

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 | 09/22/2024 | Matthew A Keaton | Changes made to this revision are executive summary, design constraints, and domain model. |
| 2.0 | 10/06/2024 | Matthew A Keaton | Changes made to this revision are included in evaluation |
| 3.0 | 10/20/2024 | Matthew A Keaton | Changes made to this revision are included in recommendations. |

## [Executive Summary](#_sbfa50wo7nsh)

Our new client, The Gaming Room, wants to develop a web-based game that is available on multiple platforms. The new web-based game will be called QuickDraw, which is based off their current game, Draw It or Lose It, only available on Android. The game involves teams guessing puzzles based on pre-rendered images from a stock library. The application must support multiple teams and players while ensuring unique team and game names. A solution is for the game to have the ability to have multiple teams with multiple people on each team where each individual team name is unique. Also, by implementing a singleton pattern to ensure only one instance of the game exists in memory at any given time.

## Requirements

**Business Requirements:**

* Develop a web-based game to serve multiple platforms
* Support for multiple teams and players in each game

**Technical Requirements:**

* Only one instance of the game can run in memory at any given time
* Ability to create unique team names

## [Design Constraints](#_2et92p0)

The design for *Draw It or Lose It* involves creating a web-based game that can support many teams and players, with unique names for each. The system will be built so it can grow easily as more people play and will allow real-time communication, like guessing answers quickly. It will work smoothly on different devices (phones, computers, etc.), and keep user data safe by using secure connections. We’ll also make sure that only one game is running at a time to avoid issues.

## [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 UML diagram shows how the game system is organized. The GameService manages everything, ensuring there’s only one instance of the game running, using the Singleton pattern. Classes like Game, Team, and Player share common properties like id and name through the Entity class using inheritance. Games are made up of teams, and teams are made up of players, which shows composition. This setup helps organize the game, making it easier to manage unique games and teams. The lines connecting each class represents the relationship between each class. Between some classes is “0…\*” which describes the relationship between the classes as “zero or more.” Meaning, the GameService class can have zero or more Games, the Game class can have zero or more Teams, and the Team class can have zero or more Players.

**"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** | Mac OS has a Unix based foundation making it secure and efficient. Advantage of Mac OS is the user-friendly graphical interface. However, Apple products are generally more costly. | Linux is open source and the most popular OS for web servers. Most Linux distributions are free making it cost efficient. Linux uses command prompt to manage making it a bigger learning curve. | Windows OS has simple graphical user interface. Advantage is its high compatibility. However, requires a paid license which can be costly for large scale deployments. | Not traditional web servers having limited resources. Provides adaptability and low latency for local apps. Weakness is limited performance abilities and battery powered. |
| **Client Side** | Uses IDE only for Mac which apple products tend to be more expensive. Extra time and expertise may be needed for developers to understand Mac programming languages. | Linux is free and open source making it cost effective. Longer time due to steeper learning curve. Expertise in cross platform frameworks needed. | Licenses can be more costly for large scale deployments. Faster development than Mac or Linux. Expertise in C# or .NET which are commonly used. | Testing on mobile apps can add some extra costs. Review process can add take extra time to complete. Extra expertise needed to understand mobile app designs. |
| **Development Tools** | MacOS languages include Swift, C++/C, and JavaScript. IDE’s in Mac, Visual Studio (VS) Code and Xcode. | Linux OS languages include C++/C, Python, Java, and JavaScript. IDE’s in Linux, Eclipse, Qt, and VS Code. | Windows OS languages include C#, C++, and JavaScript. IDE’s in Windows, VS Code, PyCharm, and Eclipse. | IOS uses Swift, Android uses Java. IDE for IOS is Xcode and for Android is Android Studio. |

