Braille Project Requirements Document

Table of Contents

1. Introduction

This section explains the purpose and scope of Braille program as well deifintions, acronyms, and abbreviations with its refereces.

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 the use case of the Braille program 5. show test cases of the Braille program
6. get an approval to Braille program project

1.2 Scope

The "Braille Project" 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, and 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 ritten in a script format.
Desc	Description
Dep	Dependency
AF	Alternative Flow
EX	Exception

1.4 References

[1] Geagea, S., Zhang, S., Shalin, N., Hasibi, F., Hameed, F., Rafiyan, E. and Ekberg, M. (2018). Software Requirements Sepcification. [ebook] chalmers, p.2. Available at: http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf [Accessed 4 Feb. 2018].

L. Brandenburg, "How to Write a Use Case", Bridging-the-gap.com, 2018. [Online]. Available: http://www.bridging-the-gap.com/what-is-a-use-case/. [Accessed: 23- Feb- 2018].

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 imapred and not visually impaired) to create scenarios. One possible constraint includes not being able to fully deliever 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 quaulity delivered matches the original idea. The quality of the project should not be comprimised 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

3.3 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-freindly 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.

4. Use cases

This section will outline the different use cases and outline requirements specification that will capture how user(s) will interact with the authoring app and achieve a specific goal. The section will describe a step by step process a user will go through to complete and accomplish a task or goal

4.1: User 1-Educator

The authoring app is designed for two users:
- Visually impaired user
- Not a visually impaired user

4.1.1 User Case 1

Name: Make a scenario
Brief Description: The user wants to create a scenario.
Actor: Educator
Preconditions: User has entered the scenario editor from the main menu.
Basic Flow:
1. User selects the "Create a new scenario" button
2. System redirects the user to the scenario maker window
3. User enters the name of this new scenario and selects whether it will be saved in the scenario list and/or another file location
4. User enters the number of braille cells and the number of buttons for this scenario.
5. User can begin making a scenario suing the create/remove command options and place the

commands above, below, or replace them.
6. Once scenario is complete, user can save the scenario and/or choose to exit and be redirected to the scenario editor page.

Alternative Flow:
AF1: User wants to edit an existing saved scenario in the scenario list.
1. User selects the scenario from the scenario list and clicks "Edit Selected Scenario" 2. System redirects to the scenario maker window 3. Return user to basic flow step 4
AF2: User wants to upload a scenario from their directory and edit that
1. User selects "Load Scenario" and finds the scenario they want to edit in their directory 2. The selected scenario is displayed in scenario list 3. User selects the uploaded scenario and and clicks "Edit Selected Scenario" 4. Return user to basic flow step 4

Exception Flows:
EX1: System fails in saving scenario to scenario list
1. System notifies user that an error has occured 2. Return user to basic flow step 3

EX2: User uploads incorrect scenario file format/missing file 1. System reminds user of what files will be accepted/that the file does not exist.
2. Return user to AF2 step 1

EX3: User enters incorrect number of cells or number of buttons
1. System notifies user they have entered an incorrect value
2. Return system to basic flow step 4

Post Conditions:
The user has created a new/existing file ready to be run.

4.1.2 User Case 2

Name: Run a scenario
Brief Description: The user wants to run a scenario.
Actor: Educator/tester
Preconditions: Scenario's need to have been already created without any errors, and user has enterred the scenario editor window
Basic Flow:
1. User selects a scenario from the scenario list
2. User clicks "Run Scenario" button
3. System redirects and runs the selected scenario

Alternative Flow:
AF1: The user wants to run an uploaded scenario 1. User clicks "Load Scenario," system redirects page allowing user to find the scenario file in their directory
2. System displays the scenario in the scenario list
3. Redirect user to basic flow step 1

Exception Flows:
EX1: Scenario file in scenario list or directory is not saved in the appropriate format.
1. System displays error message informing the user that the system does not support the format
2. Redirect user to basic flow step 1.

Post Conditions:
The selected scenario file runs.

4.1.2.1 User Case 2.1

Name: Run a scenario
Brief Description: The user wants to run a scenario.
Actor: Visually Impaired tester/educator
Preconditions: Scenario's need to have been already created without any errors, and user has entered the scenario editor window
Basic Flow:
1. User selects a scenario from the scenario list
2. User clicks "Run Scenario" button
3. System redirects and ask user if they want to run as an audio player or visual player
4. User clicks on "audio player" button
5. System begins to run scenario with a screen reader.

Alternative Flow:
AF1: The user wants to run an uploaded scenario 1. User clicks "Load Scenario," system redirects page allowing user to find the scenario file in their directory
2. System displays the scenario in the scenario list
3. Redirect user to basic flow step 1

Exception Flows:
EX1: Scenario file in scenario list or directory is not saved in the appropriate format.
1. System displays error message informing the user that the system does not support the format
2. Redirect user to basic flow step 1.

Post Conditions:
The selected scenario file runs for the visually impaired user.

4.1.3 User Case 3

Name: Record Audio **Brief Description:** The user wants to record audio.
Actor: Educator **Preconditions:** User has entered the voice recorder stage from the main menu.
Basic Flow:
1. User clicks "Record Voice" 2. System prompts user to enter the name of their new recording
3. System saves the voice.wav file in the voice file list and prompts user to start recording and end recording 6. User has options to delete selected voice file, play selected voice file, load external sounds, and exit to main menu.
Alternative Flow:

AF1: User Wants to load an existing audio file
1. User selects "Load Sound" and finds the appropriate format file in their directory.
2. Redirect user to basic flow step 4.

AF2: User wants to delete an audio recording.
1. User selects the audio file to be deleted from the audio file list
2. User clicks on "Delete Selected"
3. Redirect user to basic flow step 4.

AF3 User wants to play an audio
1. User selects audio file they want to play from the audio file list.
2. User clicks "Play sound"
3. System playsback the audio file to user
4. Redirect user to basic flow step 4

Exception Flows:
EX1: Audio file that user enters contains illegal symbols/text 1. System displays error message stating incorrect file name
2. Redirect user to basic flow step 2

EX2: Selected audio file to load is in incorrect format 1. System displays unsupported file format error 2. Redirect user to AF1 step 1 **Post Conditions:**
The user successfully records audio.

5. Test case

Execution Process	Expected Result	Pass or Fail
Create Scenario is pressed	A window with the scenario maker opens up	Pass
Edit a scenario from scenario list	Selected scenario opens up in scenario editor once edit is pressed	Pass
Load a scenario	Window opens up allowing user to select a scenario which appears in scenario list	Pass
Save selected scenario	Window opens up to save the selected scenario in local directory	Pass
Run Selected Scenario	Window redirects and asks user to run as an audio player or visual player and runs scenario upon selection	Pass
Remove Selected	System removes the selected scenario from the scenario list	Pass
Create Command in scenario maker	System opens a dialogue box allowing user to select command type and the first and second argument	Pass
Remove a command in scenario maker	Selected command is deleted	Pass
Move it up/Move it down in scenario maker	Vertically shifts the command according to direction selection	Pass
Record Voice	Prompts user to enter file name and begins recording once button is clicked	Pass
Delete Selected/Play Selected	Deletes or plays back the selected scenario respectively	Pass

Execution Process	**Expected Result**	**Pass or Fail**
Load Sound	App opens up user directory for user to select the audio file they wish to load	Pass