

Win, Lose or Draw

# **CS 230 Project Software Design Template**

Version 1.2

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.2 | 08/15/2020 | Nicholas Wood |  |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room is a gaming company that wants to put their Win, Lose or Draw mobile game onto a web-based version, and they do not know in what environment it will be set up. The hardware requirements for the project are not set and will be based on the software application decision. I would recommend using Linux as it is the most flexible and it is much easier to maintain as well as secure.

## [Design Constraints](#_2et92p0)

The game application must have the ability to have one or more teams.

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.

## [System Architecture View](#_ilbxbyevv6b6)

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](#_8h2ehzxfam4o)

There are many classes associated with this project, the first instance of this work had classes Game, Player, and Team having similar structures with each having their own declaration of ID and Name. They also had getters for each of their variables as well. The change that we implemented made the Game, Team, and Player class all extends or inherit from the Entity class. The entity class now declares the ID and Name and is responsible for the getting of these variables. This makes it easier to create an object of player, team, or game. This makes the code much for flexible and it reduces repetition of variables, if I were to add a new variable to each object then it is very simple to do that, rather than having to change each class I would just need to change one class. The Game, Team, and Player classes now have constructors for majority of their class and the Team class has a add player array that will add players if they are not already added. The Game class has a similar feature but that is with Teams being added to the array list. The GameService class is responsible for maintaining the games that are created as well as adding games based and retrieving the information from the games. The GameService also assigns the game ID and it has the player and team ID’s as well. The Program driver has the only main method, so it is the driver of the program, it is adding games to the system and the Singleton Tester is testing the program to make sure that only one instance is created.

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac has less susceptibility to security issues. Easy creation of features. Integrating servers into a larger environment is difficult. | Linux is a free and open source, able to see source code. Linux is very flexible when it comes to server needs. Has the most secure kernel. | Fewer versions to allows for easier licensing. Reducing the data duplication in the system. Able to support large amounts of data. Easy to use server management. | Mobile devices are more susceptible to security issues. Reduction in the IT needs, reduction and equipment cost. |
| **Client Side** | A license fee to use Mac is present. The software management is more expensive then most. The Mac does have a support system as well. | No license fee to use Linux but the support cost is higher than most. The expertise for Linux administrators is much higher than others. | A license fee to use Windows. There is software assurance from windows that support and problem resolution. Administration cost are lower for windows. | Website apps are more expensive to maintain and create compared to a website, they also have the added issue of having to adjust to each platform. |
| **Development Tools** | Mac is written in C because the kernel is written mostly in C. | Linux is written in C but it is flexible, there has been cases where C++, Python, and Java are used. | Windows is written in C and some parts have assembly language. | iOS, Android, and Windows phones are all written in objective-C. They are adaptions of the other existing kernels. |

## Recommendations

1. **Operating Platform**:

Based on my analysis I would recommend using the Linux as the operating platform.

1. **Operating Systems Architectures**:

The Linux operating platform has primary components: the kernel, hardware layer, system library, shell and system utility. The kernel is responsible for all the major activities as it interacts with the underlying hardware. The hardware consists of the RAM, HDD, and CPU. The system library is used to implement the functionality of the operating system. The shell is the interface between the user and the kernel. It takes commands from the user and executes the kernels functions. The system utility are programs that are liable to do individual task.

1. **Storage Management**:

The storage management of a system deals with the size of the system and the number of files it is capable of handling. For a game that deals with images and guessing the images, if there were a limited number of images then the user would get tired of it and lose interest in the game. Games are supposed to be fun and challenging to the user otherwise if it is too easy then they will quickly get tired of the game. The type of storage management for the game application would be magnetic disk to allow easy access as well as plenty of storage for the system. For the game application we are using the indexed allocation method that points to multiple locations on a disk which would require a large movement and may slow down the system. This means our disk scheduling would be circular scan which moves the disk while providing the desired request along the way. This would be the most efficient for the multiple platforms we are using.

1. **Memory Management**:

The memory management of a system deals with the transferring of files and the performances from this transfer. If the transfer was very slow then the user would not appreciate this and would not continue to use the application. If the file returned is corrupted or not loaded corrected or fully that would be unappealing to the user. For the game application disk will be used due to the ability to access any information that is desired at any moment. The allocation of memory for the disk would use the indexed allocation. The indexed allocation coupled with a free space list which links together all the free space in a disk reduces the number of pointers and it makes the system much more efficient.

1. **Distributed Systems and Networks**:

A distributed system allows resource sharing including software. Typically connected to the network at the same time. The price per performance ratio is much better in these systems. The systems are transparency with the ability of the single system without concealing details to the user. Distributed systems are scalable, and they are open making them easier to configure. One issues with this is security because the system is open to the public.

1. **Security**:

Linux’s kernel is considered the most secure kernel available. Keeping the kernel and software up to date as well as using the Linux security extensions will protect against misconfigured or compromised programs. For the users I would include a password for them to access the system, but I would also require a two-factor authentication to make sure the user is not compromised. Having a firewall limits access between two security domains and monitors and logs all connections. Since the application is through the web the HTTP is used and a firewall will allow this type of connection to pass through the system.