardware designated for the ACSS needs to have a CD-ROM or DVD-ROM drive.

**3 Specific Requirements**

# External Interfaces

The following subsections will examine the various screens that will represent the ACSS. Each of the following subsections will discuss all the subsystems that collectively represent the ACSS.

### Main Menu

Figure 3.1.1: Main Menu The Main Menu Screen

***3.1.1.1***

|  |  |
| --- | --- |
| **Item** | Create New Account |
| **Purpose** | Allow user to be redirected to the Registration Subsystem (3.1.2) |
| **Input** | Mouse click |
| **Output** | Redirect the user to the Registration Subsystem (3.1.2) |
| **Validity** | Click within the button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.1, item A1 |

***3.1.1.2***

|  |  |
| --- | --- |
| **Item** | Login with Existing account |
| **Purpose** | Allow user to be redirected to the Login Subsystem (3.1.3) |
| **Input** | Mouse click |
| **Output** | Redirect the user to the Login Subsystem (3.1.3) |
| **Validity** | Click within the button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.1, item A2 |

***3.1.1.3***

|  |  |
| --- | --- |
| **Item** | Demonstration |
| **Purpose** | Show the user how to play the game |
| **Input** | Mouse click |
| **Output** | Redirect the user to the Demonstration Subsystem (3.1.2) |
| **Validity** | Click within the button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.1, item A3 |

***3.1.1.4***

|  |  |
| --- | --- |
| **Item** | Advanced |
| **Purpose** | Allow user to be redirected to the Advanced Subsystem (3.1.2) |
| **Input** | Mouse click |
| **Output** | Redirect the user to the Advanced Subsystem (3.1.2) |
| **Validity** | Click within the button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.1, item A4 |

***3.1.1.5***

|  |  |
| --- | --- |
| **Item** | Exit |
| **Purpose** | Allow user to exit the program |
| **Input** | Mouse click |
| **Output** | Ask the user for confirmation  If user confirms, exit program; else, close confirmation screen |
| **Validity** | Click within the button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.1, item A5 |

### Registration Subsystem

Figure 3.1.2 Registration Subsystem The Registration Screen

|  |  |
| --- | --- |
| **Item** | User ID Field |
| **Purpose** | Entry field for the User Login ID |
| **Input** | Keyboard |
| **Output** | User ID typed is displayed in the text field |
| **Validity** | Alphanumeric characters |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.2, item B1 |

***3.1.2.2***

|  |  |
| --- | --- |
| **Item** | Password Field |
| **Purpose** | Entry field for the User’s Login Password |
| **Input** | Keyboard |
| **Output** | User Password will be represented by asterisks |
| **Validity** | Alphanumeric characters |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.2, item B2 |

***3.1.2.3***

|  |  |
| --- | --- |
| **Item** | Submit |
| **Purpose** | Allows the user to Submit their registration request to create a  new account |
| **Input** | Mouse click |
| **Output** | The user is redirected to the Main Menu when the creation of the new account is complete; else, returns to the Registration Subsystem (3.1.2), indicating that the creation of new account failed |
| **Validity** | Click within button |
| **Error Handling** | There should be no empty fields, there should be no existing  User ID present in the User Account Database |
| **Relationships** | Relates to User Accounts File |
| **Reference** | Figure 3.1.2, item B3 |

***3.1.2.4***

|  |  |
| --- | --- |
| **Item** | Cancel |
| **Purpose** | Allows the user to Cancel Registration and returns to Main  Menu (3.1.1) |
| **Input** | Mouse click |
| **Output** | The user is redirected back to Main Menu (3.1.1) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.2, item B4 |

* + 1. **Login Subsystem**

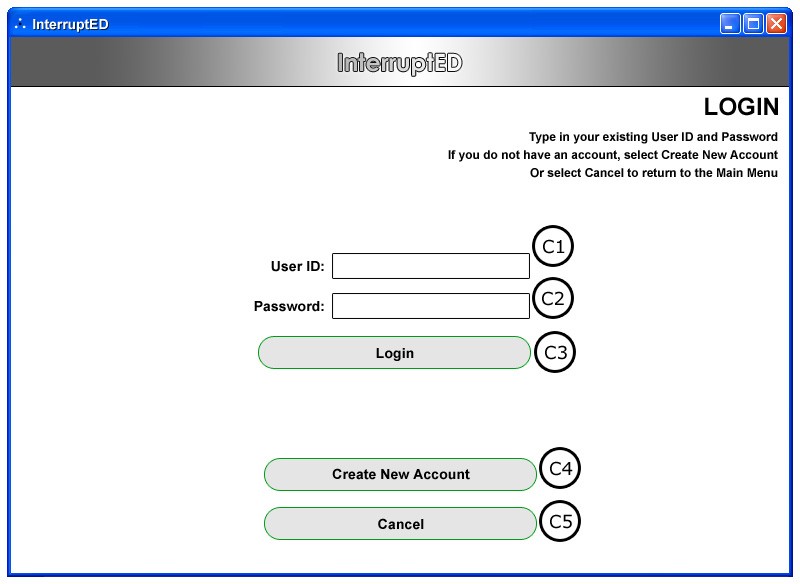


Figure 3.1.3 Login Subsystem The Login Screen

|  |  |
| --- | --- |
| **Item** | User ID Field |
| **Purpose** | Entry field for the User’s Login ID |
| **Input** | Keyboard |
| **Output** | User ID typed is displayed in the text field |
| **Validity** | Alphanumeric characters |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.3, item C1 |