## Recommendations

1. **Operating Platform**:
   1. A recommended operating platform for the game Draw It or Lose It is Linux. Linux is an open-source operating system that is made publicly for anyone to access or modify which provides the most flexibility to customers allowing them to customize the operating system in their own way. Even though Linux is an open-source system it still has strong security features that minimizes any vulnerabilities that risk user data and the applications integrity. Linux scalability makes it an ideal platform for servers and applications that handle large amounts of data and traffic (LogicMonitor, 2024). This high scalability makes Linux an ideal choice for the business to anticipate growth and be able to manage the increasing workload. Since Linux is open-source this reduces the costs needed for specific licenses where organizations can run and install the operating platform without extra costs.
2. **Operating Systems Architectures**:
   1. The security architecture provided by Linux operating platform makes it the best choice for a game like Draw It or Lose It. Linux supports multiple methods of authentication methods such as using strong passwords, smart cards, biometrics, and encryption methods like digital signatures (LigicMonitor, 2024). These features provided a high level of security that allows for access control for multiple users. Linux kernel plays a significant role that offers several advantages in making it the preferred choice over other operating platforms. The kernel is “the main component of Linux that controls the activity of other hardware components” (GeeksforGeeks, 2023). Linux’s kernel helps promote stability, modularity, and security that make it a versatile and reliable operating platform.
3. **Storage Management**:
   1. An appropriate storage management to use in Linux is logical volume management (LVM). The three main components of LVM consist of physical volumes, volume groups, and logical volumes that offer flexible storage management (Cavanagh, 2024). LVM allows the resizing of logical volumes that enable administrators to adjust storage capacity as needed. This storage component makes for a great choice for managing storage in the game Draw It or Lose It.
4. **Memory Management**:
   1. Linux uses multiple memory management techniques that boost its effectiveness and performance. Linux kernel implements paging which is “the movement of pages in and out of the main memory and storage” (Rawat, 2024). Paging is efficient because it allows the use of physical memory by dividing it into fixed size blocks called “pages.” This process supports the virtual memory in Linux to load only the necessary pages into the random-access memory (RAM) and utilize more memory than what is physically available.
5. **Distributed Systems and Networks**:
   1. To have the game Draw It or Lose It communicate across different platforms we can use a distributed software system such as a client-server model that will connect multiple devices through a central server. The server can manage and handle game logic and player data that makes it possible for players to communicate with each other on different devices. A tool such as RESTful API (Representational State Transfer Application Programming Interface) will allow players to communicate in real time. Using RESTful API allows different software applications to communicate with each other over the internet using methods such as GET, PUT, and DELETE. So, when a player moves it sends a request to the server using RESTful API, where the server processes the request and updates the game. This will help ensure that the players who are playing the game are able to see updates and any actions in the game immediately.
6. **Security**:
   1. Security is extremely important in an operating system to protect user information. Linux is known for its built-in security features that ensure confidentiality and data integrity for users. Using authentication methods such as MFA (multi-factor authentication), biometric scanners, and digital signatures in Linux will ensure that people who are authorized to use the resources are who they say they are allowing secure access into their accounts. Using best coding practices such as input validation, data sanitization, and output encoding will help prevent common malicious injection attacks such as SQL injection (SQLi) and Cross-Site Scripting (XSS). Linux open source allows anyone to be able to find, report, or fix a security issue (LogicMonitor, 2024). This allows more people to be able to fix a vulnerability before a hacker can exploit it rather than waiting for one client or vendor to fix the issue which may take extra time. These security features offered by Linux operating system helps create a safe and secure environment for players by protecting their information and playing the game on different platforms.

**References**

Cavanagh, L. (2024, March 6). *Using LVM in Linux to Manage Storage Devices*. Liquid Web. https://www.liquidweb.com/blog/lvm-in-linux-to-manage-storage-devices/

GeeksforGeeks. (2023, September 25). *Architecture of Linux Operating System*. GeeksforGeeks. https://www.geeksforgeeks.org/architecture-of-linux-operating-system/

LogicMonitor. (2024, July 24). *9 Reasons Linux is a Popular Choice for Servers*. LogicMonitor. <https://www.logicmonitor.com/blog/9-reasons-linux-is-a-popular-choice-for-servers#h-3> enhanced-security

Rawat, H. (2024, March 3). *Linux Memory Management – All You Need to Know*. Medium. <https://hemantra.medium.com/linux-memory-management-all-you-need-to-know> d1dbdda8b386