

# *Win, Lose or Draw*

# **CS 230 Project Software Design Template**

Bradley Magana

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 | 08/13/2021 | Bradley Magana | Recommendations have been updated to include an executive summary, design constraints, system architecture view, domain model, and finally overall evaluation. |

**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 would like to develop a web-based game that will be able to work with multiple platforms, the game currently only supports Android functionality. The game works by a drawing being rendered over a 30 second interval and a team attempting to guess what the image is within that time, if the team is unable to guess the drawing another team can answer after the 15 second mark. The Gaming Room needs our help facilitating the development of the game’s rollout to multiple platforms.

## [Design Constraints](#_2et92p0)

* Game needs to be rolled out to multiple new platforms.
* Design of the game when rolled out needs to be the same or like the Android version.
* Only one instance of the game can exist at any point in time.
* The game will have multiple users on a team at each time as well as multiple teams.

## [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)

Entity has a relationship with the Game, Team, and Player classes respectfully, because they all share this relationship with Entity that means that they all receive the same or similar information from Entity. Whenever these classes are taking information from the same source, they will exhibit similar properties within the classes. On the left we have Program Driver which shares a relationship, and we have GameService class connected to the Game class.

****

## 

## [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 must 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** | Very flexible and easy to use when trying to change or make additions of large and small changes. | Like what mac offers however from a cost perspective better than mac. | Far more available from a software perspective than apple while also staying better from a cost perspective like Linux. | From a severe side mobile devices are not really feasible on mobile devices, its better to have a stationary server. |
| **Client Side** | Cost would be the most of the four while time and expertise would be more than windows but less than Linux a middle ground of the two from that aspect. | Linux would require the most expertise with a cost like that of windows, this method would also require the most time. | Cost as well as time would be close to Mac, however expertise and price would be the minimum of the four. | From an availability and flexibility perspective mobile devise offer the best of the four, however making that information easily accessible on a mobile device is the most challenging of the four. |
| **Development Tools** | Relevant programming languages for mac consist of HTML, JavaScript, and CSS these are similar across all four of these operating platforms. | Linux relevant programming languages consist of HTML, JavaScript, and CSS, very similar to windows in what it offers and functionality but more difficult. | Windows relevant programming languages consist primarily of JavaScript, HTML, and CSS, Windows is easier to use than Linux and slightly more flexible than mac. | Relevant programming languages when looking at mobile devices would consist of primarily JavaScript, HTML, and CSS. |

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and recommend The Gaming Room. Specifically, address the following:

1. **Operating Platform**: A Windows approach would best suit the needs of The Gamer Room; this conclusion was reached frankly because of the wide range of software available on its platform and how inexpensive it is to work with. Furthermore, when using Windows, it requires the least amount of expertise/time and, as a result, will increase our efficiency in delivering our final product to the client.
2. **Operating Systems Architectures**: Windows 10 is built for simplicity and for those ready to get work done; within windows, core applications are some of the best productivity applications there is to offer. Windows architecture can significantly benefit our production regarding how easy and how little time it will take for us to complete specific projects while creating the application for different platforms.
3. **Storage Management**: Working with windows in my life has shown all its tips and tricks for Storage Sense; Windows can automatically clear up temporary files and transparent files that are not needed on the system and thus save us space and valuable storage. Windows Cloud Storage is also very efficient, especially in terms of accessing media remotely; this is another solution that could help us develop this project.
4. **Memory Management**: When looking at the subject of memory, Windows would allow for memory optimization to increase the mobile app's data efficiency. When moving pictures and just media in general, having these media groups readily available and efficiently processed is a massive game-changer in how efficient working on an application like this can be.
5. **Distributed Systems and Networks**: When looking at Distributed Networks, I was familiar with one application that stood out to me compared to the rest, allowing for full-scaled applications rolled out over multiple platforms. Develop4U allows for cross-platform compatibility; when creating a game or application, this method will enable you to take that game and have it transformed to work with IOS, Android, and other platforms for what you need.
6. **Security**: Windows comes preloaded with much of what we need in terms of firewalls and a certain level of encryption, allowing us to have excellent protection when developing this game across multiple platforms. Windows has tons of options, from every hour backups to virus scans that can be set up to scan on the hour to keep our information safe and ensure its protection.