

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. |

**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)

I didn’t realize this wasn’t required for the Week3 submission. Ignore this please.

| **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 people will know how to use macs at a high level. Mac will have no issue running the game because mac supports java. | Would work but is not necessary. We will have Linux servers, but you should be able to ssh in to 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. |
| **Development Tools** | Mac would be a good choice at a tool for developing this project. It will have access to IOS swift development as well as android and web development tools. | 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 | 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 | Mobile is not an option as a primary development tool, however we will need multiple IOS and Android devices to test the product. |

## 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**: <Recommend an appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments.>
2. **Operating Systems Architectures**: <Describe the details of the chosen operating platform architectures.>
3. **Storage Management**: <Identify an appropriate storage management system to be used with the recommended operating platform.>
4. **Memory Management**: <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.>
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).>
6. **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.>