|  |
| --- |
| York University |
| Design Document |
| EECS 2311 – Authoring App Group 6 |

|  |
| --- |
| Abarna Kucheri Subburaman  11-30-2018 |

Table of Contents

[A. Introduction 2](#_Toc531241135)

[1. Summary 2](#_Toc531241136)

[2. Objective 2](#_Toc531241137)

[3. High-Level Requirements 3](#_Toc531241138)

[4. Deliverables 3](#_Toc531241139)

[5. Assumptions 3](#_Toc531241140)

[B. Project description 4](#_Toc531241141)

[1. About the Project 4](#_Toc531241142)

[2. Use Cases 4](#_Toc531241143)

[3. Acceptance Test Cases 5](#_Toc531241144)

[4. High-Level Timeline/Schedule 6](#_Toc531241145)

[5. References 7](#_Toc531241146)

Software Development Project  
Design Document

Version 2.0

Project: The Authoring App

Date: November 30, 2018

Prepared and Reviewed by: Abarna Kucheri Subburaman

# A. Introduction

## Summary

|  |  |
| --- | --- |
|  | This document discusses the design of the project and provides high level structure for the development of this software system. It will cover the basic uses of each class and their role in the software. This is a technical document which provides an insight into the program’s design and is primarily intended for a reader with software development knowledge. |

## Objective

|  |  |
| --- | --- |
|  | The Authoring App is built to be an accessible app aiding educators and providing them an opportunity to be inclusive of the hearing and visually impaired students. The objective of this app is to replicate a braille cell and its features in the virtual realm. We replicate the braille cell using Treasure Box Braille which is a software that teaches students on how to read Braille. This app lets educators create question and answer scenarios to play in class and have the software give the corresponding message such as right or wrong when the students answer the questions by pressing buttons. The app is made accessible to accommodate a wide range of users. The design structure of this app is described in the sections below and is intended to provide programmers an insight into the use of this software. |

## High-Level Requirements

|  |  |
| --- | --- |
|  | Based on the needs specified by the client, we have developed list of requirements necessary for a successful project. |

The app must satisfy the following requirements:

* Program must support accessibility features in order to be used through screen readers by the visually impaired users
  + The app should work seamlessly with screen readers
* Program must support at least two operating system platforms
  + The app supports Windows and Linux platforms
* Program must have a user-friendly interface
* Users should be able to use the program at ease, that is, users should be able to freely add, edit and delete actions in a scenario.
* Program must run without errors and host error logs with messages to indicate user is doing something wrong
* Program must maintain a simple format for user to see their input actions meanwhile still maintain the correct format for the scenario in order to play it through TBB

## Assumptions

|  |  |
| --- | --- |
|  | There were a few underlying assumptions made in the design process of the program. These assumptions were made to provide user with the most optimal experience. |

The following assumptions were made in order to narrow down the criteria of requirements for the project:

* User is familiar with the technology necessary to use this application
* User is fluent in the English language
* User is running software on one of the following platforms: Windows, Linux
* User understands the concept of braille cell and how to work with one
* User is comfortable working with and searching through different directories

# B. Project design description

## About the Project

|  |  |
| --- | --- |
|  | The idea for this project was brainstormed as an alternative for Treasure Box Braille. The Treasure Box Braille (TBB), is a device which displays words or characters in Braille and the users respond through the buttons. Given that TBB is still a relatively new technology, it is not very affordable for the average user. From the educators’ perspective, it is extremely expensive for the school to invest in a huge quantity of TBB to support full sized classes. The school and/or the organization’s budget can restrain educators from utilizing this device and incorporate it into their classrooms. This project provides an affordable, user-friendly app which educators, average user or visually-impaired, can use teach their students Braille. |

## Attributes

|  |  |
| --- | --- |
|  | This project consists primarily of components such as JPanels, JButtons and JLists. In order to provide the user an interactive as well as professional clean looking interfaces, the user interfaces were built with WindowBuilder in Eclipse. The most important attribute in the interfaces in this software is the JList. This allowed the user to modify, add and remove items while still viewing instant updates on the interfaces providing the user an ease of access. Another attribute to be highlighted is the accessibility properties for each of the features to also give the visually impaired users the ease of access. |

## Class Diagrams

|  |  |
| --- | --- |
|  | Each action in the classes interlinks the various classes to flow through the design process at ease. Using labels for prompts clears any potential ambiguity for the user when entering inputs. The main menu redirects to different classes when an action is performed. There are three different choices and they can be seen in the class diagram below. |

