

# **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 |
| --- | --- | --- | --- |
| 1.5 | 8/7/2022 | Mahdi Bigdely | Provide more info about server to choose the platform |

## [Executive Summary](#_sbfa50wo7nsh)

Based on the customer’s need and request, the application should run on a server, and need to be scalable, meaning that when the users/games increase, it would be easy to accommodate with the increased demand of resources such as processing all the requests, without impacting the quality. The other requirement would be to run and process the main calculations and renderings in the server side, so that gamers/users’ resources are not impacted.  
The application should be compatible with various digital devices such as mobile phones, tablets and PC/laptops, and user should be able to run the game with a regular web-browser and internet connection.

Choosing the host/server is very important, as it dictates and somehow limits us to certain tools for development and maintenance of the application.

## [Design Constraints](#_2et92p0)

The application should be able to recognize each user’s device, the screen size and quality, and then render the game based on the best quality that is available for that device, such as screen size, resolution, and even connection speed. It also needs to coordinate and adjust the game based on different player’s devices to create a smooth experience and enjoyable game.

## [System Architecture View](#_ilbxbyevv6b6)

* Application (for windows, Mac, Android, or iPhone) is installed on the device and commands are sent to server through that using a secure and authenticated methos
* Server receives the command, authenticate, and authorizes it if everything looks good and processes the command
* The game should be save on the local hard drive of the server at certain intervals and an encrypted copy of the data should be kept on a totally separate cloud storage

## [Evaluation](#_2o15spng8stw) of the server side

* **Evaluate various platforms for their characteristics, advantages, and weaknesses for hosting a web-based software application.** Consider the following in your evaluation and articulate your findings in the software design template:
  + Does each of the operating platforms offer a server-based deployment method where the website will be hosted?   
    Yes, both OS that I will talk about offer a server-based deployment method.
  + What are the potential licensing costs to the client, The Gaming Room, for the server operating system?  
    If a Linux based operating system is chosen as the server, the cost is minimum or free, as most of the Linux based servers are open source and can be used for free, or low cost. There are many other programming tools and most of the major programming languages are available for free for most of Linux based operating systems.
  + The user’s license could be sold as monthly membership or one time payment. The maintenance, support and troubleshooting of each server

| **Development Requirements** | **macOS Server** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Not available anymore  As of April 21, 2022, Apple has discontinued macOS Server | Open source  Free  Efficient and good return of investment  Secure  Not much direct support | More support  More secure  More expensive (not free)  More support  More familiar for general audience | NA |
| **Client Side** | There are many developers and resources to design and program Mac native applications for Mac, as well as the web browsers that are already coming with the OS | Web-browsers can be used (already in the OS) and the applications need to be designed for | Just like Mac and Linux, web-browsers are available, and there are many programmers and resources to develop client side application | Applications can be accessed through Play Store and App Store, for Android and iOS |
| **Development Tools** | Different IDEs and development tools such as X-code is available  Swift and Objective C are tow programming languages to be used in Mac | Different IDEs such as Eclipse are used  C is the default langue in Linus | Visual studio is one of the IDEs that is native for Microsoft  The default programming language is VBA which makes it hard to code for networking purposes | Android Studio or XCode  Java, Kotlin and Swift are the programming languages for Android and iOS |

In general, desktop (client side) applications provide more tools and options for users compared to web-browsers that have many limitations. On the other side, client-side desktop application will incur extra cost and time and need constant upgrade; while web-browsers are multipurpose and the update need to be done in the server side and once done, everyone will benefit from it, without even noticing there has been an update, except from the performance improvement or experience enhancement.

(Browsers may get updated as well).

Installing applications may be cumbersome for some users as it takes more time and resources.

The downside with using browsers and server-side applications is the connection reliability, response time that will be out of users control, most if the time, and may negatively impact users’ experience.

## Recommendations

1. **Operating Platform**: Linux based servers could be a strong, while cost effective option that can’t host and run multiple server applications and can handle heavy traffics of internet and network. The saving could be spent on designing a strong and responsive CMS/server application to enhance user’s experience
2. **Storage Management**: Local saving on the server in addition to the simultaneous cloud saving on a totally different and independent storage backup system
3. **Memory Management**: For the dedicated server, DDR4 3200 Hz is recommended and the server should be upgradable to add more memory if needed. The server response and render time is very important and impactful on user’s experience, and both Ram and HDD play a big role here. A slow Ram may not be able to fetch data (such as pictures) fast enough from HDD. An agile and responsive and agile need to be used for memory management to enable in-memory storage of live and recent data, caching of most and frequent visited pages, have an execution plan for queries contexts and queries cache such as query results, and efficient data integration, context processing, and user sessions.
4. **Distributed Systems and Networks**: Using a headless CMS such as Strapi which uses RESTful API would be an option to accomplish differences between various devices that connect to the server to play the game. As the communication will be going through the server and CMS and each device will send the same and standardized request (using the application or browser) and gets its specific response based on its configuration.
5. **Security**: The sensitive information of the users should be encrypted and stored on a separate; location, while the main authentication to access/ modify such information should be scrutinized. Two-step verification, using HTTPS or SSL protocols and ports (at least for authentication part) are very important. However, HTTPS or SSL may not be the best option to use for the whole time when the players are trying to load high resolution pictures, as these protocols are s are slower compared to the HTTP and not encrypted connections.