|  |  |
| --- | --- |
| **Item** | Password Field |
| **Purpose** | Entry field for the User’s Login Password |
| **Input** | Keyboard |
| **Output** | User Password will be represented by asterisks |
| **Validity** | Alphanumeric characters |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.3, item C2 |

|  |  |
| --- | --- |
| **Item** | Login |
| **Purpose** | Allows user to Login |
| **Input** | Mouse click |
| **Output** | The user is redirected to the Game Menu (3.1.6) if login was successful; otherwise the Login Subsystem (3.1.3) will be  displayed again (or an error page) |
| **Validity** | Click within the button |
| **Error Handling** | There should be no empty fields, there should be no existing User ID present in the User Account Database |
| **Relationships** | Relates to User Accounts File |
| **Reference** | Figure 3.1.3, item C3 |

|  |  |
| --- | --- |
| **Item** | Create New Account |
| **Purpose** | Allows the User to register a new account |
| **Input** | Mouse click |
| **Output** | The user is redirected to the Registration Subsystem (3.1.2) |
| **Validity** | Click within the button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.3, item C4 |

|  |  |
| --- | --- |
| **Item** | Cancel |
| **Purpose** | Allows the User to Cancel Login and return to the Main Menu (3.1.1) |
| **Input** | Mouse click |
| **Output** | The user is redirected to the Main Menu (3.1.1) |
| **Validity** | Click within the button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.3, item C5 |

#### Demonstration Subsystem

The Demonstration Page will simply be playing a video to show the user how to play the game.

### Advanced Settings Subsystem

Figure 3.1.5 Advanced Subsystem The Advanced Screen

|  |  |
| --- | --- |
| **Item** | Add a new game |
| **Purpose** | Allows the administrator to add a new game |
| **Input** | Mouse click |
| **Output** | The administrator is redirected to a page for adding a new game |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.5, item E1 |

***3.1.5.2***

|  |  |
| --- | --- |
| **Item** | Modify an existing game |
| **Purpose** | Allows the administrator to modify an existing game |
| **Input** | Mouse click |
| **Output** | The administrator is redirected to a page for modifying an  existing game |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.5, item E2 |

***3.1.5.3***

|  |  |
| --- | --- |
| **Item** | Delete a game |
| **Purpose** | Allows the administrator to delete a game |
| **Input** | Mouse click |
| **Output** | The administrator is redirected to a page for deleting a game |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.5, item E3 |

***3.1.5.4***

|  |  |
| --- | --- |
| **Item** | Return to Main Menu |
| **Purpose** | Allows the administrator to return to the Main Menu (3.1.1) |
| **Input** | Mouse click |
| **Output** | The administrator is redirected back to the Main Menu (3.1.1) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.5, item E4 |

* + 1. **Game Menu**

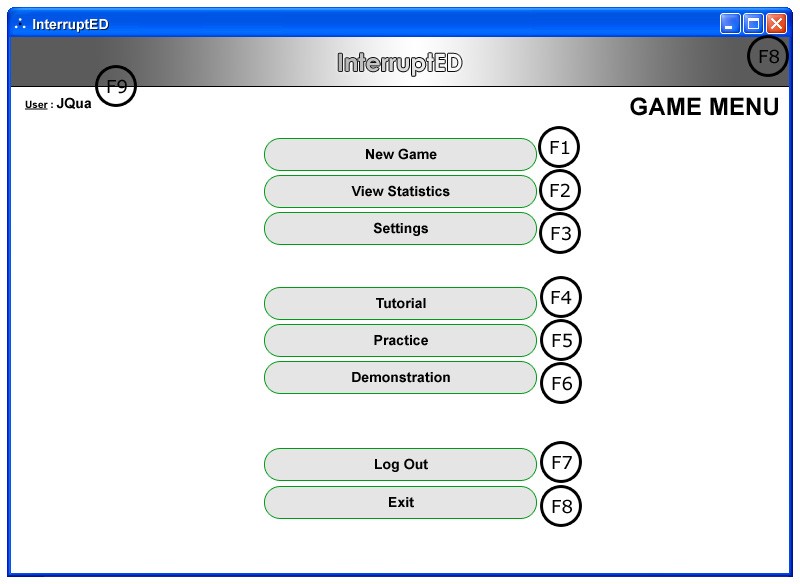
Note: Assumes user has logged in.