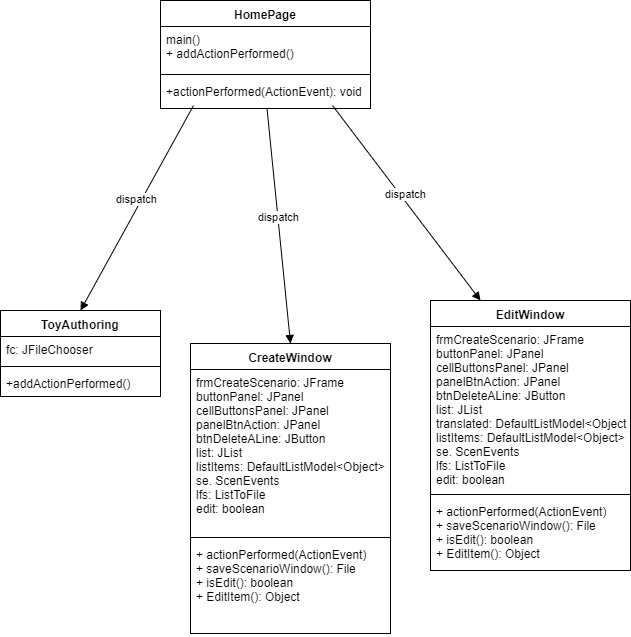
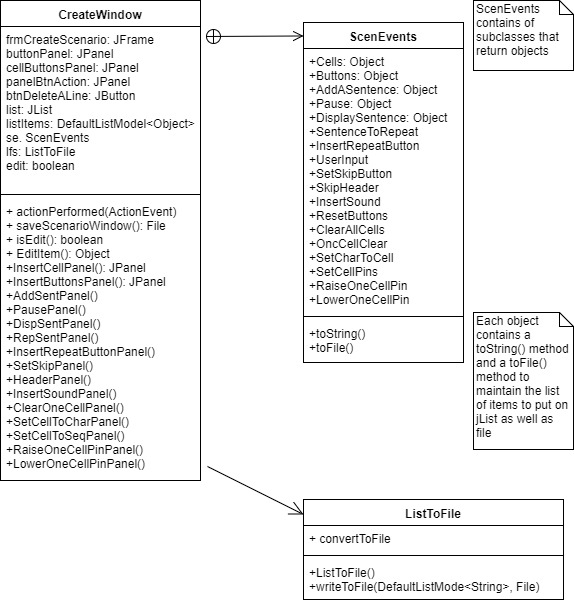


Figure 1: Class Diagram for HomePage

|  |  |
| --- | --- |
|  | The Create Scenario button action and Edit Scenario button action essentially follow the same sequence of design process but the only difference is that the Edit Scenario button includes a Translator class in the process to load the scenario file. |



## Sequence Diagram

|  |  |
| --- | --- |
|  | The following diagram showcases a sequence for one button action event in the CreateWindow. This sequence can be applied to all the buttons in the CreateWindow interface. |

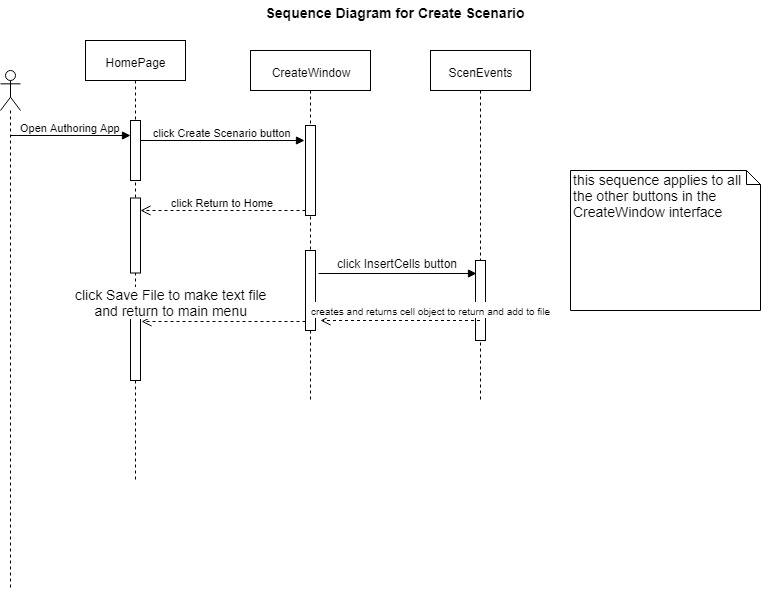


Figure 2: Sequence Diagram for InsertCells event