

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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 |
| --- | --- | --- | --- |
| 1.2 | 8/14/21 | Jorge Sierra | Updated final 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 client, The Gaming Room, currently offers the game Draw It or Lose It on Android, they would like to develop it as a web-based game and expand it to multiple platforms. The game is similar to the television game Win, Lose or Draw and it requires a large library of stock drawings as well as the ability to support multiple players and teams.

## [Design Constraints](#_2et92p0)

* Handle one or multiple teams.
* Handle multiple players in each team.
* Games and teams’ names constraints (unique).
* Only one instance of the game in memory at any given time.
* Render images from a large library at a steady rate.
* Support multiple platforms/OS.
* Game is web-based.
* Programming language is Java.

## [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 class Entity extends to the classes Game, Team and Player, meaning that each inherits attributes/functions from the Parent Class. We can see aggregation among the classes GameService, Game, Team and Player, where GameService has none to many Game(s), Game has none to many Team(s), and Team has none to many Player(s). The class ProgramDriver contains main, where the program is run, and it has an association with SingletonTester.

****

## [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** | The system supports server-side deployment through a native application: macOS Server. The costs are relatively high. | The system supports server-side deployment, it is open source which allows it to be free, but it has lower security protocols than other systems. | The system supports server-side deployment, with free and paying options. In our case the recommendation is to go with a paying option and depending on the volume of users the cost can be as high as in a Mac server. | The system does not support server-side deployment. |
| **Client Side** | High cost and it does not require a high expertise. Integration with other Apple devices. Cross-browser compatibility testing available through software and cloud. | Low cost, open source. More difficult to work on. Cross-browser compatibility testing available through software and cloud. | Medium cost, it requires low expertise. Provides a high reach due to it being one of the most popular OS. Cross-browser compatibility testing available through software and cloud. | Can be difficult to implement due to the large variety of devices. Provides a high reach. Cross-browser compatibility testing available through cloud, hard to implement. |
| **Development Tools** | Basic web code plus software like WordPress, vBulletin, and cPanel to run smoothly. Large library available, licensing costs apply. | Can run basic web code, WordPress, forum software. Large library available, mostly open source and free. | Can run applications that use ASP, .NET, Microsoft Access, or MSSQL databases. Large library available, licensing costs apply with free options available. | Libraries available as well as different frameworks, licensing cost apply. May require multiple teams. |

## 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**: The recommendation is to start development in Windows, due to the relatively low cost and low level of expertise required. It is essential to utilize a cloud-based architecture (serverless), allowing the provider to handle routine work such as provisioning, maintaining and scaling the server infrastructure.
2. **Operating Systems Architectures**: Windows based application allow access and proper management of system resources. The services provided by Windows enable applications to show a complex UI and UX. Windows uses a modular system, where one system communicates to the other and it creates a layer that can restrict the user access to the underlying hardware, although it can be overridden.
3. **Storage Management**: The adaptability of the devices that work with Windows allow to manage storage, adding or removing hard drives, factoring disks, etc. Depending on how the software is coded, the use of memory/storage can be very effective.
4. **Memory Management**: The adaptability of the devices that work with Windows allow to manage memory, adding or removing RAM. The program is required to have a large database of pictures and it needs to access to random pictures on each game. The database does not need to be stored in memory, since all the pictures will not be accessed at once, only one being showed, which would ease the use of memory.
5. **Distributed Systems and Networks**: Due to the characteristics of this game, there is no need to use a complex engine to develop the cross-platform functionality. The game engine GameMaker Studio allows to create simple games, such as point and click. This combined with it being a web-based game should allow for an easy transition into the different platforms, the main issue would be to ensure proper servers for a possible large player base.
6. **Security**: Windows allow to encrypt data and it comes with malware protection as well as firewall. For the web application it is recommended to use a secure communication channel between browser and server, such as HTTPS.