Halloween Hideout

Taylor Rosby

CST-452 Capstone Project Final Architecture & Design

Grand Canyon University

Instructor: Professor Mark Reha

Revision: 1

Date: April 4, 2020

**ABSTRACT**

Halloween Hideout is a 2D iOS platformer centered around surviving Halloween night, from the hours of 7 pm to 7 am. The game will consist of three (3) levels, each level will take place at different times in the game will get progressively harder. Throughout the player’s journey, they will collect candy pieces with a value ranging from one (1) to fifteen (15). These candy pieces will be used in the in-game store to buy costumes that will randomly appear throughout the levels. The player will be able to run, jump, and attack to clear the levels and complete the game.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 11.29.20 | Tay Rosby | Initial draft for review/discussion |
| 12.2.20 | Tay Rosby | Update Logical Solution Design |
| 2.7.21 | Tay Rosby | Update UML |
| 3.7.21 | Tay Rosby | Update UML |
| 3.22.21 | Tay Rosby | Update Enemies and Playable Characters |
| 4.1.21 | Tay Rosby | Update Logical Solution Design, Key Technical Design, Software Technologies Removed ER Diagram, Data Dictionaries, and DDL Scripts |
| 4.4.21 | Tay Rosby | Update Flow Chart, and General Technical Design Added GitHub Links |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Design Overview 4

Detailed High-Level Solution Design 5

Detailed Technical Design 8

Appendix A – Technical Issue and Risk Log 25

Appendix B – References 26

Appendix C – External Resources 27

Design Introduction

In this document is a high-level design of Halloween Hideout. The design includes mock-ups of the user-interface, flowcharts, and other diagrams that show the logic for the game. There is also playable character, costume, and enemy designs along with their actions.

Detailed High-Level Solution Design

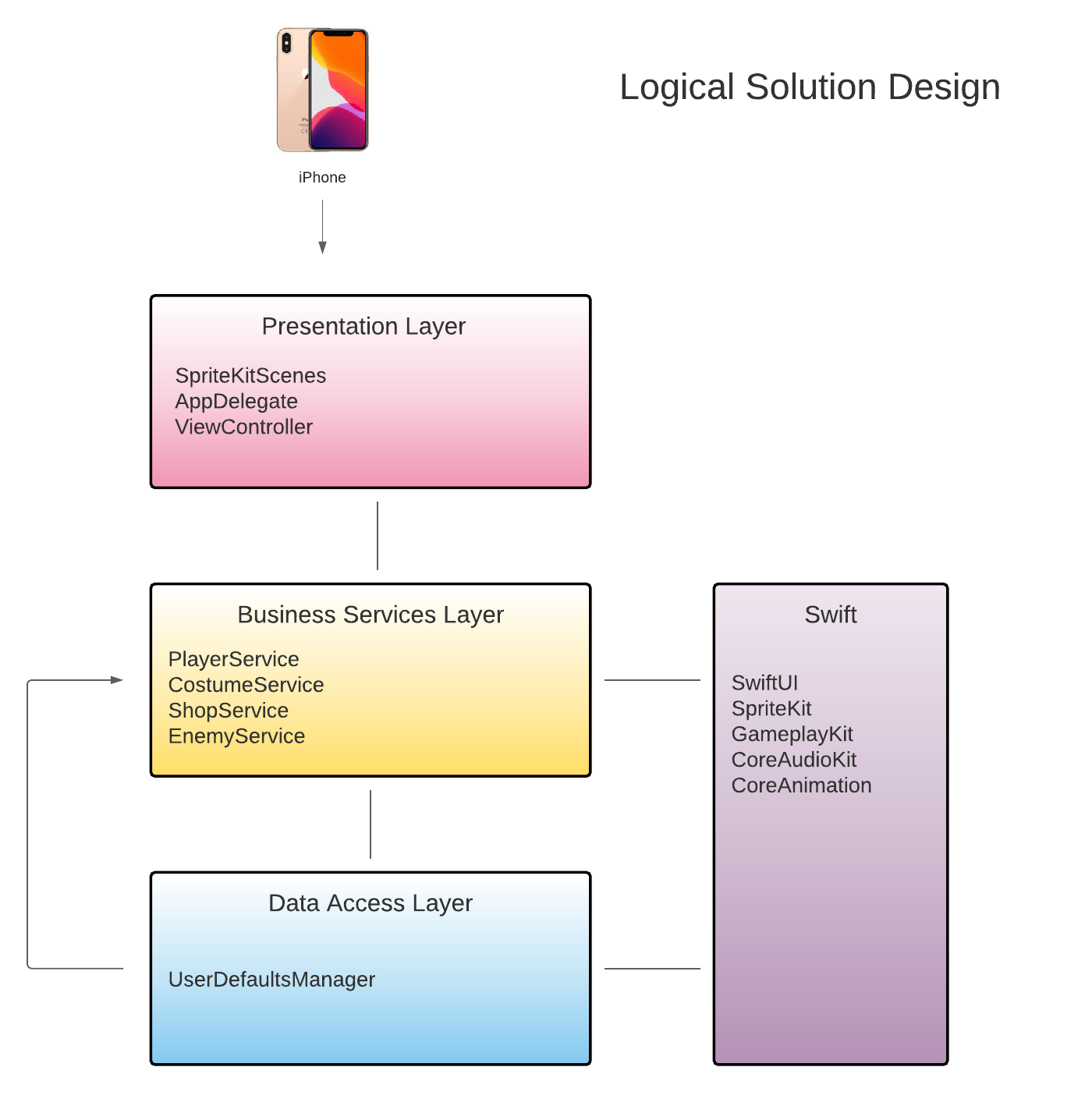
Below is a detailed overview of how Halloween Hideout will be created. This overview includes diagrams, configurations, wireframes, and explanations about design choices. There is also a list of the hardware and software technologies that will be used along with descriptions of proof-of-concept applications. The proof-of-concept applications were done in order to help make technical decisions of the application.

