

Draw it or Lose it

# **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 | 05/20/2023 | Mitchel Harmon | Changes made to the executive summary, design constraints, and domain model. |

**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 game draw it or lose it is currently only available on android and the goal of The Gaming Room Project is to develop a web-based game that is available on multiple platforms. Based on the 1980’s Tv game Win, Lose or Draw, it will play similarly were the multiple teams guess the image before the time runs out.

## 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)

The design constraints are:

* Needs to have multiple teams with unique names
* Each team has multiple players
* Only one instance of the game
* Needs to run on Multiple platforms

## [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)

Game, Team, and Player classes are inherited from the Entity class. The GameService class references the Game class, The Game class references the Team class, and the Team class references the Player class. The Entity class is a superclass in which we use the super keyword in the constructors of the subclasses to be able to access the constructor of the superclass.

**"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** | MacOS has a simple server based deployment which add to it’s ease of use and boasts about their proven security.  MacOS however often has less options when it comes to hardware. | Linux is an open sources OS which has a low hardware requirement.  Linux doesn't have as much applications made with Linux availability. | Windows has a vast amount of hardware options and has a simple server based deployment.  Windows is however less secure than competing OS. | Mobile devices store persistent data and the code cannot be seen by the user.  However mobile devices on the server side need to integrate with cloud services or physical servers. |
| **Client Side** | MacOS has a large selection of well supported web browsers but requires a Apple product to be able to use. | Linux due to being open source works with every web browser and has a wide range of well supported tools for developers. | Windows has easy cross platform testing and has quick development and deployment but has difficult time testing for MacOS browser. | Mobile devices typically have a longer development time and has difficultly testing other environments. |
| **Development Tools** | Mac OS has Mac OS X and has a costly Apple dev program but can easily run other OS through a virtual machine.  Visual studio code, xCode, Atom. | Linux has a open source community to help with development and can run the other OS through a virtual machine.  Visual studio code, eclipse, Atom. | Windows has extensive documentation for deployment and can run Linux through a virtual machine .  Microsoft visual studio, Visual studio code, eclipse. | Androids have Android Studio.  IOS has xCode.  Iconic is a hybrid open source mobile device development tool. |

## 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**:
   1. I would recommend Windows OS over the rest as windows has more software availability and offers a large suite of products that will help in development.
   2. Windows OS also has less restrictions than say apple and while that may open the doors to potential security risks when it comes to development can allow more efficient development.
2. **Operating Systems Architectures**:
   1. Windows OS User mode acts as an interface between a user and the computers resources.
   2. Kernel mode deals with the behind-the-scenes things such as inputs and outputs, networking, hardware management and so on.
3. **Storage Management**:
   1. For storage Management Using cloud-based storage would allow the game to only utilize enough storage space to run the game. Cloud-based storage can be upgraded to meet the needs of the game so as the game grows you can add more storage space to fit the need.
4. **Memory Management**:
   1. Windows OS uses demand paging which only brings pages into memory if the executing process demands them. Demand paging makes it so more processes ccan be loaded reducing the context switching time.
5. **Distributed Systems and Networks**:
   1. With Cloud storage management it’s also important to choose one that is available on all platforms such as Microsoft Azure. Azure is made by Microsoft which offers maximum uptime and is able to integrate and manage environments with their various tools and services.
6. **Security**:
   1. Since we are using cloud services it is important to choose one that securely holds data. Again, I would recommend Azure which has multilayered security and 3,500 security experts to monitor and safeguard cloud data. Another aspect of Azure that piqued my interest was Azure’s core privacy principle which states that you own your data, and they will never use it for marketing or advertising purposes.