

<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 | <07/20/2020> | <Marcus Bullock> | <Complete project guidelines.> |

**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 would like to make their android application into a web-based game that can be played on any application.

## [Design Constraints](#_2et92p0)

A few of the design constraint I can think of, is there was no budget or team size given. On top of just being a web-based game. I assume they want browser, but what browser will be the default? Will chrome be assumed as the default, safari, internet explorer Firefox?

## [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)

<It appears that Game Team and Player have a relationship with Entity. Because of this, Game, Team and Player are inherited from entity. According to the Graph, the aggregation shows that Game service at the top, with several games that leads to the relationship Game. Game then has a relationship with the teams, or Team in this case. Then Team ends with the relationship for players.>

****

## [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** | <Mac would not be the best use for server side, one of the biggest issues here would be limited data storage.> | <Linux would also not be the best choice, just simply for support issues.> | <Windows is possibly the best choice as it will be able to support and store the data server side quite easily.> | <The client is currently on mobile, however wants to branch to browser based, so mobile will no longer be the choice.> |
| **Client Side** | <Mac is pretty simplistic; the clients would be able to learn quite quickly as the user-friendly interface is very intuitive.> | <A bit of knowledge may need to be taught. Very few people again use Linux, or even know what Linux is in the first place.> | <Most people learn on windows, it’s common practice and one of the first operating systems most people operate.> | <The users are coming over from the mobile platform. They will be used how the game functions and will pick it up easily.> |
| **Development Tools** | <While this is image based, the tools Mac offers would come in handy. However, would not be the desired tools for the job.> | <The IDEs and support are limited for Linux. I would say that it’s rarely used for development if at all.> | <There are endless possibilities for windows. Depending on the route we want to go, with GIT, Visual studio devops, etc. at our disposal.> | <The development tools needed could be the same ones used prior.> |

## 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 using windows as the operating platform, I feel that it is the best well rounded platform and most used>
2. **Operating Systems Architectures**: <Windows is also easy to use, and most people have experience on windows devices.>
3. **Storage Management**: <Windows is also one of the most common for storage management, so there will be less likely a chance for support issues to pop up.>
4. **Memory Management**: <Windows is also excellent on memory management, unlike the other platforms. You can also increase the amount of memory needed, while in a few cases in the other platforms you cannot.>
5. **Distributed Systems and Networks**: <Going for a web-based game will be fine, the biggest issue is supporting every browser for each system. We will have to look at the functionality for each. Luckily, browsers can be downloaded on each platform if one is found to be more efficient. >
6. **Security**: <While Mac may be considered the most secure, windows itself can be secure as well though may require more upkeep.>