

The Gaming Room

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 01/26/2024 | Wurzburg Brumaire | Application expectations |

**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)

This outlines the approach for a web-based game application. With this document we’ll provide a clear solution for the issues with the software design. We want users to be able to access the game through different devices. To achieve that we’ll dive into different web technologies like JavaScript and HTML.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

Security – With it being a web-based application the security aspect becomes a concern. We’d need to use authentication systems, and encryption for sensitive information to protect user data as well as access.

Browsers – There is a countless number of browsers on the web so we must make sure the application will run on a great deal of those browsers constraining the users to specific browsers will affect the user’s opinions on the application. So, we must investigate different browser standards, features, and possibly finding ways to run on older browsers.

Interface Design – The interface should be visually appealing and responsive, So we must ensure we have a great sense of how the layout should be and take into consideration different sizes of monitors or TV screens.

## [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)

The UML class diagram gives an overview of the main classes and relationships.

Game Service – This class is an instance of a game in the application. It contains information such as players names, game count, team id, and the game name.

Entity – This class contains attributes shared by other classes in the model. It inherits from other classes using object-oriented programming.

Player - This class contains information on a player playing the game. Information like the players score, and status in the game.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

For Macs you’d need flexible terminal commands to configure the server, access, or make any changes. An advantage to using Mac is it can be upgraded, there’s different options for different web hosting requirements. A weakness would be that its not the best web hosting service its usually not preferred. Mac’s server-based deployment option is macOS Mojave. On the client side it’s easy to use with a bit of training, the cost would be like the cost of using windows.

Linux is like Mac with an advantage when there is a flaw with security it can be dealt with before it becomes a problem. A weakness it has is its difficult to find application to support its hosting requirements. On the client side you’d need someone who is well informed on the system basically an expert. For the cost I believe it has the lowest cost out of all the systems.

Windows is used more than any of the other systems, people are more familiar with its systems but the downside to it is its prone to viruses, so you have to be cautions while using it. On the client side there is very little training needed to use the system since almost everyone has used it and the cost would be about the same as Mac.

For mobile devices they’re becoming more popular every day, people want to be able to game on the go rather than being stuck in a stationary place. The advantage to it is being able to reach more people and is usually cost effective. A disadvantage to it is that it has poor security. Mobile devices give flexibility to clients and developers to see changes or or updates from anywhere allowing them to monitor the app.

Each system has a tool it primarily uses like Mac having swift, Linux using eclipse, windows using virtual studio. They all impact your team and the project some having a higher impact than others. With windows using virtual studios it wouldn’t set the team back much due to its ease of use. Eclipse with Linux might bring in a learning curve to get used to its interface. But any of the software systems can be sufficient to achieve the goals of the game. Each system can successfully use HTML/CSS and Java Script. The cost to run these tools all depends on the team competence in these languages, if not we’d either have to train or hire a second team to come in with this knowledge.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Macs have secure operating systems they tend to be reliable and resistant against viruses. But Macs OS may not be compatible with foreign programs. | Linux is an open system with customization but brings a learning curve because it’s not as common as mac or windows and it may have issues with software compatibility. | Windows OS is compatible with many devices and systems. Most people have seen or used a Windows OS. But Windows are more prone to malware and being breached. | Mobile devices are streamlined as far as what they can do but they are efficient when it comes to apps that are compatible with it. |
| **Client Side** | The cost of developing on a Mac has a higher cost than any of the other OS. The hardware alone on average cost more and at time you may need work arounds to use certain software with it so that adds on time constraints as well. | Linux essentially has the same constraints as Mac when it comes to compatibility and cross training. | Developing on windows may be the most reliable system to work on due to its familiarity among most people and compatibility with most programs. | Developing on a smart phone is a very inefficient way to develop software. |
| **Development Tools** | Objective-C and Swift are the primary languages used to develop Macs | Linux is mostly written in C. | C, C++, and C# are the primary languages used to develop Windows | Objective-C and Swift are the primary languages used to develop mobile devices |

## 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**: Using Azure would give you a scalable and versatile operating platform to take Draw it or Lose it to different computing environments. A cloud-based server platform provides the flexibility you need and is cost effective ensuring you can meet the requirements of the clients.
2. **Operating Systems Architectures**: Microsoft Azure supports different operating system architectures like the  Windows Server and Linux distributions such as CentOs. These architectures provide flexibility in deploying and managing the Draw it or Lose it game with different platforms, making sure compatibility and performance optimization.
3. **Storage Management**: For storage management I’d recommend a relational database management system. A relational database management system provides the ability to store and retrieve data in a strutted and more efficient way. With relational database management systems, they also bring security, recovery, and data backup.
4. **Memory Management**: Azures memory management uses resource allocation capabilities to make sure you’re getting the best performance for the game. You’ll have control over memory allocation, and you’ll be able to utilize resources based on demands. Also, with Azure you can manage runtime environments with automatic scaling to handle workloads effectively managing memory usage in the game.
5. **Distributed Systems and Networks**: As far as distributed systems the approach that will give the game the edge it needs is Azures virtual network. It allows you to communicate between various platforms in a secure way. You can monitor traffic with its features so you can distribute incoming traffic across the different data centers.
6. **Security**: security is by far the most important part of any project, and I believe Azure will provide you with the best features available. It offers encryption, access control, and secure communications. With Azure it is makes monitoring and detecting threats efficient, making sure the privacy of user’s information are secure throughout all platforms.