

**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 | 11/11/2023 | Ahnaf Zaman | Updates for Project One Submission |
| 2.0 | 11/26/2023 | Ahnaf Zaman | Updates for Project Two Submission |
| 3.0 | 12/09/2023 | Ahnaf Zaman | Updates for Project Three Submission |

**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 Game Room is looking to develop a game(web-based) called “Draw It or Lose It”. The Game is currently available on Android and the company is looking to expand its availability to serve on multiple platforms. The game involves, multiple teams consisting of several members who will guess the puzzle (a phrase, title or thing) based on clues in the form of images that the application will render from a library of stock images. The game will happen in 4 rounds with each round taking up a minute. The drawings are rendered in a steady pace for 30 seconds. Upon time completion for the current team, other teams will get an opportunity to guess the puzzle within 15 seconds.

## Requirements

* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

## [Design Constraints](#_2et92p0)

The application must be designed in a way that would provide multi-user support allowing a team structure. This creates a need for the application to be able to manage multiple instances allowing the different teams to compete. The application also must be able to store game related information including drawing information, time counters, round counts as well as score tabs. Dealing with all this information would require database interaction to successfully store, edit and retrieve the information.

Technical constraints would include the programming language that can be used to program the application. In this case, Java will be the primary choice as it has multi-platform support, available frameworks and libraries for game development and access to various documentations. The application would utilize a client-server architecture.

Utilizing a client-server architecture for smooth interaction and communication throughout the web application, using Java as the programming language due to its simplicity, additional tools/resources and support and using a database to store information would allow us to address all requirements and constraints.

## [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 different boxes represent classes in the project and we can see that Entity is the base class which is inherited by three other classes- Game, Team and Player. This allows all three classes to use the attributes and properties from the parent/super class which means they will all share common references that we see throughout like “name” and “id”. In the diagram, we can see the methods and variables defined for each of the classes. We can also see aggregation where an instance of one class has a reference to an instance of another class- Game is referenced in GameService, Team in Games and Player in Team. Besides the above mentioned classes we also see a ProgramDriver and SingletonTester that are used to run the application and test singleton instance respectively.

**"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 is not typically used for hosting web applications on a large scale, but it can be done.  -It offers a server-based deployment method where the website can be hosted using the built-in Apache server or third-party software like MAMP.  -Mac OS is proprietary software, so there may be licensing costs for the server operating system. | -Linux is a popular choice for hosting web applications due to its stability and security features.  -It offers a server-based deployment method where the website can be hosted using web servers like Apache or Nginx.  -Linux is open-source software, so there are no licensing costs for the server operating system. | -Windows is a popular choice for hosting web applications due to its ease of use and familiarity.  -It offers a server-based deployment method where the website can be hosted using web servers like IIS or Apache.  -Windows OS is proprietary software, so there may be licensing costs for the server operating system. | Both iOS and Android offer native development frameworks for building mobile applications.  -These frameworks require knowledge of different programming languages (Swift/Objective-C for iOS, Java/Kotlin for Android) and development tools (XCode for iOS, Android Studio for Android).  -The mobile application can communicate with the back-end web application running on the server using web APIs. |
| **Client Side** | XCode accelerates Apple platform development, making it effective for teams aiming for macOS and iOS. However, it might demand Mac hardware from developers, possibly calling for specialized staff or tools.  The Mac App Store offers XCode for no charge, which lowers the cost of licensing the development environment. Costs associated with Mac hardware, though, can have a big impact. | Linux offers a variety of development tool options that provide development teams the freedom to choose the ones that best meet their requirements. Linux development, however, may take additional time for setting and customizing development environments as well as knowledge of open-source technologies.  Most Linux development tools are open source and cost-free, which lowers licensing expenses. However, if necessary, there can be fees for support or enterprise-level tools. | A user-friendly development environment is offered by Windows, especially for Windows-centric programs. Unless targeting numerous platforms, several development teams might not be required. There might be a need for compatibility testing on non-Windows platforms.  There are several editions of Visual Studio, including the commercial Professional and Enterprise editions in addition to the free Community edition. Depending on the edition and team size, licensing prices can change. | Platform-specific teams are necessary owing to language and tool restrictions. Cross-platform frameworks could eliminate the need for different teams, but they also necessitate an understanding of platform-specific quirks.  Downloading XCode and Android Studio is free. When developing mobile apps, using third-party tools or cloud services may result in licensing fees. |
| **Development Tools** | For Desktop Applications:  Development Tool/IDE -XCode  Programming Language - Swift (or Objective C)  For Web Applications:  HTML, CSS, JAVASCRIPT  Or frameworks like REACT, ANGULAR JS | For Desktop Applications:  Development Tool/IDE - Eclipse is the most popular  Programming Language - C is the default one  For Web Applications:  HTML, CSS, JAVASCRIPT  Or frameworks like REACT, ANGULAR JS | For Desktop Applications:  Development Tool/IDE - Visual Studio  Programming Language - Visual Basic (default)  It is very hard to develop network-oriented apps with visual basic. But still possible.  For Web Applications:  HTML, CSS, JAVASCRIPT  Or frameworks like REACT, ANGULAR JS | Development Environment - Android Studio (default), XCode  Programming Languages - Java, Kotlin, Swift |

## 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**:

For a gaming room, a high-performance platform like PC would be the best choice. PCs offer the best graphics and performance for games, especially when equipped with powerful processors and high-end graphics cards. They also offer the flexibility to upgrade components as needed.

1. **Operating Systems Architectures**:

A 64-bit operating system architecture would be recommended for a gaming room. This is because 64-bit systems can handle large amounts of random-access memory (RAM) more effectively than a 32-bit system, a crucial factor for running games smoothly.

1. **Storage Management**:

For storage, a combination of Solid-State Drives (SSD) and Hard Disk Drives (HDD) would be ideal. SSDs offer faster load times for games and applications, while HDDs provide large storage capacity at a lower cost.

1. **Memory Management**:

A gaming room should have a system with high RAM capacity. More RAM allows for smoother gameplay and better multitasking. A minimum of 8GB RAM is recommended, but 16GB or more would be ideal for high-end games.

1. **Distributed Systems and Networks**:

A robust and high-speed network infrastructure is crucial for a gaming room. This can be achieved through a combination of wired and wireless networks. Wired networks offer faster and more stable connections, while wireless networks provide flexibility and mobility.

1. **Security**:

Security is a critical aspect of any system. For a gaming room, this could involve using antivirus software to protect against malware, setting up firewalls to prevent unauthorized access, and regularly updating all software to patch any security vulnerabilities. Additionally, secure login systems and data encryption can help protect user data.