

Draw It or Lose It

# **CS 230 Project Software Design**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design** 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 | 03/20/25 | Jon Scott | Completed design template |

## [Executive Summary](#_sbfa50wo7nsh)

The Game Room is a game design company that wants to expand its mobile game, Draw It or Lose It, to reach web-based devices. The web application must allow users to play with one or multiple teams, with each team comprising multiple players. All game, team, and player names must be unique, and only one instance of the game can run at a time.

## Requirements

* The application must be web based
* The application must connect users to Draw It or Lose It’s servers
* The application must allow PC access

## [Design Constraints](#_2et92p0)

**Budget or Financial resources**: The Game Room does not want to go over their project budget, so they have contracted out the web application design to Creative Technology Solutions.

**Compatibility:** The web application must be playable on MacOS, Windows OS, or Linux OS because they are the top three operating systems for personal computers.

**Language**: The web application must be written in JavaScript, Swift, or C to support any of the top three operating systems.

## [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 Entity class is the Game, Team, and Player class's parent class. The children classes inherit the attributes and methods of the Entity class. The GameService, Game, Team, and Player classes all have a zero-to-many relationship between each other. The ProgramDriver and Singleton tester have a uses relationship. Each class is encapsulated within themselves. The toString method is polymorphed to suit the needs of each child class. All private and protected attributes/methods are abstracted from the driver 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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | MacOS server offers Linux and Windows protocols, with a few Apple exclusives, such as Apple Filing Protocol. MacOS server is not popular and is only suited for smaller Apple networks. MacOS server has been discontinued but it was not free while it was still being offered. | Most Linux Servers support standard Internet protocols as well as Apple’s AppleTalk. The wide variety of Linux flavors offers great freedom when choosing the best OS. Ubuntu Server is an excellent free option for hosting web applications. | Windows Server supports standard Internet protocols like the TCP/IP suite. Windows XP offers integration with AppleTalk as well. Windows Server is not free and costs around $200 USD per OS copy. | FireBase is a hosting platform made specifically to host mobile applications. It offers integration with Android Studio and Google Play. FireBase starts free but scales to cost more depending on its use. |
| **Client Side** | MacOS is free to download and use, but a file transfer fee is charged if the User wants to download their old data. MacOS is considered the easiest to use but has the most restrictions out of the three top OS’s. | Linux is a free and open-source OS. Linux is considered less user-friendly than Windows but offers greater freedom. | Windows OS costs vary depending on the agreement or plan purchased. A single activation key costs $100, but special agreements are made for companies needing multiple workstations. Windows is easy to install and use. | FireBase’s cost is dependent on how it is used. The more the application needs to be used, the more FireBase bills. FireBase integrates seamlessly with Android devices, and there is no additional cost associated with Android OS. Configuring FireBase is not user-friendly and requires an IT professional |
| **Development Tools** | MacOS is primarily written in Swift and includes a collection of IDEs and tools. MacOS supports VS  Code (not Visual Studio),  JetBrains IDE, and Apache web server. JetBrains and VS code both require paid licenses to use in an enterprise environment. | Linux is primarily written in C and supports various IDE’s and tools. Linux supports Visual Studio, Eclipse, and MySQL. Linux supports the Apache web server, which can be used to host web pages. MS Visual Studio requires a paid license to use in an enterprise environment. | Windows is also primarily written in C and supports many of the same IDEs as Linux. Windows also supports the Apache web server. MS Visual Studio requires a paid license to use in an enterprise environment. There is no cost associated with Apache Web Server. | FireBase includes tools like Android Studio, Cloud Firestore database, and cloud storage to improve developers' ability to host mobile apps. Firestore database requires a paid license. |

## Recommendations

1. **Operating Platform**: A Linux distro should be used to host the web application because of the low cost, low restrictions, and wide variety of features Linux offers. A Debian distro specifically should be used because it is free, stable, secure, and provides cloud images for most major cloud platforms. A stable operating platform is essential for Draw It or Lose It to keep players happy and engaged during games.
2. **Operating Systems Architectures**: Linux architecture has three main components. The first component is the loadable Kernel Modules that allow different functionality from the Kernal. The second component is the Kernal itself, which provides the functionality for all processes to run. The third component is the system-shared libraries that allow the system to make requests from the Kernel. The three-part Linux architecture handles all processes efficiently and effectively.
3. **Storage Management**: Linux uses UNIX’s standard file system to store information. The standard file system creates/manages partitions, adds/deletes files, and optimizes the file system. Storage management saves data that Draw It or Lose It will need in the future, like scores, levels, and team information. This information is perpetuated after a system shut down but is slower to access than random access memory.
4. **Memory Management**: The first 4 GB are reserved for the Kernal to use as a cache. The remaining memory is managed by a page allocator that allocates and deallocates memory as needed. The Draw It or Lose It Game will efficiently utilize memory by only allowing a single game instance to run simultaneously. When information will be needed repeatedly or soon, the page allocator may store it in memory for faster retrieval.
5. **Distributed Systems and Networks**: Debian servers include protocols that allow compatibility with other OSes like MacOS or Windows. Good Web servers have built-in redundancy, so if one component fails, another can resume operations without interruption. Redundancy ensures players are not inconvenienced when servers eventually fail. Uninterruptible Power Supplies can prevent servers from experiencing a blackout by providing a backup power source if the main source fails.
6. **Security**: Debian Servers are highly secure. Most malware targets Windows systems because they are the most common OS. Debian handles Access Control and Authentication at its lowest level, ensuring they function correctly. Debian also offers many encryption solutions to safeguard data in transit or storage. Debian provides security support for its releases to ensure Draw It or Lose Its servers remain secure.