Figure 3.1.6 Game Menu The Game Menu Screen

|  |  |
| --- | --- |
| **Item** | New Game |
| **Purpose** | Allows the user to start a new game |
| **Input** | Mouse click |
| **Output** | The user is redirected to a new instance of a game |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to game files (i.e. videos, timing files, setting files) |
| **Reference** | Figure 3.1.6, item F1 |

|  |  |
| --- | --- |
| **Item** | View Statistics |
| **Purpose** | Allows the user to view their statistics |
| **Input** | Mouse click |
| **Output** | The user is directed to the View Statistics Subsystem (3.1.7) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to the User Records File |
| **Reference** | Figure 3.1.6, item F2 |

|  |  |
| --- | --- |
| **Item** | Settings |
| **Purpose** | Allows user to modify game specific settings |
| **Input** | Mouse click |
| **Output** | The user is directed to the game Settings Subsystem (3.1.8) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to the Game Settings File |
| **Reference** | Figure 3.1.6, item F3 |

|  |  |
| --- | --- |
| **Item** | Tutorial |
| **Purpose** | Allows user to view and play the tutorial. |
| **Input** | Mouse click |
| **Output** | The user is directed to the Tutorial Subsystem (3.1.9) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to the Tutorial File |
| **Reference** | Figure 3.1.6, item F4 |

|  |  |
| --- | --- |
| **Item** | Practice |
| **Purpose** | Allow user to proceed to the practice playing the game |
| **Input** | Mouse click |
| **Output** | The user is directed to the Practice Subsystem (3.1.10) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to game files (i.e. videos, timing files, setting files) |
| **Reference** | Figure 3.1.6, item F5 |

|  |  |
| --- | --- |
| **Item** | Demonstration |
| **Purpose** | Show the user how to play the game |
| **Input** | Mouse click |
| **Output** | The user is directed to the Demonstration Subsystem (3.1.4) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to the Demonstration File |
| **Reference** | Figure 3.1.6, item F6 |

|  |  |
| --- | --- |
| **Item** | Logout |
| **Purpose** | Allow the user to logout of their account |
| **Input** | Mouse click |
| **Output** | Logs the user out of their account and direct to the Main Menu  (3.1.1) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.6, item F7 |

|  |  |
| --- | --- |
| **Item** | Exit |
| **Purpose** | Allow user to exit the program |
| **Input** | Mouse click |
| **Output** | Ask the user for confirmation  If user confirms, then logout and exit program; else, close confirmation screen |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.6, item F8 |

|  |  |
| --- | --- |
| **Item** | Login ID |
| **Purpose** | Shows the user that the current session is active with the  displayed Login ID |
| **Input** | None |
| **Output** | Login ID shown on screen |
| **Validity** | User Logged in with the shown Login ID |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.6, item F9 |

### View Statistics Subsystem

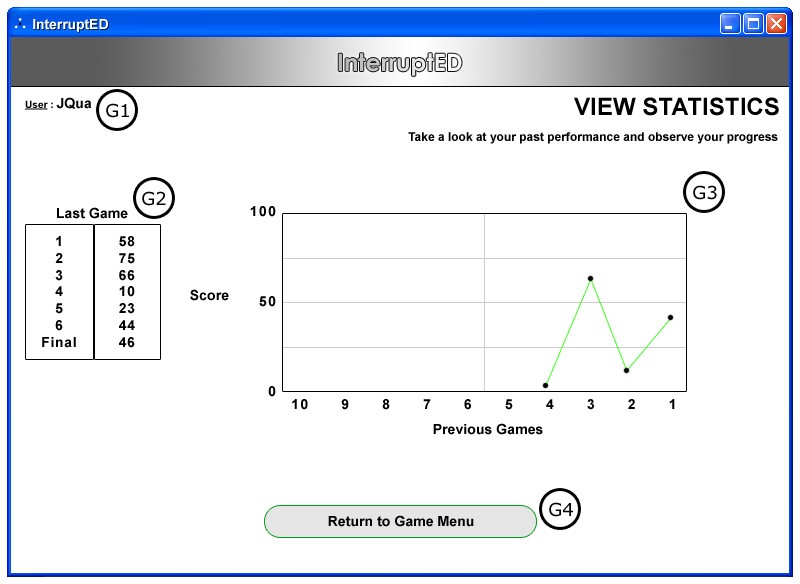


Figure 3.1.7 View Statistics Subsystem The View Statistics Page

|  |  |
| --- | --- |
| **Item** | Login ID |
| **Purpose** | Shows the user that the current session is active with the displayed Login ID |
| **Input** | None |
| **Output** | Login ID shown on screen |
| **Validity** | User Logged in with the shown Login ID |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.7, item G1 |

|  |  |
| --- | --- |
| **Item** | Game History Chart |
| **Purpose** | Shows the User’s performance in their last game |
| **Input** | None |
| **Output** | A data summary chart of User’s last game |
| **Validity** | Non-negative scores in User Records File |
| **Error Handling** | Do not display negative scores |
| **Relationships** | Relates to User Records File |
| **Reference** | Figure 3.1.7, item G2 |

|  |  |
| --- | --- |
| **Item** | Game History Graph |
| **Purpose** | Shows the User’s performance in their last ten games |
| **Input** | None |
| **Output** | A data summary graph |
| **Validity** | Non-negative scores in User Records File |
| **Error Handling** | Do not display negative scores |
| **Relationships** | Relates to User Records File |
| **Reference** | Figure 3.1.7, item G3 |

|  |  |
| --- | --- |
| **Item** | Return to Game Menu |
| **Purpose** | Allows the user to return to the Game Menu (3.1.6) |
| **Input** | Mouse click |
| **Output** | The user is redirected back to the Game Menu (3.1.6) |
| **Validity** | Click within button |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.7, item G4 |

