

<Win loose or draw>

# **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 | 03/16/2022 | Landon Johnston | Added Executive summary, constraints and domain model |
| 1.1 | 03/31/2022 | Landon Johnston | Added evaluation |
| 2.0 | 04/12/2022 | Landon Johnston | Added Recommendation |

**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)

We are looking to recreate the game called win loose or draw that’s on android platform. We are tasked to develop it as a web-based game able to be multi-platform. We will provide a library of stock drawings for clues. The game will give clues steadily over 30 seconds. If the correct awnser is not given then the other team will get 15 seconds to guess. The game will consist of 4 rounds. We will need to allow unique names if multiple members on each team.

## [Design Constraints](#_2et92p0)

Multi-platform issue because both Android and IOS use different compatibilities then web-based games.

The ability to check for Unique names and give each team an unique ID

## [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)

We start with the ProgramDriver class that contains the main method for the application. It uses directed Association with the SingletonTester class. Next the Entity class calls 3 classes, Team, Game and Player. These three Classes from the entity class. The team class can have a player but not a game and the game can have both a team and a player. The gameservice contain all three, game, player and team.

**"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)

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** | **Advantages:**  Limited  hardware  Security  Mac exclusive  **Disadvantages:**  Limited  hardware  Limited hardware  Expensive  Closed source | **Advantages:**  Limited  hardware  Open source  Low cost  Supports range of hardware  **Disadvantages:**  Applications not might be supported | **Advantages:**  Limited  hardware  Supports a lot of third party applications  Widely accepted  **Disadvantages:**  Licensing fees  Less secure | **Advantages:**  Portable  IOS support  Limited  hardware  **Disadvantages:**  Not practical or widely used  Would need either physical servers or a lot of devices |
| **Client Side** | **Advantages:**  Limited  hardware  Secure  Wide range of support  **Disadvantages:**  Have to have a Apple product with mac os | **Advantages:**  Tree structure  Open source so lots of web supported applications  Limited  hardware  **Disadvantages:**  not as user friendly | **Advantages:**  quick development  Limited  hardware  easy to use | **Advantages:**  Limited  hardware  3 main operating systems  Apple  Android  Google  Ease of use  **Disadvantages:**  Limitations on what can be done with hardware |
| **Development Tools** | Must be approved by apple  Lots of documentation  eclipse for java | **Advantages:**  can be used for multiple os using VM  no licensing fees  Limited  hardware  eclipse for java | **Advantages:**  Limited  hardware  Most documentation and easier to learn  Eclipse and visual studio | Swift ui |

## 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**: The best operating system to use is a Linux server
2. **Operating Systems Architectures**: Linux is the best option because it is free, it is also secure, being open source, the hardware adaptability is very flexible.
3. **Storage Management**: The best storage management is to use either Microsoft azure or AWS. Both allow cloud storage which allow for speed and scalability.
4. **Memory Management**: Using Linux allows us to capitalize on its partial memory management using virtual memory. This cuts out the extra requirements for physically memory. However having the correct amount of physical memory is needed. DDR4 Ram in the Linux server
5. **Distributed Systems and Networks**: Using Azure or AWS allows for ease of use when using a distributed system. Load balancing will be key for keeping the uptime to a maximum. Which maintains the best cost model.

1. **Security**: Security is important when hosting accounts and where a lot of users will interact. We will implement A firewall that disables/block all not needed ports and protocols. We will Also implement ssl for our website, using a vpn when transmitting account information. Also having a strong read write files so when they users interacts with or game, they can’t change whats there.