

Draw It or Lose It

# **Project Software Design Document**

Version 1.2

## Table of Contents

Project Software Design Document1

[**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)4

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 5/23/21 | Gabriel Feng | Drafted and revised executive summary, design constraint, and domain model |
| 1.1 | 6/6/21 | Gabriel Feng | Drafted and revised evaluation of platforms |
| 1.2 | 6/20/21 | Gabriel Feng | Drafted and revised 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 currently has an Android only application, “Draw It or Lose It”, that they would like to develop into a web-based game capable of playing on multiple platforms, with teams of different players.

## [Design Constraints](#_2et92p0)

There can only be one instance of the game at a time.

There can be multiple teams per game and multiple players per team.

Must be functional on multiple platforms.

Teams and players must have unique identifications.

Different operating systems have different development tools.

## [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 ProgramDriver class contains the main() method, where the program is executed. The ProgramDriver uses the SingletonTester to ensure that only one instance of the game is running at a time. The GameService class is used in ProgramDriver, and contains all of the current games. The Game, Team, and Player classes all inherit from Entity, as they all exhibit similar behavior. The Game class contains all of the teams that exist inside it, and the Team class likewise for its players. ****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | MacOS is capable of hosting as a web server and can be relatively easy as Apache is already installed on MacOS. | Linux provides a variety of web server software, and commonly uses Apache to run servers. | Windows typically uses ISS, which is expensive to run web servers, but is also compatible with Apache, which is free. | Mobile Devices have apps available on their respective stores that can run web servers, most free of charge. |
| **Client Side** | MacOS by default has the web browser Safari, so testing the server and application on a Safari is a must if The Gaming Room wants to include Mac users. | Linux users have a wide selection of web browsers, so ensuring that the application is tested in several of the major web browsers would include most Linux users in the potential market. | Windows has Microsoft Edge downloaded by default, but many users still use Google Chrome. Testing on these two browsers is a must to include Windows users in the market. | Many browsers have mobile versions, so testing the application on mobile devices for emulators of mobile browsers is necessary for the inclusion of mobile users. |
| **Development Tools** | MacOS provides the XCode IDE, which develops for primarily Apple products. XCode supports many languages, but Swift is typically used to develop iOS and MacOS applications. | As an open-source OS, Linux has a variety of IDEs available for development, including mainstream ones. | Eclipse and Microsoft Visual Studio are primarily used on Windows, and support various languages. Both IDEs are quite costly for licensing. | Although I could not find IDEs that run on mobile devices, XCode and Visual Studio can develop applications for mobile devices, and they can run the application from there. |

## 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 recommend using the Windows OS and Apache to create a web server.
2. **Operating Systems Architectures**: Windows OS uses a layered design that separates user mode and kernel mode which provides more recoverability in the case of a system crash. Apache is a free open-source web server software that draws upon HTTP and is widely used. For example Facebook and PayPal use Apache.
3. **Storage Management**: I recommend using an SSD for storage, as it allows for faster access time and processing in general.
4. **Memory Management**: It would be best to limit the amount of memory used at a given time to the minimum required so that there is less load on the system and less cost.
5. **Distributed Systems and Networks**: As Draw It or Lose it will be hosted on a web server, if the system has a means to access the server through the network, any system would be able to access the application. So long as the OS has a client portal to the server, the user would be able to play Draw It or Lose It on any OS.
6. **Security**: Draw It or Lose It should implement a role-based security system as well as user authentication system. By protecting the system from unwanted access through roles and authentication, administrators and players can be separated while allowing both access to the application.