### Settings Subsystem

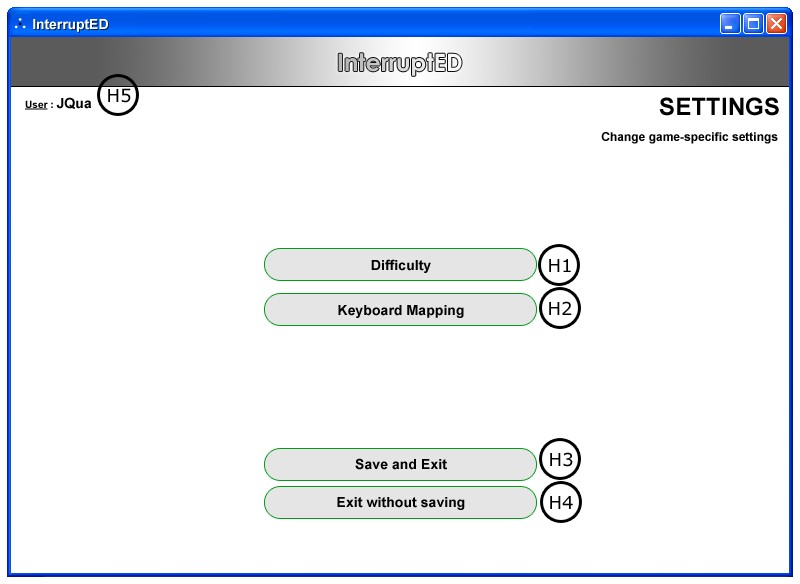


Figure 3.1.8 Settings Subsystem The Settings Page

***3.1.8.2***

|  |  |
| --- | --- |
| **Item** | Difficulty |
| **Purpose** | Allow the user to change the difficulty of the game |
| **Input** | Mouse click |
| **Output** | Change difficulty value in Settings File |
| **Validity** | Easy to Hard |
| **Error Handling** | None |
| **Relationships** | Relates to Settings File |
| **Reference** | Figure 3.1.8, item H1 |

***3.1.8.3***

|  |  |
| --- | --- |
| **Item** | Keyboard Mapping |
| **Purpose** | Allow user to set hotkeys |
| **Input** | Mouse and keyboard |
| **Output** | Set keyboard mapping in the Settings File |
| **Validity** | None |
| **Error Handling** | Prevents duplicate mappings |
| **Relationships** | Relates to Settings File |
| **Reference** | Figure 3.1.8, item H2 |

***3.1.8.4***

|  |  |
| --- | --- |
| **Item** | Save and Exit |
| **Purpose** | Allow the user to save their setting changes and return to the Game Menu (3.1.6) |
| **Input** | Mouse click |
| **Output** | Writes to Settings File and redirects the user back to the Game  Menu (3.1.6) |
| **Validity** | None |
| **Error Handling** | None |
| **Relationships** | Relates to Settings File |
| **Reference** | Figure 3.1.8, item H3 |

***3.1.8.5***

|  |  |
| --- | --- |
| **Item** | Exit without Saving |
| **Purpose** | Allow the user to return to the Game Menu (3.1.6) without  saving the changes |
| **Input** | Mouse click |
| **Output** | Prompts the user to confirm; if yes, redirects the user back to the Game Menu (3.1.6), else close the confirmation window |
| **Validity** | None |
| **Error Handling** | None |
| **Relationships** | None |
| **Reference** | Figure 3.1.8, item H4 |

***3.1.8.6***

|  |  |
| --- | --- |
| **Item** | Login ID |
| **Purpose** | Shows the user that the current session is active with the  displayed Login ID |
| **Input** | None |
| **Output** | Login ID shown on screen |
| **Validity** | User Logged in with the shown Login ID |
| **Error Handling** | There should be no error |
| **Relationships** | None |
| **Reference** | Figure 3.1.8, item H5 |

* + 1. **Tutorial Subsystem**

The Tutorial Page will be simulating a game session combined with a demonstration. As the user is viewing the instructions, the game will prompt the user to participate as if they were playing the game.

### Practice Subsystem

The Practice Page will allow the user to play a game session for a chosen level without any statistics being recorded.

### Game Interface Subsystem

Figure 3.1.11 Game Interface Subsystem The Game Interface

|  |  |
| --- | --- |
| **Item** | Interrupt |
| **Purpose** | Allows the user to indicate when they think it is okay to interrupt  in a conversation |
| **Input** | Keyboard or Mouse click |
| **Output** | The immediate feedback indicators will reflect if the interruption  is correct or incorrect |
| **Validity** | Click within button or assigned keyboard key |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to the Settings File for pre-assigned keyboard hotkeys  Indicates to the system that the user is interrupting |
| **Reference** | Figure 3.1.11, item K1 |

