

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 | <09/20/2020> | Jovan Martinez-Saldana | <Brief description of changes in this revision> |

**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 Gaming Room would like us to design their game Draw It or Lose It. This application should be able to support one or more teams and be able to populate said teams with multiple individuals as the players for those teams. The titles for team names as well as the Game should be unique in order for simplicity and so that players can determine if a name has been chosen. Only one instance of the game can exist in memory at any given time. We can design the multiple teams an individual by simply populating users on an online server when their individual game starts. Unique names can be instituted by running a username and Team name prompt that when given an input, will decide if one, the name(s) is taken, and two, if the name is appropriate. Since only one instance of the given game can exist at any one given time, this is where our live server(s) will also kick in. Once that handful set of individuals are in game, that game will exist on our server where the game can be separated from any other users trying to find a game.

## [Design Constraints](#_2et92p0)

Some of the more prominent design constraints will be that this is originally developed on Android. We will need more information if we want to branch out to iOS devices. Deciding what kind of web-based language to code in is also important to note as well such ass HTML and JavaScript. When this was being developed on Android, was there already a skeleton design for the application? Time and cost will also be a factor, such as how much man power we really need for a project of this size. Design may be restricted simply by what competition is already out there on the market. The product has to stand out enough that other companies do not claim plagiarism.

## [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)

For starters, the Classes Game, Team, and Player have an “is a” relationship with Class Entity. That means that Game, Team, and Player are all “Entity” as well and inherit from Entity. Since in the relationship all three classes that share inheritance share id and names, Entity can be consider their Super Class

The classes of Team and Player and GamesService can represent an aggregational relationship because of the “has a” relationship depicted between Team and Player. The rest of the classes tend to have just references to each other. Like GamesService references Games and Games has a reference with Team.

****

## [Evaluation](#_2o15spng8stw)

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** | Macs have generally good graphics and are easy to use (Designed for the consumer). Simple accessibility and server configuration options | Compared to MacOS, theres a lot of difficulty with navigating the system. More cost friendly overall. Command shell | User Friendly interface. Development friendly unlike Mac. Command Line options | Cost varies. Difficulty in development when trying to create something that is universal to all mobile device types |
| **Client Side** | Expensive. User must really know beforehand how to navigate the system. | Low cost. Most likely requires a high level of time and expertise to support the system. | Very consumer friendly and oriented so user does not need to understand too much from the get-go. More expensive than Linux, but usually less expensive than Mac. | From client side, user will need to push a lot of time into learning and honing mobile development skills. Mobile can be difficult to translate systems that were meant for a different system |
| **Development Tools** | JavaScript, HTML, CSS, Python. Eclipse, Visual Studio | Gedit, Vim, Netbeans, Bluefish, Geany, Aptana, Quanta Plus | JavaScript, Java, C#, PHP, Ruby etc.  Visual Studio, Eclipse | Java, Python, Swift, Ruby etc. App Inventor (For Android), Ionic, Sencha, Mag+ 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**: I think the most beneficial Operating platform would have to be Windows, as from there its easier to work off of to expand to other operating systems if need be. The tools already found on Windows often at time already translate well with other systems.
2. **Operating Systems Architectures**: From Microsoft themselves: “**Windows** is a series of operating systems developed by Microsoft. Each version of **Windows** includes a graphical user interface, with a desktop that allows users to view files and folders in **windows**. For the past two decades, **Windows** has been the most widely used **operating system** for personal computers PCs”
3. **Storage Management**: The Windows Storage Management provider
4. **Memory Management**: the memory techniques provided by The Windows Storage Management provider can be used to manage storage configurations such as the unique names of teams and users in Draw It or Lose It.
5. **Distributed Systems and Networks**: <Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).>

This can be accomplished by creating or paying a third party to create a program that can translate OS A software into OS B software. This has been done many times between Windows and Mac

1. **Security**: <Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.>

The easiest way to protect user data is to encrypt and store user information off site from any of our junctures as well as from The Gaming Room. Storing our servers offsite can give us more security in terms of us detecting security breeches faster than they can be cracked as well as keeping any potential threats “honest”. Two Factor Authentication will also be instilled every time a user tries to access the product.