

The Game Room

# **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 |
| --- | --- | --- | --- |
| 1.3 | 07/14/2021 | Travis Waggoner | Added recommendations. |

## [Executive Summary](#_sbfa50wo7nsh)

The Process I am proposing is the addition of the ability to add and retrieve information of the game information in a library format. This would allow users to access information about the game that they have not been able to retrieve before. In addition, this will create a library of games that collects information of what is being currently played.

## [Design Constraints](#_2et92p0)

The contrast of this application verses a web-based application is the web-based application would access the HTML part of the application and slow down the process. This occurs for a few reasons one of the largest is the accessing the information from one location and the communication between the two programs.

## [System Architecture View](#_ilbxbyevv6b6)

## [Domain Model](#_8h2ehzxfam4o)

This UML class shows how every class with interact with one another. The simplest is the program driver interacts with the singleton tester. Next the largest and most important is the Entity class which is the primary class that receives information from every class. The Game service sending information to the game and all three interact off one another. All then send the information back to the entity class.

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac with the right software would host the web sever with the right programs.  Does offer serv-based deployment with licensing from apple or 3rd party. | Linux will have no issues with connecting to server and hosting. Advantage would be dedicated hosting.  Does offer server-based deployment with licensing and without licensing | Can run server we with windows server because of the background programs removed.  Does offer server-based deployment with and without licensing. | Mobile devices would struggle to host a server due to their limited capacity. Does not offer server deployment with licensing or with out |
| **Client Side** | Mac requires training and specially made software in order to run multiple types of clients. Mac produces are the most expensive out of all the options.  Required to be able to write and accept code for any client. | Linux would run multiple clients well with support depending. However, writing a client would be easiest for Linux.  Be able to host multiple clients without interference with one another. | Windows is the most supported and would make use of already existing software. This would be the least time consuming but get expensive depending on the clients.  Be able to run all clients with out running into hardware strain. | Mobile Devices would be the most time consuming since there are not many host applications out there. This could be the least expensive short term if implemented correctly.  Be able to consistently maintain systems |
| **Development Tools** | Mac runs their own programing language based on C. This would require specific software to create applications. This would be expensive. | Linux uses a know language of C and could easily find many tools to create the software.  Should be little licensing cost other than clients | Windows is written in all the C languages and would be the most universal to implement.  Licensing cost for each developing tool. Many free options available. | Mobiles use many languages Java is most used for Android and apple uses their own language.  Developing tools limited and would need multiple machines to complete. |

## 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**: A appropriate operating system for game room to start utilizing draw it or lose it would be Linux. This is because the programing languages are already in place to utilize these different clients. It would also be possible to edit or change how the clients interact with the host.
2. **Operating Systems Architectures**: Linux will provide a optimal interaction between the server client and the clients themselves. This will allow for more usage in power requirements from hardware to perform more tasks.
3. **Storage Management**: The storage will utilize different directories using the tree-structured directories and each individual client as well as the data management for the data files will also be kept in their own directories with preferably with their own hardware for storage.
4. **Memory Management**: Linux will use the least amount of memory for its operating system as well give us the most control of the utilization of memory for each individual client. This benefit will give us the ability to control the memory usage by hand rather than allowing the operating system to decide for the user.
5. **Distributed Systems and Networks**: Linux would utilize the least amount of data over the network as well as give the user the most control over the bandwidth for each independent client. Disruption will be brought to a minim with the lower bandwidth and any issue will be corrected with reconnection routines build into the network setup.
6. **Security**: Establishing role control over the users will give the most protection over the users. This will prevent other users from being able to access information for other users or hosts’ data bases. Also, we will give control to the user on how much information is accessible to users individually. This will allow for the users to be able to allow more information to specific users over others. All security and privacy settings will be defaulted to there highest setting in order to protect users from the very moment they access the programs.