**Alpha Tiles: Ethiopia**

**System Design Document**

****

**Version 1.0**

**10/02/2022**

**Table of Contents**

****

**Table of Contents**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | **Introduction ..............................................................................................................** | | | **1** |
|  | **1.1** | **Purpose of the SDD ..........................................................................................** | | **1** |
| **2. General Overview and Design Guidelines/Approach ...........................................** | | | | **2** |
|  | **2.1** | **General Overview .............................................................................................** | | **2** |
|  | **2.2** | **Assumptions/Constraints/Risks ........................................................................** | | **2** |
|  | **2.2.1** | | **Assumptions ...............................................................................................** | **2** |
|  | **2.2.2** | | **Constraints .................................................................................................** | **2** |
|  | **2.2.3** | | **Risks ...........................................................................................................** | **3** |
| **3.** | **Design Considerations ............................................................................................** | | | **4** |
|  | **3.1** | **Goals and Guidelines ........................................................................................** | | **4** |
|  | **3.2** | **Development Methods & Contingencies ...........................................................** | | **4** |
|  | **3.3** | **Architectural Strategies .....................................................................................** | | **4** |
| **4. System Architecture and Architecture Design ......................................................** | | | | **6** |
|  | **4.1** | **Logical View ......................................................................................................** | | **6** |
|  | **4.2** | **Hardware Architecture ......................................................................................** | | **6** |
|  |  | |  |  |
|  |  | |  |  |
|  | **4.3** | **Software Architecture ........................................................................................** | | **7** |
|  |  | |  |  |
|  |  | |  |  |
|  | **4.4** | **Information Architecture ....................................................................................** | | **8** |
|  | **4.5** | **Internal Communications Architecture ..............................................................** | | **8** |
|  | **4.6** | **Security Architecture .........................................................................................** | | **9** |
|  | **4.7** | **Performance .....................................................................................................** | | **9** |
|  | **4.8** | **System Architecture Diagram ...........................................................................** | | **9** |
| **5.** | **System Design .......................................................................................................** | | | **10** |
|  | **5.1** | **Business Requirements ..................................................................................** | | **10** |
|  | **5.2** | **Database Design ............................................................................................** | | **10** |
|  | **5.2.1** | | **Data Objects and Resultant Data Structures ............................................** | **10** |
|  | **5.2.2** | | **File and Database Structures ...................................................................** | **10** |
|  | **5.3** | **Data Conversion .............................................................................................** | | **11** |
|  | **5.4** | **User Machine-Readable Interface ..................................................................** | | **11** |
|  | **5.4.1** | | **Inputs ........................................................................................................** | **11** |
|  | **5.4.2** | | **Outputs .....................................................................................................** | **12** |
|  | **5.5** | **User Interface Design .....................................................................................** | | **12** |
| **6.** | **Operational Scenarios ...........................................................................................** | | | **13** |
| **7.** | **Detailed Design ......................................................................................................** | | | **14** |

**List of Figures**

****

|  |  |  |  |
| --- | --- | --- | --- |
|  | **7.1** | **Hardware Detailed Design ..............................................................................** | **14** |
|  | **7.2** | **Software Detailed Design ...............................................................................** | **14** |
|  | **7.3** | **Security Detailed Design .................................................................................** | **15** |
|  | **7.4** | **Performance Detailed Design .........................................................................** | **16** |
|  | **7.5** | **Internal Communications Detailed Design ......................................................** | **16** |
| **8.** | **System Integrity Controls .....................................................................................** | | **17** |
| **Appendix A: Acronyms ...............................................................................................** | | | **19** |
| **Appendix B: Glossary .................................................................................................** | | | **20** |
| **Appendix C: Referenced Documents ........................................................................** | | | **21** |

**List of Figures**

**No table of figures entries found.**

**List of Tables**

**Table 1 - Record of Changes**

**Table 2 – Acronyms and Glossary** **19**

**Table 3 – Referenced Documents**  **20**

**Table 4 - Signatures** **21**

**Introduction**

1. **Introduction**

The System Design Document (SDD) describes how the functional and nonfunctional requirements recorded in the Requirements Document transform into more technical system design specifications from which the system is built. The SDD documents the high-level system design and the low-level detailed design specifications.

The SDD describes design goals and considerations, provides a high-level overview of the system architecture, and describes the data design associated with the system, as well as the human-machine interface and operational scenarios. The high-level system design is further decomposed into low-level detailed design specifications for each system component, including hardware, internal communications, software, system integrity controls, and external interfaces.

**1.1** **Purpose of the SDD**

The SDD documents and tracks the necessary information required to effectively define architecture and system design in order to give the development team guidance on the architecture of the system to be developed. Design documents are incrementally and iteratively produced during the system development life cycle, based on the particular circumstances of the information technology (IT) project and the system development methodology used for developing the system. Its intended audience is the project manager, project team, and development team. Some portions of this document, such as the user interface (UI), may be shared with the client/user, and other stakeholders whose input/approval into the UI is needed.

