

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 | 1/26/2020 | Stephen | Update design documents requirements, constraints, and domain model. |
| 1.1 | 2/9/2020 | Stephen | Identify development requirements for server side, client, and Development tools. |
| 1.2 | 2/23/2020 | Stephen | Recommendations |

**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 to develop a web based game that is available across multiple platforms. They currently have the game working on Android but would like to include more platforms. The Gaming Room is looking to set up the environment to facilitate cross platform gameplay for the game they are developing.

## [Design Constraints](#_2et92p0)

* We will need to assess if the current servers used for hosting the android games will need to be updated/changed to work across many platforms (including expected increases in server load)
* We must allow for cross-platform communication. (A player on android should be able to play in the same game as someone on IOS or browser.)
* All applications need to pass any app store listing requirements.
* We will need to acquire a domain name if they do not already have one.

It is important to identify the design constraints early and accurately because they have major implications on the final product. Failing to work inside the design constraints will result in a project that does not function correctly in all cases which is not acceptable.

## [System Architecture View](#_ilbxbyevv6b6)

Not Required

## [Domain Model](#_8h2ehzxfam4o)

The UML diagram contains the com.gamingroom package. The UML depicts that there is an entity class that is the parent of all game objects other than the GameService Singleton. The Entity class has an id attribute that will be inherited by all subclasses. This allows us to have a unique id for each object that inherits entity. We can also see the relationships between all the classes. We know there is only 1 instance of GameService. That instance can have 0 to many Games. Each Game can have 0 to many teams, and each team can have 0 to many players. The GameService class showcases polymorphism as the getGame function is overloaded allowing for it to accept varying parameters. There is also a Driver class that is used to start and control our program, as well as a Singleton tester class used for testing purposes.

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | While it is possible to host a server on a Mac, it would not be the best option for this project as it may not work well cross platform. | Linux would be an excellent choice for server hosting. It is very common, and many people will know how to work on it. Also, it works well cross-patform. | It would be possible to host a server on windows for this project, however I think linux would be the better option. | Mobile is not a viable option for hosting servers of this scale. |
| **Client Side** | Mac will be required for this project. Not every team will require macs, but teams on ios will. Many developers will know how to use macs at a high level. Macs will be vital for developing the IOS and testing the product on mac. | Would work but is not necessary. We will have Linux servers, but you should be able to ssh into them from any platform. If someone prefers Linux, we won’t stop them. Linux will have no issue running the game because linux supports java. | Windows is the platform most people are familiar with, so we will have many of them. Windows can do most things except for IOS, but will be used because of its familiarity and cost. Windows will have no issue running the game because windows supports java. | Supporting both platforms will require a lot of work. We will need to develop with bot the IOS and Android SDK’s which doubles the workload. It is a requirement to support them, so we will do it. Will need phones from both platforms for testing purposes. |
| **Development Tools** | Mac would be a good choice at a tool for developing this project. It will have access to IOS swift development with the XCode ide. Mac also has access to Eclipse for browser development, and Android Studio for android development. | Linux would be an average choice for this project. We would be able to develop for Android and web with no issue, however we would require a VM to develop IOS. Linux has access to Eclipse and Android Studio IDEs | Windows would be an average choice for this project. We would be able to develop for Android and web with no issue, however we would require a VM to develop IOS. Windows has access to Eclipse, and Android Studio IDEs. | Mobile is not an option as a primary development tool, however we will need multiple IOS and Android devices to test the product. We could look in to supplementing our phone inventory with emulators. |

## 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**: For The Gaming Room’s Draw It or Lose It Operating Platform, I would recommend a Linux based server platform. Using a Linux based platform will allow Draw It or Lose It to be compatible across all the platforms that are required. The Linux platform is also very common in the industry, and it will not be difficult to find experienced personnel to develop, and work on this project.
2. **Operating Systems Architectures**: The Linux system architecture is what the MacOS is based off. Because of this, it is familiar to any users who use the mac operating system. On top of this, it is extremely common, which has lead many people to come up with systems that allow for effective communication with windows operating systems as well. Linux architectures are the most common and are very capable of cross platform communications.
3. **Storage Management**: For server side storage, currently, we Draw It or Lose It does not require significant storage amounts. However, I would recommend significantly more storage than the minimum requirement. This will allow for significant future growth without requiring any hardware changes to our system. As far as types of memory, we want to use the fastest memory that is available.
4. **Memory Management**: To give our users the best experience that does not leave them waiting for image downloads, or use significant amounts of their data (on mobile), we will utilize some techniques. We will have the client systems store images locally on the device in local storage, or cache. This will allow us to utilize a “download once” approach. We will be able to expand the items that are stored locally when needed. This technique will allow us to prevent the user from constantly waiting for downloads and using significant amounts of data. We will also be able to use the client device’s processing capabilities to manipulate images rather than saving various slightly different copies of the same image. This will reduce our memory sizes and download times.
5. **Distributed Systems and Networks**: Draw It or Lose It must be able to communicate between various platforms. In order to accomplish this, we can leverage some common protocols that are available across platform. For example, we can utilize the RESTful communication scheme which will allow for us to communicate across any platform. This would include using JSON as a simple, yet effective way to transfer data across all the platforms we require.
6. **Security**: Security is vital to the Draw It or Lose It project. In order to be successful we will need to be secure at all stages of the application. This will include having encrypted communication between client and servers. We can accomplish this by using the proper keys and encrypting data during transfer. We will also need to require users to have a strong username and password combination to secure their accounts. At the server level, we will need to maintain various best practices such as using the need-to-know and principal of least privilege. These 2 techniques will be pivotal in preventing bad actors from causing significant damage even if they don manage to pass our initial security measures.