

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

Contents

[**CS 230 Project Software Design Template** 1](#_Toc46161564)

[Table of Contents 2](#_Toc46161565)

[Document Revision History 2](#_Toc46161566)

[Executive Summary 3](#_Toc46161567)

[Design Constraints 3](#_Toc46161568)

[System Architecture View 3](#_Toc46161569)

[Domain Model 3](#_Toc46161570)

[Evaluation 4](#_Toc46161571)

[Recommendations 6](#_Toc46161572)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/19/2020 | Benito Patino | Initial entries |
| 1.0 | 08/01/2020 | Benito Patino | 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 is seeking to develop a web-based game that serves multiple platforms. Currently, the game is only available as an Android application. The multi-platform game will allow for multiple teams and multiple players to compete against each other. The Gaming Room is seeking our assistance for both development of the application, as well as setting up the environment to host the multi-platform web-based application.

## [Design Constraints](#_2et92p0)

* Server Load
  + Given that there will be an increase in users, there needs to be a plan to handle the server load from the increase in users.
  + Server load may be distributed along regional areas across the United States or any other countries where the game will be available.
* Client Development for Multiple Platforms
  + Desktop
    - If the client application is to be a native Desktop application for both MacOS and Windows, different programming languages and frameworks would be required for this implementation. A Windows platform best option would be the .Net Framework with C#. As MacOS best option would be Swift. Java and JavaFx could be implemented for both Windows and MacOS.
    - If the client application is to be web browser based, then HTML, CSS, with a JavaScript framework would be the best implementation. React and Vue would be the best options for the JavaScript framework. However, it is important to consider users who are using outdated web browsers.
    - In a Linux environment, Java and JavaFx would be the best option.
    - Other options for these three platforms include: C++, JavaScript Electron, and Visual Basic.
    - Code sharing between Windows, MacOS, and Linux platforms.
  + Mobile
    - The creation of the IOS application could be implemented with Swift, Kotlin, and C#.
    - Code sharing between Android and IOS platforms

## [System Architecture View](#_ilbxbyevv6b6)

## [Domain Model](#_8h2ehzxfam4o)

The **Entity** Base class provides properties and methods that are inherited by **Game**, **Team**, and **Player** class. Each one of these classes requires a unique **name** and a unique **id.** Although the Game, Team and Player class do not have direct access to the **name** and **id** properties, the public methods provided by the **Entity** class (*getId(), getName())* allow for access to those private properties. This is a perfect example of encapsulation and inheritance.

The **GameService** class makes an association of 0 to many instances of the **Game** class. The **Game** class makes an association of 0 to many instances of the **Team** class, and the **Team** class makes an association of 0 to many instances of the **Player** class.

****

## 

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | One of the biggest disadvantages for implementing MacOS for hosting the web-based application is: cost. Another disadvantage includes the specific hardware that is needed for running MacOS. Apple product specialists will be needed for maintenance and implementation of Apple products. | Linux is the most used O.S for hosting. Linux servers are stable, no notable hardware constrains, and the cost for licensing would be cheaper than MacOS. One disadvantage could include the lack of training for administrators at The Gaming Room. | Windows provides excellent support for its products. However, the licensing cost for deploying Windows servers could be quite high. | The biggest disadvantage is restricted resources for running the application. Every mobile device has different hardware specs and these resources compete with other applications on the mobile device. If the application runs natively on the mobile device, there would no server costs. |
| **Client Side** | There would need to be teams or individuals that understand the MacOS platform and the corresponding Apple device (i.e. laptop, iMac, etc.), as well as be up to date on any software or security updates. Cross platform solution for desktop client include Electron or Java FX. Xcode with Swift can be incorporated for creating native MacOS client. | There would need to be teams or individuals that understand the Linux platform, as well as be up to date on any software or security updates. Creating desktop client applications by incorporating GNMOE desktop environment. | There would need to be teams or individuals that understand the Windows platform, as well as be up to date on any software or security updates. Making use of the .Net Framework and Windows Presentation Foundation (WPF) for making the Windows client using C#, C++, or Visual Basic. WPF is used to render user interfaces for Windows applications. | There would need to be teams or individuals that understand the IOS and Android platform, as well as be up to date on any software or security updates. React Native could be incorporated for cross platform mobile platform clients. Other cross platform technologies include Xamarin. Native |
| **Development Tools** | The top IDE’s include XCode, and IntellJ, Visual Studio for Mac, Visual Studio Code.  Languages include Java, Swift, C#, Kotlin | The top IDE’s include Netbeans, Eclipse, IntelJ, Visual Studio Code  Languages include Java, C#, Kotlin | The top IDE’s include Netbeans, Eclipse, IntelJ, Visual Studio Code, Visual Studio  Languages include Java, C#, Kotlin | On IOS platform, development could only be done on Mac using IOS tools. XCode would be best IDE for accomplishing this goal. The Swift programming language would be used with XCode.  On the Android platform, development could be implemented on Mac, Java, or Windows platforms. Android Studio is the best IDE choice. Language choice could be between Java or Kotlin.  Cross platform development can be performed with Xamarin using Visual Studio for Windows and Visual Studio Mac for Mac OS. C# is the programming language used with Xamarin.  Languages include Java, C#, Kotlin, Swift |

## 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**: Deploying the application on web server hosted on a Linux server is the best option. Cost, stability, and scalability are all benefits of this platform. Compared to a Windows Azure server, the learning curve for understanding such a platform is less.
2. **Operating Systems Architectures**: Linux is the leading OS used in servers. There are multiple variations of Linux, including CentOS, Debian, Ubuntu Server, and many others. Ubuntu Server is known for its stability and many features, but these added features requires more storage for the OS. However, Ubuntu Server is a tried and true version of Linux that will offer stability, security, and scalability.
3. **Storage Management**: Implementing Amazon Simple Storage Service(S3) will provide a scalable, durable, and high performing storage service. Back-up storage could also implement the same solution.
4. **Memory Management**: Linux memory management includes both physical and virtual memory. Virtual memory implements paging – in data is stored and retrieved from secondary storage.
5. **Distributed Systems and Networks**: A common solution for server outages is to have different regions across the country that have servers running. Also, we recommend having backup servers on standby to migrate users from a down server. This distributed architecture will allow for multiple clients to have access to the application, even when one server may be experiencing an outage.
6. **Security**: The Principle of Least Privileges should be implemented for the User role. A User role will only have specific privileges for playing, starting, exiting a game. Other privileges could include changing their username or adding a profile picture. In the case of any unauthorized access of a user account, proper implementation of the need-to-know principle would prevent any access of resources that do not concern the specific role. High level security solutions could include enforcing the use of long complex passwords and two-tier authentication – by requiring a user to enter a random code sent to their email or phone.