1. **General Overview and Design Guidelines/Approach**

**2.1** **General Overview**

The design of this project will be based on the curriculum from the Software Engineering II class. The process of designing this project will be to create a series of detailed documents that specify what we want the project to look like, and based off of those designs, the structure of the project will be formed.

**2.2** **Assumptions/Constraints/Risks**

**2.2.1** **Assumptions**

In this project, we are assuming that the software of this project is mainly coded in JAVA and XML. The code will be implemented on Android devices exclusively, therefore the IDE that will be used to develop this project will be Android Studio. The hardware for this project includes a mix of both Window and Apple devices for development, and Android devices for testing. Further specifications below.

**2.2.2** **Constraints**

Most of the constraints for this project have to do with the already created constraints of the Alpha Tiles game. For example, the color palette, the images, tile keyboard design and game visual formatting will all come from the already existing games.

As far as logic constraints, each level will only allow the user to have a certain number of lives before they lose. The user also has to use a limited number of letters from the keyboard, since too many will be longer than the word that they are guessing, and too short won’t be enough. The number of lives also must be based on the number of letters within the language as well.

**2.2.3** **Risks**

One of the main risks with the proposed design is the integration of different languages into the program. Since we are only testing this program with the English language, the results of this program may vary depending on what language is implemented. Ideally all of the languages should be compatible.

****Another risk that has been mentioned in previous documents for this project, is the risk that this project will not work for every Android device. Since there is such a large plethora of devices available in the Android community, there is no way for us to test our project on every single one. Therefore, we cannot really release this product with the ability to have tested it on all devices.

**Design Considerations**

****

1. **Design Considerations**

****

An issue that needs to be addressed as soon as possible is the ability of our program to be implemented onto an Android device. Building the android device would be useless if we are not first able to run the initial configuration on an Android device. Another issue that needs to be addressed is the compatibility of the old code from Alpha Tiles with our project. Since a lot of the logic and source code for our game will be based on the already existing code from the project has to be able to be implemented into our project and vice versa.

**3.1** **Goals and Guidelines**

The first goal that the project must achieve is the ability to implement the project in a way that coincides with the rest of the Alpha Tiles environment. Meaning that the look, feel, and sound of the game has to be in line with the other projects that are available within the Alpha Tiles environment. This includes keyboard size, app colors, images, and more. The coding must be done in the same manner as the other games on GitHub. Meaning that there must be a certain level of alignment between the style of code as well. The order in which the components are designed and implemented are not of as high importance as the functionality of the code itself. Which means that there needs to be more focus on the functionality of the code.

**3.2** **Development Methods & Contingencies**

This project will implement a structured approach to programming. The initial phase will be prototyping the model, followed by implementing the prototyped model into an actual Java and XML file format. All of the code will then be published to a private GitHub before being branched into the bigger Alpha Tiles repository. Since not all language groups have agreed to have their language implemented into the game, there will be no access to certain languages, and the developers will only develop in the English language. If need be, the program can be reverse engineered from already existing games like “Colombia from the Alpha Tiles database.

**3.3** **Architectural Strategies**

This program will be designed with input from the client Aaron Hemphill. Meaning that in a situation where their needs to be an adjustment of code based of the client’s input, the adjustments will be made immediately. With each design stage, the goals of the project will be adjusted according to the feedback of the client.

Factors that are subject to change include:

* Layout and design of UI
* Implementation of difficulty levels
* Game logic
* Sounds and visuals for achievement

**System Architecture and Architecture Design**

****

1. **System Architecture and Architecture Design**

The way the system will pass information will most likely be functions that are in the program itself. Since the project uses Java, the user is able to send information from one method to another using the parameters of the function.

For example, if there is a function called **UserInput()** that receives a string of chararters from the user, and then returns that string to a different function, the other function will then take the given input and compare it to the answer.

Since all of this will be happening on the local Android device that the user is playing the game on, there will be no need for exchange of information or involvement of other hardware devices while playing the game itself.

**4.1** **Logical View**

Graphical user interface, application, Teams

Description automatically generated

**4.2** **Hardware Architecture**

The hardware architecture of this project is distributed. Both the hardware that is used to develop the project, and the hardware used to test the project will be divided between the team members. There will be three development hardware devices being one HP Envy, one Mac Book, and one

**4.3** **Information Architecture**

As of now, there are no plans on storing any of the users personal information onto the system, but there might be a possibility that we integrate information of the users use of the app itself into the game.

1. **System Design**

****

**5.1** **Business Requirements**

This project must be aligned with the goals and values of Alpha Tiles, and SIL as a whole.

**5.2.1** **Database Design**

Since the databases for this project already exist within the Alpha Tiles project, we are simply going to use the already existing files to implement them within our own design. The current Database design is mostly centered around Excel spreadsheets, but that is not part of our process.

**5.2.2** **File and Database Structures**

Graphical user interface, application

Description automatically generated

**5.3** **Data Conversion**

Information regarding data conversions can be found in our Software Requirements Document.

