

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 | <05/24/2024> | Rose St.Clair | The Cover page, history, and summary and more has been changed to reflect the recommendations that were submitted. |
| 1.1 | 06/06/2024 | Rose St.Clair | Updated the client, server, and recommendations based on the updated pros and cons for each OS that I have gained information about. |
| 1.2 | 06/22/2024 | Rose St.Clair | Updated the operating platform, operating system architectures, storage management, memory management, distributed system and networks, and security section. |

**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 objective of the gaming room initiative is to adapt the Draw it or Lose it game for compatibility across various platforms, excluding Android. This game entails teams competing against each other to decipher assigned pictures within a time frame, spanning four rounds. Should a team fail to guess the picture, members of the opposing team each have 15 seconds to attempt for points.

## Requirements

The program's specifications dictate its compatibility across platforms beyond just Android. Each team will consist of multiple players, maintaining balance. Simultaneously, only one game can occur for the teams. The uniqueness of the game and team names enables others to check availability, ensuring distinctiveness.

## [Design Constraints](#_2et92p0)

They necessitate it to be web-based for accessibility. The game will comprise four rounds, each lasting one minute. It will center on several teams attempting to correctly guess a picture pulled from a library to earn points. If unsuccessful, the opposing team gains 15 seconds to guess. The client requests the picture to be rendered at the 30-second mark. Only one game can be stored in memory at a time, achievable through unique identifiers for the game, team, or player.

Top of Form

## [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 serves as the primary handler for the game, team, and player classes, inheriting their attributes. All four classes created for this project mutually reference each other. The programdriver class is responsible for constructing the project to fulfill the client's expectations. Through the programdriver class, we can access and execute the program. The singletontester facilitates testing within the design constraints.

**"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** | Advantages: By accessing the Apple website, you can gain access to iOS for a fee of $20, which also grants access to software aiding webpage creation.  Disadvantages: Limited hardware choices and necessitates a Mac system. Updates solely available through Apple and come at an additional cost. Highest expenses involved. | Advantages: Linux usage is free, though the cost may vary depending on the server requirements. It offers open-source software and operates on minimal hardware specifications. A cost-effective solution with excellent stability, regularly updated to protect against security risks. Its decentralized open-source nature enhances flexibility.  Disadvantages: Limited adoption due to user unfamiliarity. Software availability is relatively low. | Advantages: Windows requires a purchased license for usage, with various server packages available at different prices depending on requirements. It enables straightforward server deployment and boasts extensive software compatibility. Additionally, it provides a diverse array of hardware options and benefits from frequent updates due to its large user base.  Disadvantages: However, it comes with a high operational cost, despite being the preferred choice for many users. It's less secure compared to Mac and Linux systems, being closed-source and restricted to updates solely through Microsoft. Moreover, customization options are limited. | Advantages: Costs vary from zero to $320 per month, contingent upon CPU and memory requirements for the desired program. With a solid reputation, covering the cost is feasible, and it enjoys widespread daily usage. Capable of managing server-side calls to optimize database queries and storing persistent data securely. The code remains concealed from users.  Disadvantages: Integration with cloud servers or a physical server is necessary for server-side operations. |
| **Client Side** | Advantages: Apple's closed-source nature contributes to relatively low costs, although it's not as commonly utilized. It boasts a broad selection of well-supported web browsers and offers numerous tools for creators. Facilitates easy cross-browser testing and maintains moderate development deployment and time efficiency.  Disadvantages: Usage necessitates an Apple product with a Mac operating system. | Advantages: Being open source, this option entails low costs and provides extensive browser support alongside numerous creator tools. It features a tree file structure accommodation for organization.  Disadvantages: However, proficiency in usage is required, necessitating user education. | Advantages: Additionally, it provides a diverse array of browsers along with tools for creators. Facilitates easy cross-platform testing and enables quick development and deployment.  Disadvantages: Testing for MacOS browsers is comparatively more challenging. | Advantages: Your proficiency lies in Android app development.  Disadvantages: Challenges arise in testing across different environments and browsers, leading to longer development times. |
| **Development Tools** | Documentation for deployment on Mac is available. Running this on Mac and Linux is convenient through a virtual machine. However, the application must undergo approval by Apple. Joining the Apple Developer Program costs only $99 per year and provides access to Eclipse for Java and Mac OS X. | Deployment can be done at any given time. Running it on Windows and Mac is seamless via virtual machines. It features both a shell prompt and terminal. Being an open-source community, there are no licensing costs involved. Additionally, it includes Eclipse for Java. | You have the flexibility to deploy at any time. There's comprehensive documentation available for deploying on Windows. Linux can be run through a virtual machine. There are no licensing costs involved. It provides Visual Studio Code for JavaScript development, along with Eclipse. | Xcode 12 is utilized for iOS deployment. Joining the Apple Developer Program for iOS costs only $99 per year. Additionally, it features SwiftUI for development purposes. |

## 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**: Upon further investigation, my recommendation remains in favor of Linux. It presents numerous advantages with the only drawback being the necessity for users to undergo training. Linux provides superior security measures and ensures constant updates.
2. **Operating Systems Architectures**: The rationale behind this choice is the absence of licensing fees, albeit necessitating training for users. Linux provides access to a wide array of excellent software and programming languages for program development. Moreover, it operates in a decentralized manner and demands lower hardware specifications.
3. **Storage Management**: The default XFS filesystem is highly versatile, catering to various needs effectively. It organizes each project efficiently, ensuring convenient access. You'll find all the necessary resources readily available to craft your desired creations.
4. **Memory Management**: It partitions the available physical memory into pages, each with its unique physical address. These pages are then linked to virtual addresses, granting each process its designated memory space corresponding to the physical memory. The memory management supervisor administers the allocation process.
5. **Distributed Systems and Networks**: Linux employs various software components, amalgamating them into a unified system. Furthermore, these components are interlinked via both local and wide area networks.
6. **Security**: Linux is renowned for its robust security features, with a reputation for minimal susceptibility to viruses and malware. This aspect plays a crucial role in safeguarding clients' information from unauthorized access and misuse. Moreover, Linux's security is continuously reinforced through regular updates, ensuring resilience against potential security threats.