

<Draw It or Lose It>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_heading=h.30j0zll)1

[**Table of Contents**](#_heading=h.1fob9te)2

[**Document Revision History**](#_heading=h.2et92p0)2

[**Executive Summary**](#_heading=h.tyjcwt)3

[**Design Constraints**](#_heading=h.3dy6vkm)3

[**System Architecture View**](#_heading=h.1t3h5sf)3

[**Domain Model**](#_heading=h.4d34og8)3

[**Evaluation**](#_heading=h.2s8eyo1)3

[**Recommendations**](#_heading=h.3rdcrjn)5

## [Document Revision History](#_heading=h.2et92p0)

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 03/18/21 | Patrick Valencia | For this revision, I have updated the Executive Summary, Design Constraints, and a descriptive review of the Domain Model. In addition, working game, team, and player classes with the new addition of the Entity class. |

**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](#_heading=h.tyjcwt)

The Game Room wants to create a web-based version of their android game app Draw It or Lose It, but they do not know how to set up the environment. That is where Creative Technology Solutions comes in. We will build their requested software following their requirements of:

* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

By following their request thoroughly we would be able to fully construct their game to a web based application for a wider audience to enjoy.

## [Design Constraints](#_heading=h.3dy6vkm)

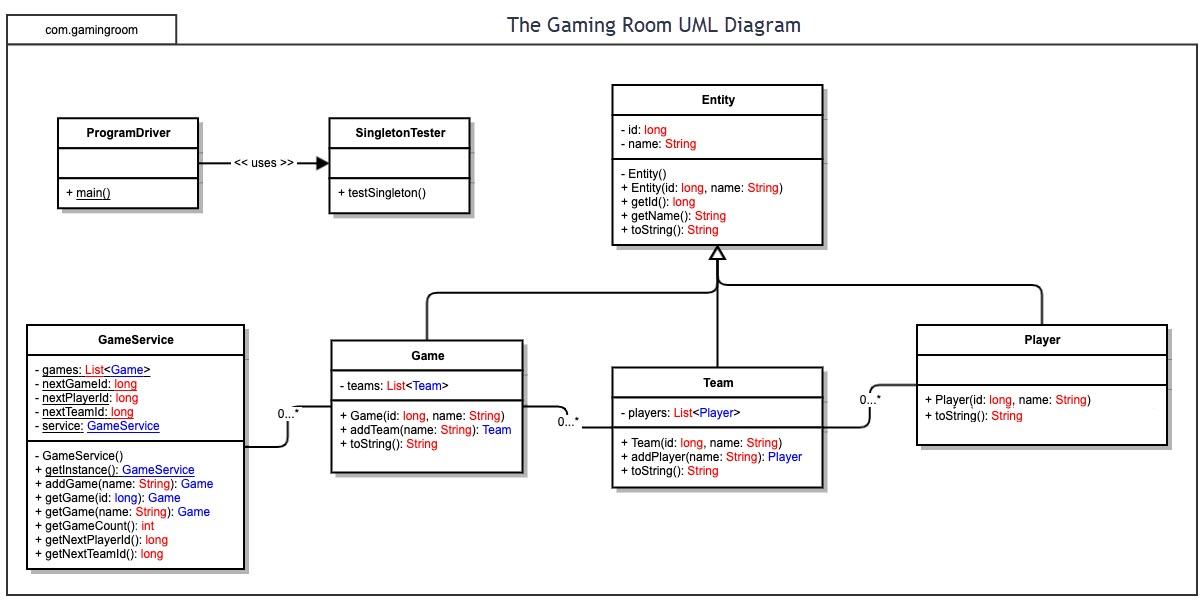
There are multiple design constraints that we will encounter with building this application. The first one is that the game must run on multiple platforms. Due to it being a web based game application, the game must run while taking into account peoples usage of different web browsers and operating platforms. Another constraint is the check for a unique team/game name. This one isn't as hard as the previous one but it is a constraint we must take into account with the development process. The final constraint is checking whether there are multiple game instances running and limiting it to one game at a time. This one is also a simple check but just like the unique names, it is something we must take into account.

