

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 | 06/15/2021 | Miranda Putnam | Project Three software design document. |

**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 wants to expand their Android application game, Draw It or Lose It, to a web-based game. The web-based game expanded from their Android app will be functional on different platforms. The web-based game will also have one or more teams, multiple players on each team, with unique game and team names and only one instance of a game existing in memory at a time. The software created will do this by using a system of different classes and OOP principles to address the clients software functionality requirements for games, teams, and players.

## [Design Constraints](#_2et92p0)

1. Staying consistent with the look and branding of the original Android Draw It or Lose It application is important for the users as well as the client. The existing look of the game needs to be implemented in the web-based game.
2. The Gaming Room is asking for a web-based application game, so hardware with an operating system to run the web server will be necessary.
3. The game will utilize images which will require storage space to be taken into consideration.
4. Security is necessary for the web-based application because the game requires unique identities for players.

## [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 UML Gaming Room class diagram below uses several OOP principles to demonstrate the software. First, encapsulation is used by making the instance variables private but the accessor methods public. This creates a class and keeps it separate from other classes in order to restrict access and give each class its own unique attributes. In the UML diagram for The Gaming Room the classes are Entity, GameService, Game, Team, and Player. There is also ProgramDriver to hold the main method and SingletonTester to hold the singleton method, so only one instance of the objects exists at a time. Another OOP principle used in this UML diagram is inheritance. The Game, Team, and Player classes all inherit from the Entity base class, meaning they all inherit the attributes listed in the Entity base class in addition to their own attributes. The last OOP principle used is polymorphism. The Entity base class behaves differently depending on which class is called on, either Game, Team, or Player. In other words, the Entity base class has many forms. The GameService, Game, Team, and Player classes are also all associated from left to right, with a multiplicity of zero to many. So, GameService can have a game, Game can have a Team, and Team can have a Player, but the reverse is not true.

****

## [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** | **Pro**: High security with built-in antivirus software, regular software updates to patch security gaps, and password and multiple user settings.  **Con**: Moderate to very expensive systems.  **Pro**: The quality is consistent and trusted because only Apple makes Macs.  **Pro**: Can be equipped with most software’s necessary to run the web server. MacOS server available.  **Pro**: Good customer support and simple to repair.  **Con**: Difficult to upgrade and customize hardware if necessary. | **Pro**: Open-source, customizable, and able to be enhanced if necessary.  **Pro**: Secure system as applications require authorization from the admin user.  **Pro**: Free to use with no licensing fee.  **Pro**: Speed. Able to handle a large amount of users and perform tasks with ease due to low memory usage.  **Con**: Many software’s and applications are not available for Linux.  **Pro**: Can be used as a secure and flexible server.  **Con**: No user guide for troubleshooting issues. | **Pro**: Many software’s and applications available including server functionality.  **Con**: Weak security. Role-based access is difficult to implement and the system is not built to be secure so outside security software is required.  **Con**: High licensing cost.  **Pro**: Various devices available ranging from low to moderately priced. | **Con**: Not a lot of storage space which could result in decreased speed.  **Pro**: Minimal physical space taken up within the office space.  **Con**: This server would not be stationary which could result in tracking issues. Could be easily lost or misplaced.  **Pro**: Low security threat as malware attacks are uncommon and most devices lock and require a passcode.  **Con**: Software and applications are limited based on compatibility with the mobile device.  **Con**: Small interface and device could be difficult to interact with.  **Pro**: Wide range of devices available with a wide price range. |
| **Client Side** | **Pro**: Compatible with many different browsers.  **Con**: High cost with a small range of devices.  **Pro**: Simple interface with minimal expertise required.  **Pro**: Default browser Safari has high speed, is free, secure, and requires minimal expertise. | **Con**: Interface can be difficult to learn, a high amount of expertise is required.  **Pro**: Uses Firefox as a the default browser, which is both fast and secure.  **Pro**: Low cost and simple to install on many different devices. | **Pro**: A minimum amount of expertise is required to interact with the system.  **Pro**: Wide variety of devices with low to moderate pricing.  **Con**: Quality of device can range greatly because anyone can make a Windows PC.  **Pro**: The default browser, Microsoft Edge, is fast and has available security features. | **Pro**: A portable device, can be used on the go and take up minimal space.  **Con**: Can be expensive for a quality mobile device that offers good speed and memory. |
| **Development Tools** | **Pro**: Many IDE’s for software development, such as Eclipse, are available for download on Mac.  **Pro**: Mac includes Xcode software for free, which supports the Java and HTML languages.  **Con**: Xcode has no inherent JavaScript support, so additional tools and extensions may be necessary. | **Con**: Linux does not include a default IDE.  **Pro**: There are IDE’s available for download on Linux that are free, such as Eclipse for Java, HTML, and JavaScript. | **Pro**: Windows has a basic free IDE called Microsoft Visual Studio, with extensions available for Java use, HTML, and JavaScript.  **Con**: Any editions beyond the basic Visual Studio have a fee. | **Pro**: There are IDE’s available for mobile coding in Java, JavaScript, and HTML, such as AIDE.  **Con**: Limited capabilities with potential licensing fees depending on the mobile device.  **Con**: Difficult for use on a development team with multiple people at a time, as the device is small. |

## 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**: An appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments is Mac.
2. **Operating Systems Architectures**: Mac operating platform is a good choice for The Gaming Room because it is highly secure, has consistent quality, is simple to repair if needed, and has the necessary tools either built in or available for download in order to expand Draw It or Lose It to other computing environments. These tools include MacOS Server and a variety of IDE’s to choose from to support the use of Java, JavaScript, and HTML.
3. **Storage Management**: An appropriate storage management system to be used with Mac operating platform is a local disk. The images required for Draw It or Lose It games would be stored on a disk and loaded into memory when a game begins. This would only require 1.6GB of storage.
4. **Memory Management**: Mac uses various memory management techniques. For the Draw It or Lose It software, Mac will track RAM usage and activities to allow the client to monitor their memory usage and free up space easily if needed. Draw It or Lose It will use memory for the image files on a local disk only when a game is started up, which Mac will easily accommodate.
5. **Distributed Systems and Networks**: The Gaming Room will use a distributed system and network to communicate Draw It or Lose It between various platforms. A distributed system has multiple components and machines that communicate in order to create one seamless system for the user. In this case, the distributed system will consist of a server and multiple clients who will use the Draw It or Lose It web-based game. The client will access the game from a browser and make requests to the server that the server will then execute. The Gaming Room will use a wide area network, or the internet, in order to make their distributed system functional. In order for the system to remain connected and to avoid outages, both the client are server will need to ensure they are strongly connected to the internet and remain so during game play.
6. **Security**: In order to protect user information on and between various platforms, a secure operating platform, browser, and internet connection will be a necessity. Luckily, Mac operating system on the server side has great user protection and security capabilities. Every Mac comes with built-in antivirus software, frequent software updates to patch security gaps, and password capability as well as multiple user set-up available. This means that during communication back and forth with the server from the client, their information will be secure on the Mac operating platform. The Draw It or Lose It users will just need to ensure they are using a secure browser will accessing the game.