

Draw It or Lose It

# **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 | 11/13/2022 | Lawrence Abbott | Initial Software Design Template Creation |
| 1.1 | 11/27/2002 | Lawrence Abbott | Operating Platform Evaluation |
| 1.2 | 12/09/2022 | Lawrence Abbott | 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)

Creative Technology Solutions (CTS) is an industry leader and has a rich history in consulting and development services. On behalf of The Gaming Room, the following document outlines the requirements and proposed solutions for a development strategy to expand the web-based game *Draw it or Lose it*, from solely an Android platform game to a multi-platform game. It should be noted that while CTS will begin software development of the application and lay the groundwork for the project, there is an expectation that The Gaming Room will continue software development services, thereafter.

## [Design Constraints](#_2et92p0)

* The game application will be designed with the ability to have one or more teams involved in gameplay.
* The game application will be designed with the ability to have multiple players on a team.
* Only one instance of the game can exist in memory at any given time.
* Active games names must be unique, with the ability for users to check whether a game name is currently in use.
* Active team names must be unique, with the ability for users to check whether a team name is currently in use.
* Active player names must be unique, with the ability for users to check whether a player name is currently in use.

## [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 **ProgramDriver** class is the main driver for the game application.
* The **ProgramDriver** class is associated with the **SingletonTester** class.
  + More specifically, a **ProgramDriver** uses a **SingletonTester** to ensure that there is only one instance of the game in memory.
* The **GameService** class has a zero-to-many relationship with the **Game** class.
* The **Game** class has a zero-to-many relationship with the **Team** class.
* The **Team** class has a zero-to-many relationship with the **Player** class.
* The **Game**, **Team**, and **Player** classes are inherited from the **Entity** class.
  + This inheritance relationship speeds development time as instantiated **Game**, **Team** or **Player** objects can use the properties and methods implemented in **Entity**. For example, an instantiated **Team** object can utilize the getName() method of **Entity**, without the need to implement it within the **Team** class.

**"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** | Features easy to use GUI for easy administration without the need for an extremely tech savvy administrator. The server software is proprietary, which means it requires a yearly license purchase. However, recently Apple has announced it has discontinued macOS Server and has deprecated many of the previously supported services. This uncertainty suggests the need for ongoing configuration changes as Apple removes previously supported features. | Known for its performance and reliability. This non-proprietary option is what powers most of the internet today. The fact that it is open source comes without the need to purchase a yearly license. However, the heavy use of command line interfaces when configuring the server requires a more tech savvy administrator. Formal support is not an option, but support can be found in the vast-open-source community forums. | Features an easy-to-use GUI, which can simplify the administration process. However, our research suggests that this option is less stable when compared to Linux and Mac server options. Additionally, as Windows Server is a proprietary server, it will require a yearly license activation fee. Also, recently Microsoft updated their license requirements, requiring one license for every two physical cores of the server | While there are some notable mobile server options, our research suggests that this type of server lacks robustness in terms of scalability and performance when compared to the traditional server options. |
| **Client Side** | Due to the nature of the client-server architecture, and the fact that the application will be accessed using an internet browser, developers will be required to develop using Mac solely for the purposes of testing the application using the Apple Safari browser. Otherwise, it will be left up to the developer as to which operating system they prefer to develop with. | Due to the nature of the client-server architecture, and the fact that the application will be accessed using an internet browser, it will be left up to the developer as to which operating system they prefer to develop with. | Due to the nature of the client-server architecture, and the fact that the application will be accessed using an internet browser, developers will be required to develop using a Windows operating system solely for the purposes of testing the application using the Microsoft Edge browser. Otherwise, it will be left up to the developer as to which operating system they prefer to develop with. | Due to the nature of the client-server architecture, and the fact that the application will be accessed using an internet browser, developers will be required to test the application using an iOS device solely for the purposes of testing the application using the Apple Safari browser. However, it is not recommended that a mobile device be used in the development of the application. |
| **Development Tools** | The MySQL database can be managed using Sequel Ace or MySQL Workbench. There are multiple options for IDE, including PhpStorm (PHP), IntelliJ (Java). Development and testing can be conducted using Google Chrome, Mozilla Firefox, Apple Safari, and Microsoft Edge, at a minimum. | The MySQL database can be managed using DataGrip or MySQL Workbench. There are multiple options for IDE, including PhpStorm (PHP), IntelliJ (Java). Development and testing can be conducted using Google Chrome, Mozilla Firefox, and Opera, at a minimum. | The MySQL database can be managed using MySQL Workbench. There are multiple options for IDE, including PhpStorm (PHP), IntelliJ (Java). Development and testing can be conducted using Google Chrome, Mozilla Firefox, and Microsoft Edge, at a minimum. | Development and testing can be conducted using Google Chrome, Mozilla Firefox, and Apple Safari, at a minimum. |

## 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**: It is our recommendation to use Ubuntu, a Linux distribution, for the backend server operating system, as our developers have used this server option in the past and are familiar with the setup and administration process. Additionally, no additional cost will be incurred for a yearly license purchase, as Linux comes free of charge. In terms of scalability, reliability, and performance, we feel this is the best option to expand Draw It or Lose It to other computing environments.
2. **Operating Systems Architectures**: The Ubuntu project, as described on the Ubuntu wiki, is committed to a regular six-month release schedule, with security updates for 9 months after release (longer still for Long Term Support releases). Only in severe cases, stable releases receive updates, but as a general rule, releases are not changed unless absolutely necessary. This speaks to Ubuntu’s reliability and security.
3. **Storage Management**: MySQL is our choice for storage management. This open-source option, known as the world's most popular database, is a great option for scalability and efficiency.
4. **Memory Management**: Linux uses an implementation of demand paging and virtual memory to manage memory inside the system. In terms of virtual memory, every memory access uses a virtual memory address. When the CPU decodes an instruction, it translates the virtual memory address encoded in the instruction into a physical address that the memory controller can understand. The most advantageous part of the virtual memory implementation is that the operating system can load programs that are larger than the physical memory. Demand paging is often referred to as lazy evaluation as pages are brought into memory only if the executing process demands it, at which point the demanded pages are swapped from secondary storage to main memory. This effectively decreases the time needed for loading at program startup and reduces the time associated with context switching.
5. **Distributed Systems and Networks**: Draw It or Lose It will be available on various platforms, and as such, will require efficient communication between each of the connected devices. In taking this into consideration, we feel that it is best to design the game as a client-server architecture, whereby communication between the various clients and the server will take place via a REST API. This has the benefit of being flexible as the application can be written in whatever programming language is appropriate for the platform it is running on, while providing a common interface for accepting HTTP requests. This design choice supports multiple clients and leaves the door open for future expansion onto additional platforms.
6. **Security**: Security will be managed at the server level whereby users will be required to authenticate through a specific endpoint of the REST API. Once authenticated a user session will be granted to the user. User privileges will be role based and access to resources will be restricted to the roles a user is a member of. Each API endpoint will include a check of the user’s role and access will not be granted unauthorized users. Furthermore, when establishing connection to the game all internet traffic will be sent over HTTPS to protect the exchange of data while it is in transit.