

<Name-of-Software-Application>

# **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 |
| --- | --- | --- | --- |
| 1.0 | 07/14/2021 | Charles Hopkins | This will incorporate side-by-side team gameplay. Unique identifiers per user whilst guaranteeing an instance of one game. |

## [Executive Summary](#_sbfa50wo7nsh)

This will address the problem of a four round minute guessing game where each game will have multiple players assigned to a single instance. By creating one instance of a game stored in memory at any given time, utilizing these unique identifiers thereby creating multi-teamed multi-instanced games. Each game lasting one minute.

## [Design Constraints](#_2et92p0)

* Multiple Teams
* Unique Player Identifiers
* Four Rounds
* Minute Long Rounds
* Multiple Teams with Multiple Players per Instance

## [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)

From the UML diagram below, the ProgramDriver runs the SingletonTester. The GameService node accesses the game which accesses the team which accesses the player node. Game, Team and Player nodes are nestled under the Entity node for a singular instance, which is why GameService is not because it would be utilized outside of the instance.

****

## [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 has to 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** | Commands for terminal made to configure the server, changes to the environment and overall access. As with Windows and even Linux specifically Ubuntu, third-party applications that monitor packets sent and received can aid as an extra security measure to control the flow of data, preventing malicious intent. | Commands for terminal made to configure the server, changes to the environment and overall access. As with Windows and even Mac, third-party applications that monitor packets sent and received can aid as an extra security measure to control the flow of data, preventing malicious intent. | Added management applications available and drivers as opposed to other OS’s. As with Mac and even Linux, third-party applications that monitor packets sent and received can aid as an extra security measure to control the flow of data, preventing malicious intent. Alternatively, Windows provides access control features which can also aid in controlling the flow of data whilst monitoring malicious software intrusions. | It would be wise if the server was not mobile with a singular IP for domain purposes. In most cases, it can be difficult to protect against malicious intrusions. With software dedicated to jailbreaking and or even rooting devices, protecting against malicious modifications to packets sent and received, it could be difficult to limit these intrusions. There are also software’s dedicated to mimicking mobile devices for the purpose of cheating in games, such as blue stacks. That being said, data could also be encrypted to control what is and isn’t genuine. |
| **Client Side** | Cost will be about the same as Windows. Standard skill & time will be required. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Cost of development will be low. Skill will need to be the best available. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Cost of development will be like Mac. Skill required is less sophisticated than Mac though. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Because mobile device requires a web-based html this will be more one size fits all. Though, this method will be a little more difficult to implement as there are many variables to consider, such as screen size, aspect ratio and different hardware that functions differently. |
| **Development Tools** | While developing for mac, Eclipse for Swift and even C++ can be utilized. Though macs can run nearly any language we would need regardless of cost intense ones, it would be wise to limit usage to HTML, CSS and Java libs to allow cross support and functionality with all platforms. | While developing for Linux, Eclipse with the usage of a plethora of languages. Though Linux can run nearly any language we would need regardless of cost intense ones, it would be wise to limit usage to HTML, CSS and Java libs to allow cross support and functionality with all platforms. | Given that Windows is a far easier development platform, and supports a plethora of languages, we will still want to limit usage to HTML, CSS and Java libs to allow cross support and functionality with all platforms. | Utilizing Swift and or Java even C++, a multitude of applications can be made, but in the interest in developing a game with cross-platform support HTML, CSS and Java libs will provide the advantage. Furthermore, each of the games will work nearly exactly identical to one another. |

## 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 good starting base to work from would be a mobile platform. If this application will contain a mobile version, then starting with mobile version would be wise as it would need to be the least sophisticated version. Starting with the mobile version would provide a baseline for what else will be needed in future platforms. While at the same time, not over doing the development of this application.
2. **Operating Systems Architectures**: As development for both iOS and Android will be the start basis, it would be wise to lead with either C++ or Java/HTML/CSS for development to be sustainable and adaptable for both OS’s. Development of the server will commence along with the client which should cut down greatly on other platforms work later builds.
3. **Storage Management**: Utilizing an online approach to developing with HTML/CSS with Java libs will allow primarily server-side storage. All that would need to be considered client side, is storage of the current instance and characteristics of the current run. Once the user connects to the server, they would download their unique user data for their personal account.
4. **Memory Management**: Developing a database of different pictures would be the primary source of data stored for game files. Instead of storing these on the client’s device, they can be stored primarily server side to conserve space on the user’s device. At the same time these files could temporarily be stored client side, no different than how YouTube operates.
5. **Distributed Systems and Networks**: Given that most of this application would be server side, everything involved would be predominantly stored server-side. Meaning that all devices connecting to the server will have access to the same content. However, these devices will need the capacity to connect to the server to play as intended. Without any form of internet connection, the game would be rendered unplayable. Publishing this game on both the iOS and Android stores though would be relatively easy except for needing to make small payments in order these games to be displayed on their respective stores.
6. **Security**: Provided games won and lost are stored in a counter of sorts per account, this information would need to be maintained and secured on a server independent of the device and game server for storing images. Though iOS and Android have servers for storing user data like game data, it would not be wise to store this for all player because it would be unreachable to different devices. However, there are some services which would allow users to store data from a plethora of sources or devices. Google cloud offers plenty of APIs in a plethora of languages allowing for user data storage at flexible rates. Security would be maintained by google and offers authentication features ideal for game security in order to protect not only game data but also user information.