

Draw It or Lose It

# **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 | 03/20/2021 | Emma McCarter | Prepare a software design document and begin developing a game application, addressing the software requirements, architecture, and recommendation. |

## [Executive Summary](#_sbfa50wo7nsh)

Draw It or Lose It is currently an Android only app in which teams compete to guess what is being drawn, this game is team based and is structured with a time limit. The game needs to be developed into a web-based game that will be able to be used by multiple platforms.

## [Design Constraints](#_2et92p0)

* Only one game can take place at a time due to the memory
* The team names must be distinctive
* The need to assign players to teams
* The software has to support a multiple team platform

## [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)

The Entity has long and string variables, as well as constructors. Game, Team, and Player methods all inherit from the Entity class. The GameService class is the only one that does not inherit from the Entity class. This will allow them to have an ID and Name. The Game function allows for players to be added and teams to be formed. The Team, Game, and Player classes all generate from each other. For example, a game cannot be created if two teams are not available to play.

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | This can be difficult, but you would need to find someone skilled, and this is where the cost of work will increase. Mac OS does work fast and efficiently so the data can be accessed and used at a good speed. | Linux is the most common server used and is by far the cheapest option. Because it is the most common, many developers know how to use it. | Windows has the ability to run multiple servers. The biggest disadvantage is the amount of time, which cost money. The advantage is that many people are familiar with Windows. | The processing power of a mobile phone is little to none. This is the least likely option for a developer. |
| **Client Side** | Mac would be the least used platform for clients. The cost to run on the OS is very high. Applications do run well and have the ability to be transformed in intricate ways but that all costs more money | Linux is the most used OS, but it is also the hardest for users. It is the most used because it can be modified exactly to the user’s needs, which is the cheapest of all servers. | Windows is the most user friendly compare to Mac and Linux. Windows is more preferred for gaming as well. | Mobile phones allow for an easy way to download application. Although, it is hard to collaborate on a phone than on a desktop. In order to create an OS for a mobile phone, I must first be created on Mac or Windows. |
| **Development Tools** | HTML, Java, Python, Eclipse, GitHub, Notepad | HTML, Java, Python, Eclipse, GitHub, Notepad | HTML, Java, Python, Eclipse, GitHub, Notepad | HTML, Java, Python, Eclipse, GitHub, Notepad |

## 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**: Operating platforms are very different between a PC game and cell phone game. The game design varies and so does the abilities of the game. Platforms compete to win and want to have the best structure for their users. Platforms ultimately determine the quality of the game so that developers can use the platform to the best of their abilities to produce an eye catching game. The best platform at this time is Windows.
2. **Operating Systems Architectures**: Windows has a long architecture. It contains the following: a hybrid kernel, HAL, drivers, and range of services. The kernel mode allows for code to run, can reference any memory address, and is the most trusted function of the operating system. You will first have to determine if your computer is a 32-bit version or 64-bit version to determine you kernel.
3. **Storage Management**: The best option is going to be a solid-state drive (SSD). It will speed up everything that requires disk access, and it is usually a smaller drive. The SSD is most desirable because it can hold all of your Window files, games, and programs. The main reason for a SSD is the high speed of accessing files, which is a main priority for most users.
4. **Memory Management**: Windows recommends that on a 32-bit Windows, there is up to 4 gigabytes of memory, whereas, on a 64-bit Windows, there should be at least 8 gigabytes. You must remember that threads cannot access memory that belongs to another process, which protects it from being corrupted. That is why all threads can access its virtual address space.
5. **Distributed Systems and Networks**: Many networks, for example a gaming network, includes a database that can be shared among multiple users that are all distributed and interact with one another over the network. This is especially important to network game developers as they try to create inter player communications and need a starting point. That should always be the main focus when creating any game, players want to interact and share among themselves. This can also be dangerous and there should be backup to each of them as many players take advantage of being able to interact with other players, can use foul language, be disrespectful, and may need to be banned for a short period of time, or indefinitely.
6. **Security**: Going off of my previous point, security is a major factor to any IT. There will always be threats, those who want to steal data, money, or infrastructure. Data protection is a top priority for game software developers due to so many people wanting to be on top, there will always be cheaters trying to get your data.