

Draw It or Lose It

**CS 230 Project Software Design Template**

Version 1.0

**Table of Contents**

**Document Revision History**

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 11/09/21 | Concepcion-Ardon | Find out the operating system. Find out the budget and deadline. Go over the requirement details and suggestions. Read over the restrictions and policies. |

**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**

The client The Gaming Room wants to develop a web-based game that serves multiple platforms based on their current game, Draw It or Lose it, which is only currently available for Android. The main idea of this game is that multiple teams are going four rounds at a minute each. When a picture is pulled from a library of images one team has to guess till time runs out. If not answered each opposing team member gets to answer till the 15-second time limit. Figure out what type of operating system will be used, the budget, and the deadline. Get more detail on the requirements.

**Design Constraints**

* Needs one or more teams involved
* Each team has multiple players
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Must run on multiple platforms.
* Only one instance of the game can exist in memory at any given time.
* What is the budget for the software?
* What is the deadline for the software?

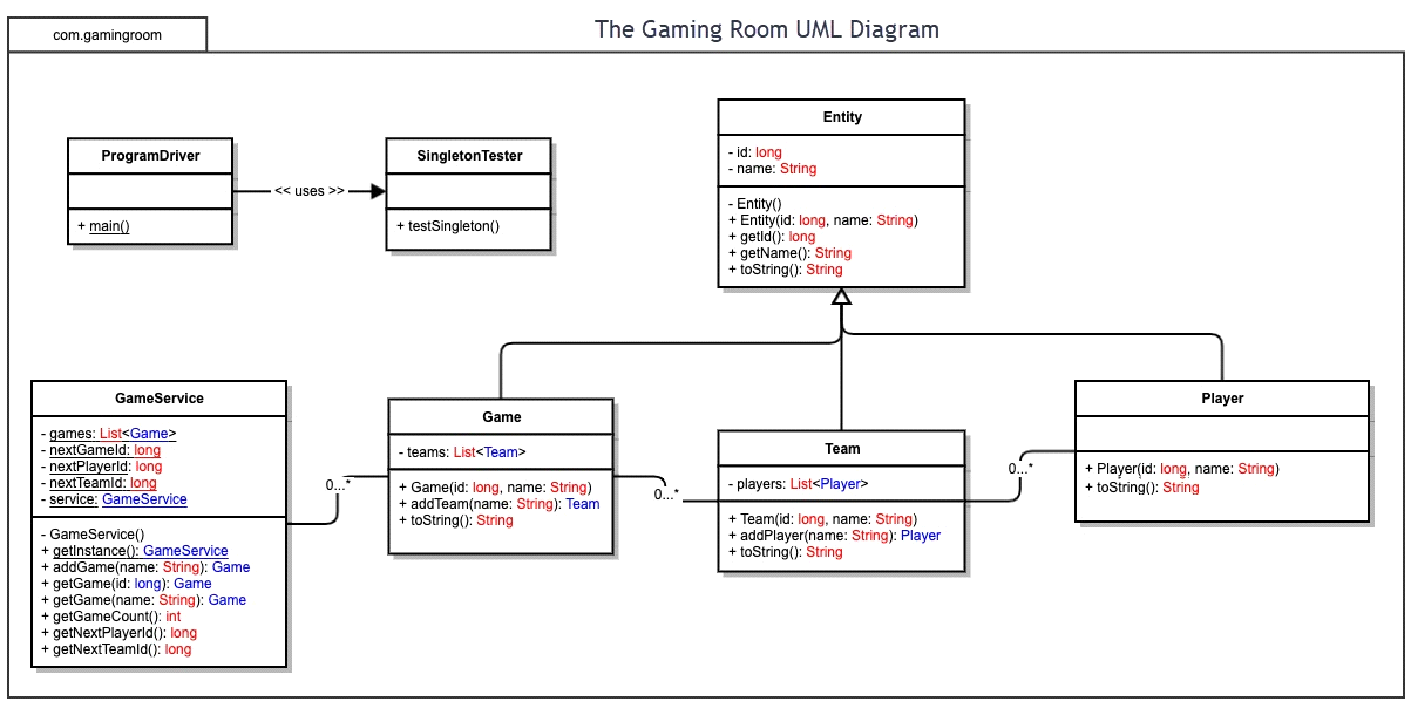
These are the requirements that are needed to be followed while developing the code and software. Although it is only the game aspect, we still require looking at application development. The Gaming Room would like this to run on all devices. This means we already have it on android but require working it into another mobile device. Along with other machines like Windows, Linux, and Apple. To meet these requirements we will need to find a way to either re-write the code in swift for IOS or come up with a way to use existing code to be run on other devices by inheriting other languages.

