

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

## 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 |
| --- | --- | --- | --- |
| 1.0 | <05/18/2020> | <Taro Serigano> | <Initial Version> |

**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](#_sbfa50wo7nsh)

The game, “Draw it or Lose it” will most likely be released in 5 major platforms, Windows, Mac, Linux, iOS and Android. The problem here is that how we will be able to maintain consistency across different platforms and minimize differences due to the different environment in each platform.

## [Design Constraints](#_2et92p0)

My main concern is that because this application is heavily time sensitive and involves multiple people of multiple teams, any second of lagging will cause a huge issue. We’ll have to ensure that everything functions in a strictly timely manner.

## [Domain Model](#_8h2ehzxfam4o)

attributes and properties that are overridden by the three classes.

Also, we can see one multiple “one-to-many” relationships in from GameService to Game, from Game to Team, and from Team to Player classes. For example, one GameService has List<Game> object that can be produced multiple sets of games and so on and so forth.

We can also see that the ProgramDriver class uses Singleton class of SingletonTester. This is to make sure that only one instance of the GameService class can exist in memory at any given time.

Team and Player has a relationship of "has a", like, Team has Players. Also, Game has Teams, and GameService has Games. In Has-A relationship, one instance of one class has a reference to one instance of another class.

Game, Team, and Player have "is a" relationship with Entity and they all inherit from Entity class. This means that they three all share the same attributes such as id and name. Entity class is considered Super class.

****

## 

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Terminal commands are more flexible. Also is it highly configurable with the server, access, and enabling changes. | Also, terminal commands are flexible and configuration options are highly available like Mac, but far more low cost. | A variety of software is very rich. | It is able to be tracked at a single place. Generally, specs are better in other devices. |
| **Client Side** | Some good knowledge and expertise are necessary. Costs are similar to Windows. | Advanced knowledge, experience and expertise are necessary to operate. | Expertise and experience are not highly required. Costs are similar to Mac. | Flexibility is given to both clients and developers to see updates all time. It is somewhat more challenging to implement than other platforms. |
| **Development Tools** | Most development tools nowadays have adapted compatibility across Max,Linux and Windows nowadays. It just varies in how you install these tools depends on the OS. Common languages are HTML/CSS/JavaScript for frontend development, but some languages like javascript can also used for backend by node.js. Other common backend languages like Ruby, PHP, Java, Python etc. Tools: Visual Studio, Eclipse, Github etc. | | | Again, both frontend and backend languages afore-mentioned are employed for mobile (even Javascript nowadays). But the development tools such as Android Studio and SDK tools, XCode, AppCode are mainly used. |

## 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**:

Although there are many attrctive Linux based options, Microsoft does offer one of the best server operating systems for business as well. Microsoft Windows Server is developed from Windows 10 OS and offers comprehensive package that one needs to build a server. Just like other Microsoft applications, it is easy to install and available through the Microsoft official store. Microsoft Windows Server also supports cloud-ready environment, which makes it user-friendlier than most Linux based systems. The drawback is that although it resembles a lot of Windows features, Windows Server does have some features that are different from the Windows.

1. **Operating Systems Architectures**:

I’d highly recommend Microkernel architecture. A microkernel architecture includes only a very small number of services within the kernel in an attempt to keep it small and scalable. The services typically include low-level memory management, inter-process communication and basic process synchronisation to enable processes to cooperate. In microkernel designs, most operating system components, such as process management and device management, execute outside the kernel with a lower level of system access.

1. **Storage Management**:

I’d recommend using the S3 buckets offered by AWS. S3 is highly durable, scalable cloud storage that is easy to manage, operate and very cheap to use. Leveraging the storage system is highly costly and time consuming, but with Amazon S3, you can save significant amount of time and cost. S3 also offers options for replications over different Availability Zones, therefore it is very fault-tolerant, you wouldn’t need to worry about losing your precious data!

1. **Memory Management**:

I’d recommend using Memory mirroring**:** It is one of the best memory management techniques in which the physical memory is separated into two logical channels, and the first channel mirrors the second one. So, if the main memory is temporarily not in use, its mirror would be set to function till it gets back to normal working. Everything is done by the memory controller without downtime and without client’s notice.

1. **Distributed Systems and Networks**:

A network operating system provides an environment in which users, who

are aware of the mutiplicity of machines, can access remote resources by either logging in to the appropriate remote machine or transfeering data from

the remote machine to their own machines. Currently, all general-purpose

operating systems and even embedded operating systems such as

Android and iOS are network operating systems.

1. **Security**:

This would be a very broad topic to cover and it would take a large number of pages to fully cover. But in summary, security can protection countermeasures can be broken down into several basics. First, the real threats can actually exist inside. The data shows that a lot of hacking have been performed from the inside; employees. It is highly recommended to always provide the minimum privileges to employees based on their roles. This can reduce the possible risks of employees abusing their chances to hack into the organization’s system. You can also utilize cryptography. Today, you can almost never store users’ password as is and you should always encrypt them. We also have to be aware that the recent technology has enabled us to be introduced to some devices that can read million patterns of encryption per second. Therefore, Cryptography will not solve all the security issues and you cannot rely solely on it. You always have to add different layers of different security approach. User authentication is almost a must have today to enhance users’ security. The research shows that as you would add more number of characters for password, it becomes exponentially difficult for automatic devices to hack the users’ password, for example. I’d recommend to implement some thorough user authentication that are considerate and yet easy to follow to ease out users’ experience.