

<Name-of-Software-Application>

**CS 230 Project Software Design Template**

Version 1.0

**Table of Contents**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | <05/20/23> | <Janncy Mota> | <OS chosen, "Draw it or Lose It"> |

**Instructions**

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

**Executive Summary**

We develop a Gaming Room that is made to run on any platform. It's going to be named "Draw It or Lose It" with it only being on android for the time being. People who play this game will have multiple teams ongoing in each of the four rounds. From there a person will guess from a library of images with one of the groups guessing until the time goes to zero. If it isn't answered both of the teams will get twenty seconds. If no one answers nobody gets the point.

**Requirements**

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

**Design Constraints**

* It will need more than one team.
* Teams consist of more than one person. [2 or more.]
* The game and team names will be unique to the user and will let them use a available name.
* Only one game can run.
* Is able to run on multiple platforms.

**System Architecture View**

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

The requirements are the ones listed in order to follow the code and the software that is planned. For the game aspect to be made, and to over look at the rest of the development of the application. When it comes to the Gaming Room it will need to be able to run on any device, such as android, windows, linux, and apple. In order for us to make it work we need to just come up with a way to rewrite it in their seperate languages or a way for it to inherit the languages in the different computer languages.

**Domain Model**

When it comes to Entity and according to our diagram it will create a relationship between the Game, Team, and Player classes. Making it inherit and get the information that Entity passes on. We see this on our UML design, making each class share a common inheritance like name and id. If we take a look at how they react to one another we see that Team and Player is a "has a" since it contains certain stored information. We can all this a aggregation in the UML format. Each of these contain informations, like the GameService with the reference of Games.



**Evaluation**

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements and look at the situation holistically, as it all has to work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| **Server Side** | The terminal will be flexible and allow the user to configure the server's web address, or even make changes to the game.  It will have various options to use different web addresses, allowing the user to switch it to a certain variety of choices.  The disadvantage is it could be difficult for the user to navigate through. | Similar things apply for Linux the characters secured are what are most perferred.  Security being important and the disadvantages being checked.  Another disadvantage is that it will be harder to navigate through to applications for web support. | There will be more software in comparison to the other OS.  It's going to be dominant and likely will have more of an audience that will be on this particular platform.  Advantages will be that it has a high source requirement, will be able to load far quicker than the others, and it will be easy to navigate through and feel comfortable for the user.  The disadvantage is that it's going to have trouble with viruses and easy to catch them. The tech support will be rather poor. | This will be better for the server allowing it to be tracked. But in comparison to the other devices the specifications are better.  The characteristics are that it's more popular because people tend to use their phones and take them everywhere, also will allow them to be mobile while they're on the game/server.  It's going to have the advantage of a wider audience reach, more compatibility, and the cost will be less an allow for a broader audience of people.  The disadvantage is that it's going to be selective to certain mobile phones. Security is also going to be an issue. |
| **Client Side** | For this to be done there's going to be some time and experience required. The cost will be similar to the Windows system. What is required is to ensure that it's all compatible with the rest of the web browser platforms and mobile devices. | A great deal of expertise is needed for Linux. The cost will be affordable and cosnidered minimum. | For windows it will be easy to navigate through, but for someone who is a newbie some time will be required in order to learn. The cost will be around the same as Linux. | This provides a great deal of flexibility, since it's mobile and a broader audience as well. It is somewhat more difficult to implement things into it compared to the other devices. |
| **Development Tools** | Running languages when it comes to mac we can mix with notepad++ and Mac can run more languages. The list consists of HTML, CSS, or Javascript and also supports fronted programming languages like Java, Python, PHP, and Ruby. | When it comes to Linux you're able to work with visual studio, eclipse, and use notepad++ since those are all good to navigate through. The languages that consist that are fronted programming languages are Python, PHP, Java, and Ruby. | This one in particular is easier to navigate through than Linux since it's windows. Visual studio, eclipse, and there are more languages that can be used on it. Notepad++ can be used as well and the languages that are there are Java, Python, PHP, and Ruby as well. | The amount of apps that can be created are limitless. The languages that can be used are HTML, CSS, Javascript, and it supports front end general languages like Java, Python, PHP, and Ruby. |

**Recommendations**

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

* **Operating Platform**: When it comes to where it should start, I feel as though Windows is the most safe option since it's easy to navigate through and you don't need much expertise when it comes to it. Along with the pay being minimum as well in order to get things working on the project.
* **Operating Systems Architectures**:
* The Windows system will have a great deal of applications that can help and enable the GUI. Along with more system recourses that can be helpful to the user. The applications of graphics, multimedia, and much more will allow the user to have more of a variety especially since they can use many user accounts or switch to different servers.
* **Storage Management**: <Identify an appropriate storage management system to be used with the recommended operating platform.>
* Windows in general has much storage management options that allow you to just fill up files on not only your hard drive but your email as well. The built in storage allows for easy methods to save the progress of your work for projects on any size so they don't get deleted carelessly and so you don't lose them since you'll be able to name the folders.
* **Memory Management**:
* When it comes to making the game you're going to need a database and a folder with the information. A library that you can access at any given moment. When you're working on your IDE and opening files it's much needed as well since it will help with creating the game.
* **Distributed Systems and Networks**:
* Since all of the operating systems are different I believe that the way to accomplish and connect all of them together would be by using this program that I was taught about. The program that I am considering is Pycharm because it's a language that everyone uses and it's rather known as well. It can be placed on iOS, Android, and a great deal of other options that will help with distributing the game and making sure that the language is understood in terms of programming to fully establish the game.
* **Security**:
* When it comes to windows there are built in security softwares that will help you incase you come across a virus or need protection. Malware, viruses, and security threats are bad things and no one wants to run into those because it could mess up your project and the work that you have saved, or even potentially ruin your computer.