## [System Architecture View](#_heading=h.1t3h5sf)

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](#_heading=h.4d34og8)

The UML class diagram below is the completed class diagram with the ProgramDriver, SingletonTester, Entity, GameService, Game, Team, and Player classes. ProgramDriver relates only to SingletonTester through association. GameService relates to Game with its multiplicity in which it goes from zero or more instances. Game and Player class both relate to the Team class by the same multiplicity of “0…\*” which means from zero or more instances. As for the final relation, Game, Team, and Player class relate to the Entity class through an inheritance relation. The inheritance relation is the object-orientated programming principle and is useful in order to fulfill the software requirements as the entity class would share this relation with the Game, Team, and Player class which are needed components to create the web based game.

****

## [Evaluation](#_heading=h.2s8eyo1)

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 Mac Operating system is extremely flexible and easy to use terminal commands to configure the server, access the server, or make changes, the one downside would be the cost of software in comparison to other OS are higher | The Linux operating system has an advantage of being more cost friendly due to all the open source applications out there for server hosting software but at the cost of digging through community forums to fix problems | The Windows operating system has the most expansive and available server hosting software but that also comes at the cost of experiencing horrendous technical difficulties and at times awful windows updates | There is really no viable reason as to why a server host should be on a mobile device. The server should be stationary in order to provide the best connection possible |
| **Client Side** | For the Mac Operating Systems, a moderate amount of expertise and time is required in order to keep and maintain proper use with clients. The cost factor is normally a bit higher when compared to windows | For the Linux Operating System there would have to involve a maximum level of expertise and time. It's so high due to the use of linux being very tricky to grasp and most information being located in community forums. As for cost it is low because there are a lot of open source server hosting applications | For the Windows Operating System, there is barely any expertise required as for time it can range from low to medium amount required depending on what's needed. As for cost, I would also go with low to medium cost as there are several programs or options to easily run the client side | For any mobile devices, there is incredible amounts of flexibility for client side development. There is a moderate amount of time and expertise required as for costs it would also be a moderate amount. |
| **Development Tools** | Common languages for the Mac Operating System are HTML/CSS/JavaScript and supporting libraries to support the frontend and general. Tools involve PyCharm, Eclipse, Visual studio, Github, Notepad++, etc. | Common languages for the Linux Operating System are HTML/CSS/JavaScript and supporting libraries to support the frontend and general. Tools involve PyCharm, Eclipse, Visual studio, Github, Notepad++, etc. | Common languages for the Windows Operating System are HTML/CSS/JavaScript and supporting libraries to support the frontend and general. Tools involve PyCharm, Eclipse, Visual studio, Github, Notepad++, etc. | Common languages for any Mobile device are HTML/CSS/JavaScript and supporting libraries to support the frontend and general. Tools involve PyCharm, Eclipse, Visual studio, Github, Notepad++, etc. |

## Recommendations

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

1. **Operating Platform**: There are a multitude of operating platforms that would work incredibly well in the expansion of Draw it or Lose it, but in my opinion the best development platform would be for windows. The windows operating system is incredibly widespread and incredibly easy to develop on due to such widespread tools and support for said tools.
2. **Operating Systems Architectures**: As previously stated, the Windows operating system is incredibly expansive as there are over 1 billion devices globally that run Windows. With such an expansive amount of devices, it seems like the go to in development. Additionally, there is an equally impressive and expansive support for server side hosting applications. With said applications there are also numerous community forms or application support in which developers can use to help streamline building.
3. **Storage Management**: An appropriate storage management system that should be used to host these files would be DriveHQ. I can make this recommendation based on the fact that these files should be hosted on a cloud storage system as they would have the most reliability in not going down for maintenance or power outages. Additionally, it provides competitive prices and services. Its interface is incredibly intuitive and easy for users to utilize.
4. **Memory Management**: Some memory management techniques that could be utilized would be to optimize the images shown to users in changing its RGB properties/resizing to use less memory, or try to make sure that there are no memory leaks within the application. If these are taken in mind when developing the application, memory management would be a breeze and not use such a drastic amount of resources from the user.
5. **Distributed Systems and Networks**: When it comes to distributed systems and networks, connectivity and outages shouldn’t vary too much due to how the storage management would be used. Since it will be cloud based storage there would be minimal outages as there are multiple servers in which data would be stored and users would be able to connect to them.
6. **Security**: We want the users to have the confidence that we will protect their data. For this instance since user data will be stored on our servers, we can use a 2 factor authentication method in order to make sure that said users are getting to their accounts properly. We won't just stop at 2 factor authentication. In addition, the data on said servers would also be encrypted just in case of a data breach and only certain accounts would have certain privileges in order to minimize damage if an account falls into the wrong hands.