|  |  |
| --- | --- |
| **Item** | Correct Feedback Indicator |
| **Purpose** | Indicates to the user with a blinking star when a correct interruption has been performed. The stars will initially be empty implying how many interruptions there will be. Each star will then be filled upon each correct input. |
| **Input** | None |
| **Output** | Display the correct action animation (blinking star) |
| **Validity** | ø |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to the Interrupt button and Timer File |
| **Reference** | Figure 3.1.11, item K2 |

|  |  |
| --- | --- |
| **Item** | Incorrect Feedback Indicator |
| **Purpose** | Indicates to the user that an incorrect interruption has been  performed |
| **Input** | None |
| **Output** | Display the incorrect action animation |
| **Validity** | ø |
| **Error Handling** | There should be no error |
| **Relationships** | Relates to the Interrupt button and Timer File |
| **Reference** | Figure 3.1.11, item K3 |

|  |  |
| --- | --- |
| **Item** | Video Display Area |
| **Purpose** | The area for displaying video to the user |
| **Input** | None |
| **Output** | Displays the video corresponding to the current level |
| **Validity** | Video file to be played exists |
| **Error Handling** | If video file does not exist, then the current game will exit and  notify user of the error |
| **Relationships** | Relate to video file and timer file |
| **Reference** | Figure 3.1.11, item K4 |

|  |  |
| --- | --- |
| **Item** | Timeline Indicator |
| **Purpose** | The timeline to show the progress of the video. |
| **Input** | None, user cannot move the timeline slider |
| **Output** | Playing Timeline |
| **Validity** | Synchronize with video |
| **Error Handling** | If not synchronized, set the timeline indicator to zero |
| **Relationships** | Relate to video file and timer file |
| **Reference** | Figure 3.1.11, item K5 |

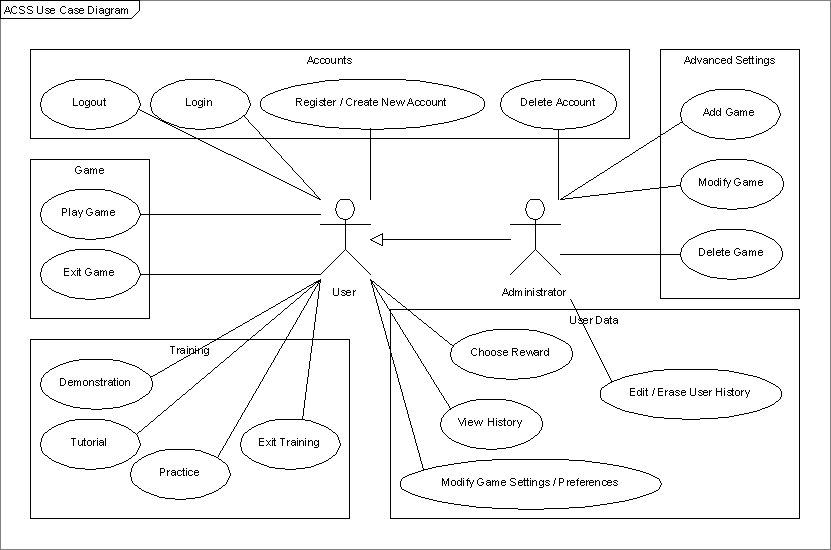
|  |  |
| --- | --- |
| **Item** | Quit Game |
| **Purpose** | Allows the user to quit the current game |
| **Input** | Mouse click |
| **Output** | Ask the user for confirmation  If user confirms, redirect user to the Game Menu (3.1.6); else, close confirmation window |
| **Validity** | Mouse click within button area |
| **Error Handling** | Video will be paused when the confirmation window is presented, and resumes if the confirmation is cancelled |
| **Relationships** | None |
| **Reference** | Figure 3.1.11, item K6 |

|  |  |
| --- | --- |
| **Item** | Exit |
| **Purpose** | Allow user to exit the program. |
| **Input** | Mouse click |
| **Output** | Ask the user for confirmation  If user confirms, exit program; else, close confirmation window |
| **Validity** | Click within button |
| **Error Handling** | Video will be paused when the confirmation window is presented, and resumes if the confirmation is cancelled |
| **Relationships** | None |
| **Reference** | Figure 3.1.11, item K7 |

|  |  |
| --- | --- |
| **Item** | Emergency Indicator |
| **Purpose** | Bonus points can be obtained if the user clicks on the  Emergency icon before it disappears |
| **Input** | Mouse click |
| **Output** | Bonus points added to the user score |
| **Validity** | Click within button |
| **Error Handling** | None |
| **Relationships** | None |
| **Reference** | Figure 3.1.11, item K8 |

# Functions

This section explains all functional requirements from a developer’s point of view. It is categorized by subsystems.