**5.4** **User Machine-Readable Interface**

The user classes for this project will include only four groups since it is a much smaller scale project. The project developers, source code reviewers, project owners and the game players. Developers will have access to edit and add source code as they please to the project. Source code reviewers refers to the people within the project that critique and give advice on the source code, but do not have access to edit the code. Project owners are the people who the source code will be passed onto once the developers have completed their section of the project. They will be the ones developing, assessing, and updating the project later down the line. And finally, the game players are the ones who will be interacting with and reviewing the product. They will be the consumers and also give feedback on the UI but not on the source code, nor are they meant to have any form of access to the source code.

**System Design**

****

**5.5** **User Interface Design**

**A picture containing whiteboard

Description automatically generated**

1. **Operational Scenarios**

Since this game is an extension of the larger Alpha Tiles app, the first few steps of the operational scenarios have been developed. Once the user has selected our game, they will be led to the first level. In all of the levels, there is an onscreen keyboard that the user interacts with to enter their answer. There will also be a box with a concealed image, as the user starts guessing more and more of the correct letters, the image will start to reveal itself. At the top of the page will be a series of yellow dots that symbolize the level the user is on. In the top right corner, there will be a series of X’s that symbolize the number of lives the user has left. Depending on the user’s guesses the lives will either decrease or stay the same.

Some scenario examples include the following:

1. User inputs answer à Answer has no correct letters à One life is removed from top right à if number of lives = 0 exit.
2. User inputs answer à Answer has some correct letters à Relative number of tiles are removed from image box to reveal different angles of the image. à One life is removed from the top right à if number of lives = 0 exit.
3. User inputs answer-> Answer is equal to the word. à All tiles from image is removed. à If next level exists move to next level. à exit level.

**Detailed Design**

1. **Detailed Design**

The system can be designed using the code that exists on the Alpha Tiles repository. Since the game is meant to be integrated with the already existing Alpha tiles app, the code needs to be compatible and in the same template as the other games. The keyboard logic, the images, and the level structure are all already present in games such as “Colombia”, therefore do not need to be completely redesigned. However, the tile displaying image, the function that changes the life counter based on the image, and the correctness of each letter has not been implemented. Therefore, there needs to be testing of plenty of components into the game itself.

Another component that needs to be integrated is the testing of the games onto the Android devices. When testing the game on real devices, there needs to be different types of drivers present based on the device being tested on. If the application works only on the virtual simulation, and not on the hardware testing of the program, the project will be a failure. Therefore, it is necessary to test the hardware early.

**7.1** **Hardware Detailed Design**

Since this project is aimed to be used by multiple users around the world, the assumed hardware design is hard to quite grasp since it contains all Android compatible devices. However, for the purpose of development, there will be three test devices to run our program on. A Motorola MV smartphone with Motorola Drivers, an Ematic smart Tablet with Ematic drivers, and an unknown third device.

**7.2** **Software Detailed Design**

The main software that will be used for this project is Android Studio. Android Studio is an IDE that is specifically designed to design android apps. As of writing this project, the latest version of Android Studio is Dolphin 2021.3.1. It is likely that the team will update the application as time goes on. The other main software that the project itself has to run on is Android. Since the Android software levels vary we have set it to have a minimum API of 7, and be functional with API 19.

**7.3** **Performance Detailed Design**

Points of failure might include the keyboard not integrating right, the image of the word not displaying, the words on the screen not displaying right. Possibly the non-correct colors display, maybe issues with the IDE (android studios), or possibly problems with integrating the android app to an android device, and finally problems with testing the project on an android device.

**Appendix A: Acronyms and Glossary**

**Table 1 - Glossary**

|  |  |  |
| --- | --- | --- |
| Term | Acronym | Definition |
| User Interface | UI | the means by which the user and a computer system interact, in particular the use of input devices and software. |
| Application Programming Interface | API | a way for two or more computer programs to communicate with each other. |

**Appendix B: Referenced Documents**

|  |  |
| --- | --- |
| **Table 2 - Referenced Documents** |  |

|  |  |  |
| --- | --- | --- |
| Document Name | Document Location | Issuance Date |
| System Requirements Documentation | Canvas | September 30th, 2022 |
| Alpha Tiles Repository | https://github.com/AlphaTiles/AlphaTiles | 2020 |
| Ethiopia GitHub | https://github.com/brukmula/AlphaTiles | August 29, 2022 |

**Appendix C: Approvals**

**The undersigned acknowledge that they have reviewed the SDD and agree with the information presented within this document. Changes to this SDD will be coordinated with, and approved by, the undersigned, or their designated representatives.**

**Table 4 - Approvals**

****

**Document Approved By** **Date Approved**

****

**Name: Bruk Mulatu, Documentation Specialist-LeTourneau University**  **Date**

****

**Name: Brodie Bachmayer, Software Specialist – LeTourneau University** **Date**

****

**Name: Noah Garcia, Software Specialist -LeTourneau University** **Date**

****