

<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 | <mm/dd/yy> | <Your-Name> | <Brief description of changes in this revision> |

**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 is a gaming company that is attempting to create a web-based game that can serve multiple platforms. The game they have created Draw it or Lose It is already available as an android application, but they have contacted CTS to make the game program more available on multiple platforms. The game loosely resembles the 1980’s television show Win, Lose, or Draw. The game is multiplayer based where each team has time constraints to guess images supplied to them by the user interface.

## [Design Constraints](#_2et92p0)

The game Draw It or Lose It is already available for android, but the client is wishing to expand the games to facilitate all platforms. To combat the current system restraints the game was shifted to a web-based platform which allows user across many different operating systems and platforms to enjoy the game.

## [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)

As we can see in the UML diagram below the program driver is created to run the program. The program driver contains the main code and will access the other classes to return information in them. The entity class is created as a parent group for Game, Team, and Player. It is utilized to create an ID for an instance of the game that is running/created. Once created the entity class allows the users to join one of the teams. Inside the game instance the team, player, and game are recorded and set. Each players ID will be saved in the player class that is created while their team is listed in the team class, and the game instance will save the team and players information as well as create a game ID that is saved in a system. The GamerService subclass cannot “live” without the Game, Team, and Player class. The GameService class is utilized to log and save all of the players, team, and game information as well as information pertaining to the amount of games, and the games ID. It is also utilized to create a new game by moving the players ID, team ID, and game ID to another instance of the game. As we can see their can be zero to unlimited instances of the game created which will allow as many users as necessary to play the game.

****

## [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** | Easy to access and make changes, but not as customizable as other platforms. | Easy to access and make changes and extremely cost effective, but not as user friendly as other platforms. | Easy to access and make changes, the most used platform with the most variety of software available. | Not as easy to manipulate when compared to other platforms. Also not as compatible across multiple devices. |
| **Client Side** | Mac is not as user friendly as other platforms, and many people would need to learn the platform. | Linux is an extremely cheap and or free platform, but it is the least user friendly requiring users to spend time learning the platform. | Windows is the most widely used platform, and is the most user friendly. It will have the least down time when for users to manipulate. | Mobile platforms can be tricky for users to operate, and substantial time is needed to make a game that will be available to multiple platforms. |
| **Development Tools** | Mac developers can utilize CSS, Python, HTML, JavaScript, etc. as well as multiple user-friendly IDE’s. | Linux developers can utilize Python, Java, ruby tec. as well as multiple user-friendly IDE’s | Windows developers can utilize CPP, C, Python, HTML, JavaScript, Java, etc. as well as multiple user-friendly IDE’s | Mobile developers can utilize PHP. HTML, C++, Python, JavaScript, ETC. There are also multiple user friendly IDE’s. |

## 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 would recommend a mobile platform like Android. Possibly android and IOS. The game is more than likely going to be played by users on a mobile devise opposed to a PC or Laptop. If the user opted to play the game from a platform it will still have a web-based version available, or it could be played in an emulator very easily from the Steam store.
2. **Operating Systems Architectures**: Like stated above I would go with a mobile platform like android, which is very close to and often compatible with Lunix. The game could be made on Linux quite easily to be utilized on a mobile platform.
3. **Storage Management**: I would allow the users device to store the game files which could easily communicate with a server to save stats. The current Game ID and information could be saved and hosted on one of the user’s devices which would allow for other users to easily join their game, and to save on server traffic. Call of Duty use to do this but phased it out due to the host having a benefit from a faster connection. This will not be an issue with this game due to it being turn based. Only a minute amount of data needs to be saved on the user’s device besides game which will include the users profile, game stats, and the current game instance they are in.
4. **Memory Management**: A game of this nature will likely not require hardly any memory. The game could be set up as a flash game online, and the images or graphics of the game are the only thing that might take up RAM usage. Any smart phone from the last 10 years would likely have plenty of memory to run a small framed game like this without an issue.
5. **Distributed Systems and Networks**: If the game is set up as a web-based application it could be hosted server side, but more than likely it would be much more feasible to have 1 of the players host the game on their device and have other users connect that way. This will drastically lower the server traffic but might cause some connection issues. If the connection issues persist the game could easily be ran directly off a web-page like many other flash games of this type.
6. **Security**: Security protocols would be a must since it is an open connection to an individual’s device. I would have a security team review the code, and security protocols to ensure there is no vulnerabilities in them that would allow access into an individual’s personal information. If the game is run as a web-based flash program security is not as important because the users own firewall and virus protection could block unwanted traffic very easily.