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | |  |
| **Description** | **Rationale** | **Results** |
| 1. Hello World Application | Used to check if the tooling was functioning properly | Learned the basics of running an application in Swift |
| 1. War Game Application | To learn how to connect the view to the back end | Learned to connect the storyboard to the ViewController, how to edit the storyboard to fit various screen sizes and basic Swift |
| 1. Matching Card Game Application | Learn how OO Principles apply in Swift | Learned OO Principles in Swift, how to animate objects, and how to add sounds to objects |
| 1. To Do List Application | Learn how CoreData works | Learned how to setup an application with CoreData, how to add CoreData to an application without it, learned how the CRUD functions worked |
| 1. Ninja Game Application | Learn SpriteKit and GameplayKit | Learned how to programmatically configure storyboards, how to add a player in a specific spot, and how to have a moving camera, |
| 1. Space Game Application | Validate knowledge learned in previous applications | Learned how to add enemies to the screen while animated |

|  |
| --- |
| Hardware and Software Technologies |
| 1 - XCode v. 12.1 |
| 2 - Swift v. 5.3 |
| 3 - SwiftUI v. 5.3 |
| 4 - SpriteKit v. 5.3 |
| 5 - CoreAudio v. 5.3 |
| 6 - CoreAudioKit v. 5.3 |
| 7 - CoreAnimation v. 5.3 |
| 8 – UserDefaults v 5.3 |
| 9 - GameplayKit v. 5.3 |
| 10 – OSLog v. 5.3 |

**Logical Solution Design:**

Below is a diagram of the logical design for the application. The diagram describes how the application will interact with itself, along with some libraries that will be used in the development of the application.



**Physical Solution Design:**

Below is a diagram of the physical design for the application. The diagram shows an example of the device the application will run on and the specifications for the device.