**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.

**Domain Model**

The class Entity creates a relationship between Game, Team, and Player class. This means they all inherit or get the information from the Entity class. The UML diagram shows this with inheritance. Furthermore, each class will share common references like “name” and “id”. Making the Entity class a superclass. Looking at their relationship, we can see Team and Player is a “has a” type. While Game has a Team and GameService has Games. When we use a UML diagram, we call it aggregation (HAS-A). When a user “has a” It means that it's an instance of one class and has a reference to an instance of another class. Looking at this diagram, we can see that GameService has a reference of Games, Games a reference of Team, and Team a reference of Player.



**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** | It has a flexible terminal command to configure the server, access, or make any changes.  Characteristics It is popular in web hosting  Advantages It is upgradeable, it has various options for different web hosting requirements  Disadvantages are that it is less preferred for web hosting services | The same goes for mac plus more cost-friendly  Characteristics Secured, most preferred.  Advantages Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  Disadvantages It is more difficult to find applications to support the web hosting required needs | More software is available compared to other OS.  Characteristics It is dominant to the other platforms. Close platform  Advantages High resource requirements, and less loading time.  Disadvantages easy virus susceptibility, poor tech support. | It's better if the server is immobile and can be tracked in a single place. Specifications are better for other devices.  Characteristics It is more popular, and has high portability.  Advantages Have a wider reach, better compatibility, cost-effectiveness  Disadvantages It is highly selective to various smart mobile devices Poor security |
| **Client-Side** | Moderate expertise and time required. Cost similar to windows. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Maximum expertise and time required. Minimum cost. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Minimum expertise and time required. Cost similar to mac. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | When running languages on macs we can run swift. While mixing in nice tools like notepad++. Though Macs can run all languages. Languages consist of but are not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. | Linux can work with Visual Studio, eclipse, along with notepad++ for a pleasant and easy-to-use tool. Along with many more languages and tools. Languages consist of but are not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. | It is easier to use than Linux but can run the same as it. Furthermore, Visual Studio, eclipse to name a few of the many languages. And with multiple tool's notepad++ is a simple-to-use tool. Languages consist of but are not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. | It can create many apps using android and swift. Both of the languages and software packages can be run on all three machines. Languages consist of but are not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be 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**: I would recommend The Gaming Room starts with Windows devices because it has more software available along with minimum expertise and cost to get projects going. Plus It also won’t run into a shortage of IDE’s to work with.
* **Operating Systems Architectures**: Windows provides services used by all Windows-based applications that enable applications to show a Graphical User Interface (GUI) while accessing system resources and more. These applications are also referred to as Graphics and Multimedia, messaging, and web services. These services can be used by a user account or a server specifically.
* **Storage Management**: Windows 10 comes with a good feature called storage sense. This allows the company to scrutinize and manage files on their hard drive, along with how much space it can take up. Other features include being able to choose the save locations for apps making them easier to access. Furthermore, the company can also use the cloud to save data. The built-in storage system allows for easy file creation and placement for large projects, and they won’t get lost or carelessly deleted.
* **Memory Management**: While creating the game we will need to create a database or library with lots of pictures. The memory allocation allows for easy storage of pictures outside the default picture folder. It allows us to keep the companies whole project together in a more secure area on the computer. This includes when working with the IDE and opening files from it to create the game.
* **Distributed Systems and Networks**: Each operating system is different, and we investigated ways to publish the game to run on all dives. I found out to Develop 4 which enables cross-platform game creation. It’s an IDE that can be run on any device. Once the game is created you can simply export the game file into the web, iOS, Android, and many more options that will allow to cross-play. It will help with the dependencies. To prevent other problems like an outage, connectivity, and the company will need to make sure their servers are strong enough to support large player volumes along with backup power for power outages.
* **Security**: Windows comes with built-in security protection software. Although to secure user data and information it is recommended to use another source. Although if we are discussing what is on the machine windows comes pre-equipped with protection. The system can scan for malware (malicious software), viruses, and security threats. It all happens in real-time, and because of the threats change the system updates automatically to keep the system and user information safe and protected.