

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 | 09/16/2020 | Felipe de Araujo | Requirements to develop a multiplatform application |

**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)

Our objective with this software design document is to explain how to create the web game Draw It or Lose It, allowing players to pick unique names and enter in teams with also unique names.

## [Design Constraints](#_2et92p0)

While the back-end of this web game could be done with java, there must be a different programming language for the front-end of it. Focusing on the convenience of the users it is recommended to use a front-end language such as Flutter or React (React is preferable because it is widely used), while also planning its structure to be responsive (the same website can open and rescale automatically to basically any screen format), what will allow more users to enjoy the experience.

## [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)

There is a hierarchy starting from foundation that is the GameService, which is the principle class for the game structure, and provides the basic resources to ensure:

* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

Then the game, team an player. Those four classes are combined at the Entity class. There is also a Program Driver using a Singleton Tester in order to initiate.

****

## [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** | Apple being the only licensed maker of Macs, no longer make them thinking for the server side, so it won’t be efficient for that. | Linux is a cheap and good alternative for the server side, considering that the platform is part of a startup. | Windows is also good with servers, and they offer more support than the Linux alternatives, but they are usually more expensive. | In order to offer an online and connected experience, there must be a server side besides the localhost on a mobile device. |
| **Client Side** | The work, if done correctly, should be the same for all three instances. The preferable language is React, since it is close to React Native, and it allows a better communication with the server side, while providing great tools for responsiveness. | … | … | There could be a web PWA version of the platform, and also a native app, that could be programmed with React Native in order to work natively with android and IOS without the need of developing the application separated for each option. |
| **Development Tools** | Since it will be an online web game, the language is the same for all four instances. | … | … | … |

## 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**: Linux is a cost effective and friendly option, considering the requisites.
2. **Operating Systems Architectures**: It will be taken a layered structure approach, in other to maintain the modulatory and allow further expansions in an easier manner.
3. **Storage Management**: It will be used the Unit Track system, that is an online self-storage management system and works on any device with an Internet connection—Windows and Mac desktops, and any tablet or smartphone
4. **Memory Management**: It will be used the memory mirroring technic in order to create a secure and efficient system. If the goal is to display 200 quality images, there are ways of doing that that won’t require almost any power from ours server side. By using the image address that is given by third party websites, the server work will be transferred to those websites, and the implementation will be much easier, there must be a precaution regarding the copyrights of the images that are selected, but nowadays there a many websites that allow this process.
5. **Distributed Systems and Networks**: a network operating system will be chosen in other to work between multiple platforms.
6. **Security**: All the important information will be encrypted, and same goes for the database, which will have a firewall. The authentication process can have a two-step factor (not necessary).