A picture containing graphical user interface

Description automatically generated

Detailed Technical Design

**General Technical Approach:**

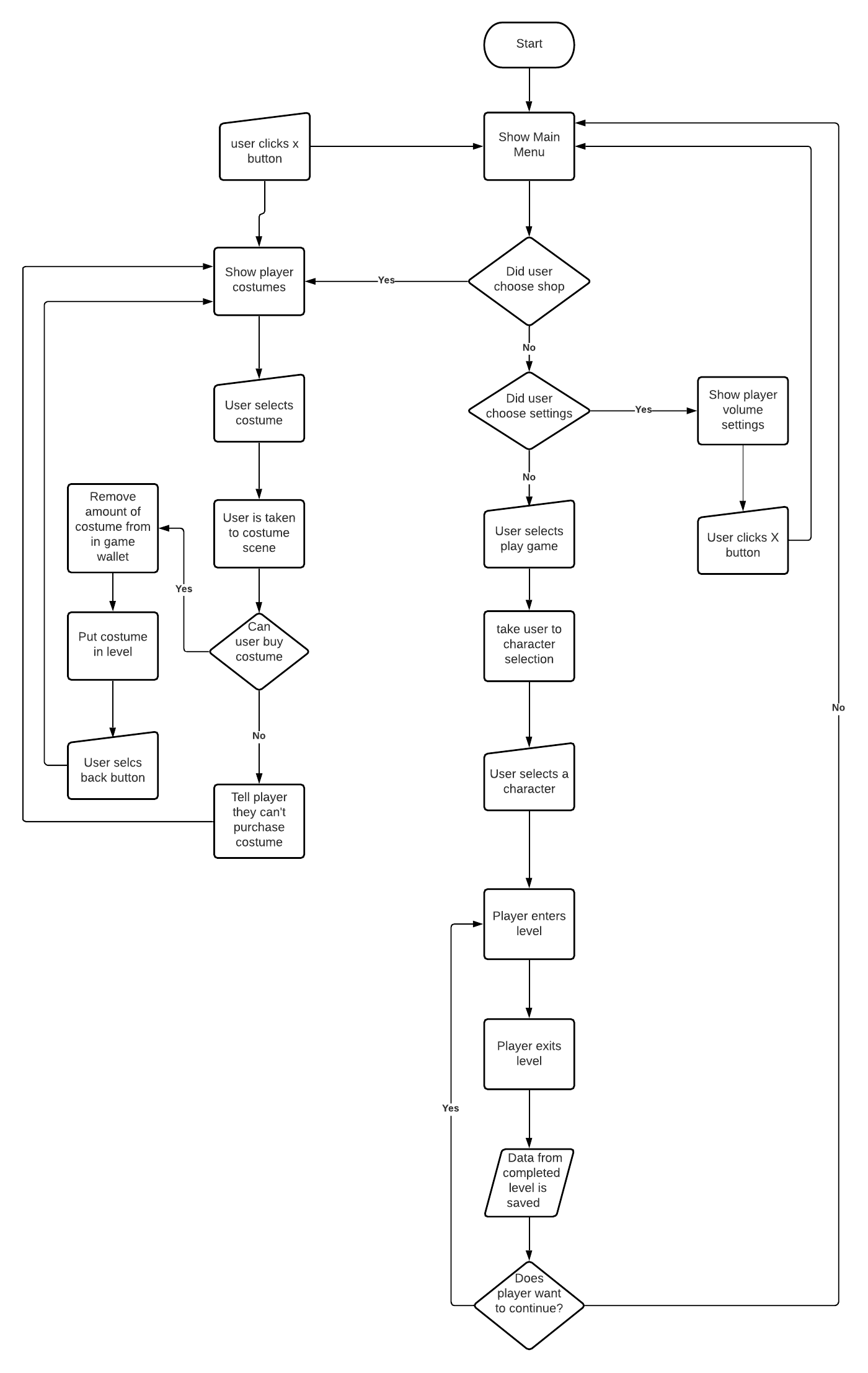
The Halloween Hideout project is a 2D iOS platforming game. The game will allow the user to pick a character from a selection of three and play through levels. In these levels the user and the enemies will have the ability to run, jump, and attack. The player will be given buttons at the bottom of the screen allowing them to do these actions. The user can also collect candy which will increase their in-game wallet or candy bag by 5, 10, or 15. The candy can be used in the in-game store to purchase a variety of costumes. These costumes will the user an advantage in the levels, will appear at random in the levels, and will appear on the character until their time limit ends. After each level, the game will save and update the user data. The application will be stored on the user’s device and will utilize the internal database.

**Key Technical Design Decisions:**

The CoreAnimation library will be used to animate the player, enemies, and other parts of the application that will need to be animated. The CoreAudio and CoreAudioKit libraries will be used to add background music, sound effects, and allow the user to change the volume of the audio. SwiftUI will be used to create the storyboards. UserDefaults will be used to manage the save data. The GameplayKit and SpriteKit will be used to manage the sprites and game logic. OSLog will be used to log application events for debugging. These libraries and frameworks were chosen to make the development process easier and utilize the tools given within the language.

**Flow Charts/Process Flows:**

Below is a flow chart showcasing the overall flow of the game along with a UML sequence diagram showcasing in detail the process of buying a costume from the in-game store.



