

The Gaming Room

# **CS 230 Project Software Design**

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 | 11/20/2024 | Jordan Bankston | Implemented revisions for the cover page, the document's revision log, the executive summary, the system architecture framework, the domain model, and the design limits. |

[Executive Summary](#_sbfa50wo7nsh)

Designing a web-based application, 'Draw It or Lose It,' for The Gaming Room is the objective of the endeavor. Currently accessible on Android, the objective is to expand its availability to encompass a variety of platforms. This will improve user accessibility, encourage team-based interactions, and optimize the game for scalability. The proposed solution is designed to satisfy the client's needs for secure user experiences, seamless integration, and robust performance.

## [Design Constraints](#_2et92p0)

Cross-platform compatibility, restricted hardware capabilities, and the preservation of a distinctive naming mechanism are among the design constraints. Furthermore, the application must function within a web-based distribution system, necessitating reliable server-client communication and data synchronization.

## [Domain Model](#_8h2ehzxfam4o)

The relationships between the Game, Team, and Player classes are illustrated in the UML class diagram. The Game class aggregates Teams, while the Team class aggregates Players. The Entity class functions as a superclass. The data structure is optimized, and reusability is guaranteed by object-oriented principles, including inheritance and aggregation.

**"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 systems are distinguished by their secure environment and user-friendly interface. They are suitable for developers with established resources, as they offer a stable server configuration, but they are accompanied by higher hardware and licensing costs. | Linux offers unparalleled cost-effectiveness and adaptability for server-side development. Developers with sophisticated command-line skills will find it to be an ideal choice due to its extensive customization capabilities and open-source nature. | The robust support for enterprise-level applications is the primary reason why Windows servers are so extensively used. In comparison to Linux, they have relatively higher licensing fees, but they also offer an intuitive GUI and integration with Microsoft tools. | Although mobile devices are not well-suited for traditional server roles, cloud-based solutions enable the offloading of server-side operations. Nevertheless, there is a substantial disparity in connectivity and efficacy among different devices. |
| **Client Side** | Frequently, Apple's development guidelines must be followed when developing client-side applications for Mac. Mac is a premium choice for end consumers due to its high-quality user interface and consistent quality, despite the potential cost of development. | Linux clients are highly customizable and cost-effective; however, the development process can be time-consuming and necessitate advanced skills. The open-source ecosystem provides a plethora of instruments that facilitate adaptability. | Windows is a highly adaptable platform for client-side development, as it supports a variety of frameworks and has a large user base. The cost is moderate, and the development time can be reduced as a result of the extensive resources. | Specialized frameworks, such as React Native or Flutter, are necessary for cross-platform development on mobile devices. Although the costs of development are subject to change, mobile applications provide unparalleled accessibility for users who are constantly on the move. |
| **Development Tools** | Visual Studio, PyCharm, and Xcode are frequently employed. Swift, Objective-C, and JavaScript are the most commonly used languages in Mac development, which guarantees both performance and compatibility. | The most frequently used tools are Visual Studio Code, IntelliJ IDEA, and Eclipse. Linux development is optimal for full-stack and backend applications due to the availability of supported languages such as PHP, Java, and Python. | Development programs that are frequently employed include Eclipse, PyCharm, and Microsoft Visual Studio. Windows provides a comprehensive development environment that supports languages such as JavaScript, C++, and C#. | Xcode and Android Studio are the principal tools for mobile development. They enable the development of mobile applications that are highly efficient, as they support languages such as Swift, Kotlin, and Java. |

## 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:** I recommend using Windows as the operating platform due to its adaptability, comprehensive support for development tools, and compatibility with a variety of software environments. Its widespread use guarantees that consumers and developers alike will have access to the resources and support they require throughout the process.
2. **Operating Systems Architectures:** The Windows operating system offers a comprehensive support system for application development and deployment, as well as a well-organized architecture. Its API ecosystem facilitates customization and scalability, while its graphical interface streamlines operations.
3. **Storage Management:** The Windows operating system offers a comprehensive support system for application development and deployment, as well as a well-organized architecture. Its API ecosystem facilitates customization and scalability, while its graphical interface streamlines operations.
4. **Memory Management:** Advanced memory management techniques, including virtual memory allocation and paging, are implemented by the Windows platform to guarantee optimal performance for resource-intensive applications, such as "Draw It or Lose It."
5. **Distributed Systems and Networks:** To facilitate seamless communication among participants on various platforms, it is advisable to implement a distributed architecture. The game can ensure consistency and reliability across devices by utilizing Windows Server's support for distributed databases and networking protocols.
6. **Security:** Windows Defender is a built-in application that is designed to protect against malware, and security is a top priority. Furthermore, the implementation of third-party security protocols (e.g., SSL/TLS encryption) will guarantee safe data transmission between servers and consumers.