Project Requirements

Andrew Maywapersaud, Jinho Hwang, Rajvi Chavada

Table of Content

1. Introduction Page 2
   - 1.1 Purpose of this document
   - 1.2 Scope
   - 1.3 Definitions, acronyms

2. Overall Description Page 3-2
   - 2.1 Product Perspective
   - 2.2 User Characteristics
   - 2.3 Project Constraints
3. Specific Requirements Page 4
   - 3.1 External Interface Requirements
     - 3.1.1 User Interface
   - 3.2 Functional Requirements Page 5
     - 3.2.1 The user
   - 3.3 Performance Requirements Page 6-7

Introduction

This section explains the purpose and scope of Braille program as well definitions, acronyms, and abbreviations with its references. The document was shaped in the format similar to this[1]

1.1 Purpose of this Document

The following document is created to:

1. explain what the project and the Braille program is for
2. track down the requirements set by customers
3. track down the references used
4. show use cases of the project
5. show test cases of the Braille program
6. get an approval to Braille program project

1.2 Scope

The "Authoring Appt" is a visually-impaired educator assistant application which provide an easy way for the educators to create scenarios to provide an entertaining braille learning experience to the students the educators are teaching. This program will provide the educators tools to import, export, create, and edit the scenario and import custom voice lines.

1.3 Definitions, Acronyms, Abbreviations

Term	Definition
Scenario	A custom made case study for the users to create to provide an entertaining educational experience to Braille learners. The scenario file is written in a script format.
Desc	Description
Dep	Dependency
AF	Alternative Flow
EX	Exception
RAT	Rational

2. Overall Description

The following section will give a breakdown of the project. The breakdown will explain different the interactions between classes and functions internally. The basic functionality will be explained, and a description of user and implementer interaction will be explored. Finally, the constraints and assumptions for the project will be presented.

2.1 Product Perspective

A shallow description of the project can be stated by a device that allows kids, including visually impaired kids, to learn how to read braille. The device will display characters/words to the user who then respond to the question by pressing buttons. The main system is a software to help educators to create these scenarios and questions.

Essentially, the main system is an authoring app that must be usable by visually impaired users. The authoring app will provide facilities and functions to create the flow of the scenario where the user can ask questions and receive answers. A function for the user (educator) to record, save, and upload the audio will be given. Furthermore, there will be a facility implemented to save the scenario in an appropriate format and then test the scenario using provided software

2.2 User Characteristics

There are essentially 2 types of users that can use the application: those visually impaired, and those who are not. Each of these users cannot navigate through the application in the same way as one another and the application must be compatible for both types. A screen reader for the visually impaired will be provided for them to use the authoring app.

2.3 Project Constraints

Scope: Essentially the scope of this app is to create a working authoring app that enables the user (visually impaired and not visually impaired) to create scenarios. One possible constraint includes not being able to fully deliver the app to the visually impaired users. The app should include all features compatible for all users and should be easily accessible to these users. The app should be easy to navigate and create scenarios for them.

Risk: The authoring app is to connect to an external hard drive that displays the braille cells for the children to learn. There is a risk of miscommunication between the software and hardware and a risk of data being lost in the transmission arises.

Quality & Time: Given the timeframe for the project, the team must ensure that the quality delivered matches the original idea. The quality of the project should not be compromised in

relation with time, the team must ensure that the functions and features promised are also delivered.

3. Specific Requirements

This section contains, highlights, and explains all the functional and quality requirements of the system.

3.1 External Interface Requirements

3.1.1 User Interfaces

Essentially there are two types of users who will be interacting with the graphical interface: educator, and a visually impaired educator. The interface must be implemented in such a way that both users find ease in navigating through the app. For visually impaired users, a screen reader function will be implemented that will allow them to map out different buttons and fields of the app.

There will be five main windows of the app:

1. The main menu: Here the user can enter the scenario editor, record voice, and exit the program.
2. The voice recorder: Here the user can view all the voice file lists, record voice, delete selected voice, play selected voice, load sounds, and exit the menu.
3. The scenario editor: In this window the user will have be able to view the scenario list and have access to: create a new scenario, edit selected scenario, save selected scenario, load scenario, and run selected scenario functions.
4. The scenario maker: This window will open up once the user selects "create a new scenario." In this window, the user can create their own scenario and edit existing scenarios as well. The user can set the file name, the number of pins and buttons, and have the ability to create and remove commands and save the file.
5. Run scenario: When the user selects a scenario to run, a window will open up that uses the provided testing software to run the scenario they have created in a visual player mode or an audio player mode.