### Accounts System

* + - 1. ***The system should allow users to login.***

|  |  |
| --- | --- |
| **Description** | The system should allow users to login to their account  to continue the learning session with their saved progress |
| **Sequence of Operations** | 1. User must type in User ID and Password in the corresponding text fields 2. User must click on the Login button |
| **Validity Checks** | User must login with the correct User ID and Password |
| **Input** | Registered user id and password, and select “LOGIN” or “CLEAR” |
| **Output** | If “LOGIN” is selected, then verify and continue to Game Menu  If “CLEAR” is selected, then reset the fields |
| **Error Handling** | If user id and/or password are incorrect, then display an error message |

* + - 1. ***The system should allow users to logout.***

|  |  |
| --- | --- |
| **Description** | The system should allow users to logout of their account when necessary. |
| **Sequence of Operations** | 1. User clicks on the Logout button 2. User is logged out |
| **Validity Checks** | User logged in with a valid account |
| **Input** | User selects “LOGOUT” option and a confirmation will be displayed and then user selects either “YES” or “NO”. |
| **Output** | Display a message indicating that the User has logged  out |
| **Error Handling** | ø |

* + - 1. ***The system should allow users to register an account.***

Note: User cannot delete their account

|  |  |
| --- | --- |
| **Description** | The system should allow users to register an account |
| **Sequence of Operations** | 1. User clicks on Create a New Account 2. New Account form is displayed, fields are filled by User 3. User clicks Submit, then the system checks the User Table in the database; if no errors, then add new user into the User Table |
| **Validity Checks** | All user fields are filled out for account creation |
| **Input** | Fill out the following fields: user id, password, age range.  Select the “REGISTER” option |
| **Output** | Display either a success or failure message depending on if the registration was successful or not. Then go to the Login Subsystem (3.1.3) |
| **Error Handling** | If certain fields are not filled out, or the same User ID  already exist, then print out the error message and tell the user to try again. |

* + - 1. ***The system should allow the administrator to delete user accounts***

|  |  |
| --- | --- |
| **Description** | The system should allow the administrator to delete user accounts |
| **Sequence of Operations** | 1. The administrator chooses an account to be  deleted |
| **Validity Checks** | User account exists |
| **Input** | Select an account from a list |
| **Output** | Selected account is deleted |
| **Error Handling** | None |

* + 1. **Game System**
       1. ***The system shall allow the user to play the game.***

|  |  |
| --- | --- |
| **Description** | The system shall allow the user to play the game. |
| **Sequence of Operations** | 1. The user is validly logged in, and is in game mode 2. The game is in session |
| **Validity Checks** | Video must be of valid format, and playing smoothly.  Check if the user has a valid login id. |
| **Input** | Video file |
| **Output** | Playing video on screen. |
| **Error Handling** | If the video is corrupted, then the game will not proceed, and an error message will be shown to the  user |

* + - 1. ***The system shall allow the user to exit the game at any point.***

|  |  |
| --- | --- |
| **Description** | The system shall allow the user to exit the game at any point |
| **Sequence of Operations** | 1. The user is in a game session 2. The user clicks on the Exit Game button 3. The system asks for an Exit Game confirmation 4. The user clicks verifies the confirmation 5. The user exits the game |
| **Validity Checks** | Check if the user is currently in the game.  Check if the user has a valid login id. |
| **Input** | Click on Exit Game button |
| **Output** | Exits the game, back to the main screen. |
| **Error Handling** | ø |

* + 1. **Training System**
       1. ***The system shall allow the user to view a demonstration of how to play the game.***

|  |  |
| --- | --- |
| **Description** | The system shall allow the user to view a  demonstration of how to play the game |
| **Sequence of Operations** | 1. User clicks on the Demonstration button 2. The system enters the Demonstration module |
| **Validity Checks** | Program is running |
| **Input** | User clicks on the Demonstration Button |
| **Output** | Demonstration video is played. |
| **Error Handling** | If the video is corrupted, then the game will not proceed, and an error message will be shown to the user |

* + - 1. ***The system should allow the user to practice playing the game.***

|  |  |
| --- | --- |
| **Description** | The system should allow the user to practice playing the game. |
| **Sequence of Operations** | 1. User clicks on the Practice button 2. The system enters the Practice module |
| **Validity Checks** | Check if the user is logged in |
| **Input** | User clicks on the Practice Button. |
| **Output** | Practice game is in session. |
| **Error Handling** | If the video is corrupted, then the game will not proceed, and an error message will be shown to the user |

* + - 1. ***The system should allow the user to view and play the tutorial.***

|  |  |
| --- | --- |
| **Description** | The system should allow the user to view and play the  tutorial. |
| **Sequence of Operations** | 1. User clicks on the Tutorial button 2. The system enters the Tutorial module |
| **Validity Checks** | Check if the user is logged in |
| **Input** | User clicks on the Tutorial Button. |
| **Output** | Tutorial is in session. |
| **Error Handling** | If the video is corrupted, then the game will not  proceed, and an error message will be shown to the user |

* + - 1. ***The system should allow the user to exit the training system at any time***

|  |  |
| --- | --- |
| **Description** | The system should allow the user to exit the training  system at any time |
| **Sequence of Operations** | 1. User is in the training system (i.e. demonstration, tutorial, or practice) 2. User clicks exit |
| **Validity Checks** | User is in the training system |
| **Input** | User clicks on the exit button |
| **Output** | User is returned to either the Game Menu (3.1.6) or Main Menu (3.1.1) depending on if the user is logged  in |
| **Error Handling** | None |

