

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 | 03/16/25 | Evan Nagy | initial creation of the CTS Software  Addition of Server side, Client Side and  Recommendations  Addition of Recommendations |

**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 have been asked to by CTS facilitate the development of a web based game Called Win it or Loose it. The purpose of this web application is to allow it to be used on multiple platforms. There can be either one or several more teams within the game. Each team can have a player. There will be one game instance that can have 1 or more teams with one or more players all these things must be unique. In order to avoid multiple game instances, we should use a singleton pattern to enforce this rule.

## Requirements

* the application will display images from a library of stock drawings as clues.
* A game consists of four rounds of play lasting one minute each.
* Drawings are rendered and are fully complete at the 30-second mark.
* If the team does not guess the puzzle before time expires, the remaining teams have to offer one guess each to solve the puzzle with a 15-second time limit.
* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

Draw it or Lose it is going to be a web-based application. This application is meant to also be available on multiple platforms example, “android and Linux”. We will be using Javascript for this application primarily. This is due to the main requirement of it being web-based. We will also be using Java for this project since it has synergy with android studio which can be written in either java or Kotlin. Later, we will have to delve into other frameworks such as android and IOS to create mobile applications connected to this website.

We will have to conduct more research into alternative API. For example, Amazon web services, they are heavily focused on Internet related activities and may help us deliver our code onto the internet more efficiently.

## [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 Program Driver Class will be used as our Main class which is used to start the game. We will associate a Singleton Tester with the main logic in order to test if our instances are the only instances of our game. Our ProgramDriver will be using a single instance of our GameService class which will have all the components needed to play a game.

For GameService, we will set its constructor to private in order to prevent more instances of that game being created. We must always use the getInstance() method in order to instantiate our GameService. getInstance() will have a check within it to determine whether we already have an instance of a game or not. It will then start an instance .

Our Driver class will have access to the addGame() Function only after the GameService has been started. This function will have the ability to check and prevent any game instances with the same name from being created. This class will have an itterator that prevents games with the same name from occurring. After that, the list called games will update its list to include this new one.

AddTeam() method will be used to add another team to a game that has been created. In order to stop another team object with the same name being added to a game. We will need to include an iterator to check the team names. Finally, the list teams will include this new team object.

AddPlayer() method can be used to add a player to a team. A team must first be formed in order to add a player. To stop a player name conflict, an iterator will be employed to prevent a name collision from occurring.

The classes Player,Team and Game are all subclasses of our Entity object. We have protected attributes, name and id. Only overloaded constructors may be used in order to prevent Null objects from being created.

Multiple Object-Oriented Programming (OOP) techniques are being displayed in this UML design. Overloading of constructors and extending the Entity Class are displays of polymorphism and inheritence. When we refer to our logic for adding teams, we are demonstrating encapsulation and abstraction. Although it is impossible to create a team object, we can still create a team via the addTeam Method.

**"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** | Mac is naturally Intergrated with other Apple devices and contain Tool Kits pre-installed. Mac is considered to have a good level of security attached to their products. Other advantages include:user-friendly, unix compatable,  Disadvantages include cos, scalability and low customization. | Linux is an opensource operating system with thousands of variants. It is lightweight, has a high level of performance, stable, reliable and secure.  Advantages include:  Cost effectiveness, flexibility, and a wide selection of tools.  Disadvantages include:  Incompatible with some software solutions and it contains a learning curve | Windows is a closed source operating system that is well integrated in society. Thus it contains strong compatibility advantages when compared to other operating systems.  Advantages include:  User Friendly, Contains Gui tools for managing your computer and easy integration with Microsoft technologies.  Disadvantages include:  High cost, low customization and heavy reliance on Microsoft | Mobile Devices are convenient and portable. They hand a user friendly touch interface and diverse hardware available. They have a few advantages such as mobile services and location based services.  Advantages Include:  Accessibility, touch interaction and application for the web  Disadvantages include:  Network dependency, battery life, and performance limitations. |
| **Client Side** | Expensive for users, Moderate Time and expertise Required  Training is required to navigate the OS and it`s functionalities | Alot of training is required to correctly use linux. It is very Flexible, support must be carefully considered when developing applications for this OS. It has a high level of performance and customizable. | Windows is considered the default for clients. It is the standard OS for your average person and thus, most people know how to use it. It comes with a wide variety of tools. | There are a large number of designs for Mobile devices. This includes hardware and software. You have to optimize performance and security because they are not known for such things. |
| **Development Tools** | Many languages can be use with this OS. Languages such as front-end languages, (HTML/CSS) and back-end languages like JavaScript. It has access to many libraries need for development and access development tools like Github and Pycharm. | Many languages can be use with this OS. Languages such as front-end languages, (HTML/CSS) and back-end languages like JavaScript. It can leverage the customization of linux systems and can run Ruby,PHP and Python. | Many languages can be use with this OS. Languages such as front-end languages, (HTML/CSS) and back-end languages like JavaScript. Developer Tools include Eclipse, command prompt and PyCharm. | Many languages can be use with this OS. Languages such as front-end languages, (HTML/CSS) and back-end languages like JavaScript. It can access many libraries that support front-end development and can access languages like,php,C++ and python |

## 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**: In the case of gaming, I would recommend the Windows OS. All games are supported on Windows because It is the most widely used OS in the world. Therefore, the client will be instantly familiar with this operating system, and this will lead to a smoother experience for the user.
2. **Operating Systems Architectures**: Windows has a Simple GUI for you to use. Windows supports a wide variety of languages and IDE`s for development. This can include VSCode and Eclipse.
3. **Storage Management**: due to windows ease of use. It is easy to navigate through storage with their Gui or command line tools. In fact, the OS manages the storage though disk manager. Windows also has a cloud-based service that you can use to keep your documents equal across systems. This will allow collaboration and storage management to be easy as our project grows in scope.
4. **Memory Management**: Windows users can choose either virtual memory or physical memory. Virtual memory is often used for larger programs and would be considered a boon for us. Being able to protect your physical memory and extend its use are two major benefits over pure physical memory.
5. **Distributed Systems and Networks**: working with the OS is a likely problem. The developer team members should communicate with one another to find solutions for issues like lagging, queuing problems and overloaded servers
6. **Security**: Windows can user Windows Defender, “an anti-viral software”. Which can be constantly active on a user`s system. VPtii services are also available to further secure user data. When considering this, it will be crucial to conduct regular security checks to guarantee user safety and prevent unauthorized access to any personal data.