3.2 Functional Requirements

3.2.1 The User

3.2.1.1 Functional Requirement 1.1

ID: FR1
Title: Create a new scenario(s)
Desc: The user should be able to create and implement their own scenario(s).
Dep: None

3.2.1.2 Functional Requirement 1.2

ID: FR2
Title: Save created scenario(s)
Desc: The user should be able to save their created scenario in the scenario list or as a file with specified format.
Dep: FR1

3.2.1.3 Functional Requirement 1.3

ID: FR3
Title: Edit Existing Scenarios
Desc: The user should be able to select and edit an existing scenario from the scenario list.
Dep: There should be existing scenarios.

3.2.1.4 Functional Requirement 1.4

ID: FR4
Title: Load Scenarios
Desc: The user should be allowed to navigate through their directory and load selected scenario to add to scenario list.
Dep: User's directory

3.2.1.5 Functional Requirement 1.5

ID: FR5
Title: Run Scenarios
Desc: The user should be able to select a scenario and run it.
Dep: FR1

3.2.1.6 Functional Requirement 1.6

ID: FR6
Title: Record user audio
Desc: The user should be able to record their own audio in an appropriate file format. Also, the user should be able to utilize the screen reader if they do not want to record

their own audio.
Dep: Screen reader

3.2.1.7 Functional Requirement 1.7
ID: FR7
Title: Screen Reader
Desc: For the users that are visually impaired, the feature of a screen reader should me implemented and can be accessed in a easy way.
Dep: Type of user

3.2.1.8 Functional Requirement 1.8
ID: FR8
Title: Audio player vs Visual player
Desc: The user should have an option to run the selected scenario as an audio player or visual player if they are visually impaired.
Dep: Type of user, FR5

3.2.1.8 Functional Requirement 1.9
ID: FR9
Title: Hotkeys
Desc: Visually impaired user mostly uses keyboard to navigate and to assist them, adding hotkeys to each feature must be done.
Dep: FR1 to FR8 and FR10

3.2.1.8 Functional Requirement 1.10
ID: FR10
Title: Logger
Desc: To know which feature user mostly use, key logger must be implemented.
Dep: FR1 to FR9

3.3 Performance Requirements

This requirement section outlines the performance specifications that the user can expect from the software.

3.3.1 Easy Navigation to Scenario Editor

ID: QR1
Title: Easy Navigation to editor
Desc: The user should be able to easily find and navigate to the scenario editor easily from the main menu.
Rat: In order for the user to easily access the editor window.

3.3.2 Usage of Scenario Lists

ID: QR2
Title: Usage of the scenario list
Desc: The scenarios displayed in scenario list should be outputted in a user-friendly way and easy to understand. Selecting a scenario from the list should only take 1 click.
Rat: In order for the user to use the list view easily.

3.3.3 Usage of Scenario Maker

ID: QR3
Title: Usage of the scenario maker
Desc: The functions and features of the scenario maker should have a user-friendly layout that is easy to navigate and use to conduct various tasks (i.e create command, remove command, save, etc). Features and functions should not be hard to find.
Rat: In order for the user to use the scenario maker easily.

3.3.4 Pre-existing scenario deliverability

ID: QR4
Title: Usage of the scenario editor to modify scenarios.
Desc: Pre-existing scenarios that have been created or loaded should be displayed in a user-friendly way in the scenario list and the features to modify the scenarios should be easily accessible and easy to find.
Rat: In order for the user to modify scenarios.

3.3.5 Usage of Audio Recording

ID: QR5
Title: Usage of the audio recording.
Desc: The steps taken to record user's audio should be simple to use and easy to save upon recoding completion.
Rat: In order for the user to easily record their audio.

3.3.6 Usage of Screen Reader

ID: QR6
Title: Usage of the screen reader
Desc: The visually impaired users should be able to navigate and turn of the screen reader feature easily and without any hassle. Also, the screen reader should allow the user to navigate through the authoring app without any difficulties.
Rat: In order for visually impaired users to use the app.