* + 1. **User Data System**
       1. ***The system may allow the user to view their performance history.***

|  |  |
| --- | --- |
| **Description** | The system may allow the user to view their performance history. |
| **Sequence of Operations** | 1. The user clicks on the View History button 2. The system shows the user history on screen |
| **Validity Checks** | Check if the user is logged in |
| **Input** | User clicks on the View History button |
| **Output** | User history is displayed |
| **Error Handling** | None |

* + - 1. ***The system may allow the administrator to edit/erase user records.***

|  |  |
| --- | --- |
| **Description** | The system may allow the administrator to edit/erase user records |
| **Sequence of Operations** | 1. The administrator clicks on the edit/erase user records button 2. The system brings up the edit/erase user records screen |
| **Validity Checks** | Check if the administrator is logged in |
| **Input** | Administrator clicks on the edit/erase user history button |
| **Output** | User data is edited/erased |
| **Error Handling** | User has no history. The system shows “No History”  on the screen |

* + - 1. ***The system may allow the user to modify their game settings***

|  |  |
| --- | --- |
| **Description** | The system may allow the user to modify their game settings |
| **Sequence of Operations** | 1. The user clicks on the settings button 2. The system displays the game settings |
| **Validity Checks** | Game Settings File exists |
| **Input** | User clicks on the game settings button |
| **Output** | The system redirects the user to the game settings page |
| **Error Handling** | Creates a default Game Settings File if it does not exist |

* + - 1. ***The system may allow the user to choose a reward.***

|  |  |
| --- | --- |
| **Description** | The system may allow the user to choose a reward |
| **Sequence of Operations** | 1. The user will indicate the reward they want |
| **Validity Checks** | Check if the user is logged in |
| **Input** | User selects the reward |
| **Output** | Reward is chosen and set in User Settings File |
| **Error Handling** | None |

* + 1. **Advanced Settings System**
       1. ***The system may allow the administrator to add levels / games***

|  |  |
| --- | --- |
| **Description** | The system may allow the administrator to add levels / games |
| **Sequence of Operations** | 1. The administrator clicks on the Advanced Settings button 2. The administrator clicks on the add levels / games button |
| **Validity Checks** | Check if the game is valid |
| **Input** | User indicates the game to be added |
| **Output** | If game is valid, it is added into the system, else user is notified that the add was unsuccessful |
| **Error Handling** | Notify user if game was unsuccessful in being added |

* + - 1. ***The system may allow the administrator to modify levels / games***

|  |  |
| --- | --- |
| **Description** | The system may allow the administrator to modify levels / games |
| **Sequence of Operations** | 1. The administrator clicks on the Advanced Settings button 2. The administrator clicks on the modify levels / games button 3. The administrator selects the level / game to be modified |
| **Validity Checks** | Level / game exists |
| **Input** | User indicates the level / game to be modified |
| **Output** | After modifications; if game is still valid, it is updated in the system, else user is notified that the update was unsuccessful |
| **Error Handling** | Notify user if game was unsuccessful in being modified |

* + - 1. ***The system may allow the administrator to delete levels / games***

|  |  |
| --- | --- |
| **Description** | The system may allow the administrator to delete levels  / games |
| **Sequence of Operations** | 1. The administrator clicks on the Advanced Settings button 2. The administrator clicks on the delete levels / games button 3. The administrator selects the level / game to be deleted |
| **Validity Checks** | Level / game exists |
| **Input** | User indicates the level / game to be deleted |
| **Output** | Level / game deleted |
| **Error Handling** | None |

* 1. ***Performance Requirement***

This section describes the expected performance requirements. This is an estimation of the system, and all the numerical values may vary depends on how large the final application is.

### Static numerical requirements

1. The system shall support only one terminal.
2. The system shall support only one simultaneous user on each machine; however, it shall support multiple users to create personal accounts and access the system on the same machine at different times.
3. The system shall run on both Mac (OS 10.\*) and Windows (Windows 98 and above) machines with at least 128 MB of memory and a CD or DVD ROM.
4. The video clips attached to the system shall play on both QuickTime and Windows Media Player with the necessary codec.

### Dynamic numerical requirements

1. The system shall be loaded and functioning within 15 seconds 95% of the time after starting the application.
2. Each account shall be stored and activated within 5 seconds after creation.
3. Each video clip (less than 5 minutes) shall be loaded completely within 30 seconds 95% of the time
4. Each user input during the session shall be responded to (a simple sign which indicates if the timing of interruption is correct or incorrect) within 3 seconds 98% of the time.
5. Each sessional grade report shall be generated within 5 seconds at the end of each session 95% of the time.

# Logical Database Requirements

### File Format

The system must store all the user account information as well as the sessional grade records. All the data shall be stored in text-based flat files. For each user account, the login ID, name, password, age, email address (optional) shall be stored in one file. The email address gives the user option to receive any further information or update about the software. Each attribute shall be delimited by a semicolon, and all the entries shall be sorted alphabetically by the login ID. Furthermore, for each user account, there shall be a grade report file which contains every sessional grade result for a certain user. A grade report file shall contain the following attributes: session ID, date (dd/mm/yyyy), duration (minutes), video ID, number of pauses, and number of correct timings. Each entry shall also be delimited by a semicolon and sorted alphabetically by the session ID. Here is an example which shows how the data shall be stored and presented.