Chart, box and whisker chart

Description automatically generated

**Sitemap Diagram:**

N/A

**User Interface Diagrams:**

Provided are User Interface diagrams showing the main hub of the game, storyboarding for a level in the game, and character, enemy, and costume designs.

|  |  |
| --- | --- |
| Graphical user interface, bar chart  Description automatically generated | |
| The first-page user will see upon opening the application. One button will start the game, another will take them to the costume shop when unlocked, and the final button will go to the settings page | |
| A picture containing chart  Description automatically generated | |
| The settings page will allow the user to adjust the sound effects volume, background music volume, or the master volume. They can get back to the previous screen by clicking the back button. | |
| Graphical user interface  Description automatically generated | |
| This screen is the costume shop. Each costume will be placed in the image placeholder. They can scroll through the costumes by using the arrow on the bottom right. They can return to the previous screen by clicking the button in the bottom left. | |
| Graphical user interface, diagram  Description automatically generated | |
| This popup will appear when the user clicks on a costume. It shows another image of the costume, a description of the costume, a button to purchase the costume, and a button to return to the rest of the shop. | |
| Chart  Description automatically generated | |
| This is the character selection screen. The user will select a character and be taken to the first game screen. | |
| A picture containing chart  Description automatically generated | |
| In this scene, the user has been placed in the level. There is a candy to collect and an enemy to defeat. | |
| A picture containing bar chart  Description automatically generated | |
| The user collects the candy by jumping on the block, giving them 10 pieces of candy in their in-game wallet. | |
| Chart, bar chart  Description automatically generated | |
| The user jumps off the block and approaches the enemy. | |
| Chart, histogram  Description automatically generated | |
| The user defeats the enemy and makes it to the end of the level. | |
| Enemies   * Will need to be hit 1 time to be defeated   Movement   * Run * Jump * Attack * Idle * Die * Hurt | A picture containing light, traffic, looking, dark  Description automatically generatedA close up of a sign  Description automatically generated |
| Costumes   * Provide varying buffs for 5 seconds   + Attack up, invincibility. Invisibility * Costumes can be bought in the costume shop * Costumes cost different amounts of candy * Will be placed into level once purchased by the player | A close up of a device  Description automatically generatedA picture containing dark, table, holding, person  Description automatically generatedA picture containing clock  Description automatically generated |
| Playable Characters   * Each character has the same stats and abilities * The player gets to choose what their character’s base costume is   Movement   * Run * Jump * Attack * Idle * Die * Hurt | A picture containing clock, dark, sitting, large  Description automatically generatedA picture containing dark, lit, person, holding  Description automatically generatedA close up of a logo  Description automatically generated |

**UML Diagrams:**

Below is the UML diagram for the Halloween Hideout project. The diagram shows how the application classes will be organized as well as how they will interact with one another.

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

Diagram, schematic

Description automatically generated

**Service API Design:**

N/A

**NFR’s (Security Design, etc.):**

Xcode has a feature where the constraints of images, buttons, etc. can be changed depending on the device that is being used. The height and width of each phone is separated into two categories: compact (C) and regular (R). Xcode also has a feature where the developer can see what their application looks like on multiple devices and the height and width category for each device. Using both of these features, the NFR will be handled by changing the constraints of the objects in the storyboard and checking the look of the application to ensure everything is placed for best access for the user.

**Operational Support Design:**

The project will support logging by using OSLog and unified logging. A log message will be put at the beginning, end, and any other exit point of each method. The message will have either an info, debug, error, or fault log level.

**Other Documentation:**

N/A

Appendix A – Technical Issue and Risk Log

Below are the list of issues and risks involved in completing the application.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| I/R | What is the issue or risk? | How will this impact scope, schedule, and cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

Appendix B – References

Beginner Lesson Resources. (2019, September 04). Retrieved from https://codewithchris.com/beginner-youtube/

How to Make an App for Beginners (2020). (n.d.). Retrieved from https://www.youtube.com/playlist?list=PLMRqhzcHGw1ZkH8RuznGMS0NZs0jWQQ5a

Inc., A. (n.d.). Swift.org. Retrieved from https://docs.swift.org/swift-book/

Technologies. (n.d.). Retrieved from https://developer.apple.com/documentation/technologies

Appendix C – External Resources

|  |  |
| --- | --- |
| **GIT URL:** | *https://github.com/tayrosby/HalloweenHideout*  *https://github.com/tayrosby/HalloweenHideoutPortfolio* |
| **Hosting URL:** | *N/A* |