\*Note: the file format and type may vary when the system is being developed.

#### File: UserAccounts.txt

**Format: Account ID;Name;Password;Age;Email**

[AccountA;G. Jonse;1234567;10;gjones@email.com](mailto:Jonse%3B1234567%3B10%3Bgjones@email.com) AccountB;S. [Dickens;9876545;12;sdickens@mail.com](mailto:Dickens%3B9876545%3B12%3Bsdickens@mail.com) AccountC;S. Herns;0101010;8;null

.

.

.

\*Note: the account ID in this file is unique, and the email address is optional

**File:AccountB\_grades.txt**

**Format: Session ID;Date;Duration;Video ID;Number of pauses;Number of correct timing**

001;08/11/2006;4.34;023;5;3

002;12/03/2007;5.00;007;10;7

003;11/05/2007;3.04;021;5;5

.

.

.

**\***Note: the session ID in this file is unique.

* + 1. **Accessibility and Security**

Only the user who has access to a certain account can access the grade report file belongs to that account. None of the users has access to modify any of the data files.

# Software System Attributes

### Reliability

The system will not crash on invalid data files or input files. If video files are played in a format that is not recognized, the system will ask the user to recode the file in the correct format or the system will produce an error (refer to section 3.5.4 Maintainability for adding data files). The data files will be stored on a nonvolatile storage device such as a hard drive, so that the data files can be retained when the system is shut off. The system

will check the validity of the user files when they are loaded. If there is an error the system will ask the user to make a new account. If an error occurs in the data file when the movie is being played, the system displays an error message and stops the lesson and displays the lesson results. If there is a fatal error the system shuts down without crashing the computer it is running on.

### Availability

The system will only run infrequently. The system will allow the user to restart the application after a crash. All data beyond the last save point will be lost. The user will be able to load his or her data file after the system has been restarted and continue using the system. The system will have an average run time of 15 to 20 min per session depending on the user, although the user may use the system longer.

### Security

The system will use the computer’s default operating system security. The system will not use any of its own security features since the software is not web-based and it is not going to be run over a network. The system will be contained on one computer. The data files will not be encrypted since the data stored in these files should be easily modifiable and readable in case the system is used for research. The system checks the validity of the data files when it is running and if the data is invalid the system outputs an error to the user. The system will not keep log files of the User’s usage.

### Maintainability

The researcher will be able to update the system with new video content and data files. The system will ask the user to specify a directory where the videos will be stored. To add a video it must be put in this directory and there must be a corresponding data file for the video. The corresponding data file will contain information about the pauses in the video, where the user can interrupt into the conversion and the skill level of the video. The product will be built using components that are as independent as possible to make the system easily modifiable. All components of the system will be modular and be as independent as possible. Each component will have a different function: GUI, the video player and the data file reader/writer. Users will be able to add new accounts. The system will allow the user or the researcher to add and change the rewards for lessons.

### Portability

The product will be able to run on Mac OS X and Windows. The software will be written in a platform independent programming language for portability; there will be no platform specific code. We will write all the software using Java. All the video files will

be in the MPEG1 format that can be played back on Mac OS X and Windows. We will use the Java Media Framework for support of the playback of video files. The system’s data files will be portable between Mac and Windows. The user will have to install the required video playback codec separately if it is not already installed on the computer they are running the system on although this is unlikely since most computers will have an MEG1 codec already installed. The system will also require that the Java Runtime Environment (JRE) Version 5.0, Update 6 be installed on the computer.

**4 Change Management**

Any requests to change the project scope and requirements shall be discussed by all the members of the team 2Communiate. A change will be made only when the majority of the team and the project manager, Jeffrey Qua (jlqua99@hotmail.com) agree on the change. In this case, the SRS document shall be updated by the team members in order to reflect the changes, and a date of change shall be noted in the file. If this change request is made by the client or anyone outside of the team, he or she will have to contact the team communication manager, Wei Lin (kode\_slinger@hotmail.com) via email. If a change request is made by a team member, he or she can raise it during the weekly team meeting or contact other team members via email.

All the contact information of each team member can be obtained from the team website at [http://www.umobileco.com/cs319/.](http://www.umobileco.com/cs319/)

**5 Document Approval**

By signing this document, I approve the content of this SRS document.

Client:

Name (printed): Signature: Date:

## Professor:

Name (printed): Signature: Date:

## TA:

Name (printed): Signature: Date:

## Project Manager/Web Master:

Name (printed): Signature: Date:

## Minutes/Configuration Manager:

Name (printed): Signature: Date:

## Communication/Research/Training Manager:

Name (printed): Signature: Date:

## Risks Manager:

Name (printed): Signature: Date:

## Software Version Control Manager:

Name (printed): Signature: Date:

## Process Manager:

Name (printed): Signature: Date:

**6 Supporting Information**

# 6.1 Outline of Section 3.1

### Main Menu

* + 1. **Registration Subsystem**

### Login Subsystem

### Demonstration Subsystem

### Advanced Settings Subsystem

### Game Menu

### View Statistics Subsystem

### Settings Subsystem

### Tutorial Subsystem

### Practice Subsystem

### Game Interface Subsystem