

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 | <03/16/2023> | <Hadi Abdelaal> | Initial Release including the game design constraints. |

## [Executive Summary](#_sbfa50wo7nsh)

Based on their existing game, Draw It or Lose It, The Gaming Room plans to create a web-based game that can be accessible by several other devices. The idea behind the game is similar to that of a well-known game show in which players are asked a series of questions.

## Requirements

*These are the requirement needed:*

*A multi-team, multi-player game is required. The game must function on all supported platforms. Validate input using names of active players and teams. The game may only be played in a single instance at any given moment.*

## [Design Constraints](#_2et92p0)

The Game Room explicitly wants its game to be playable on iOS and Android devices. The existing code base may have certain problems as a result, which must be assessed in light of the needs of the various applicable platforms. If the demands increase, there may be a need for more employees and development time.

## [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 Game, Player, and Team classes all derive from the Entity class, which is a superclass. The Game class, the Team class, and the Player class are all referenced in the gameService file. The Game class also refers the Player class. ProgramDriver employs the Singleton class and houses the program execution. Combined, the SingletonTester and ProgramTester classes are used to implement the required criterion for many teams or players and one game permitted at a moment.

**"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** | One of the best terminal commands that can be used for any server configuration, it also has many more software’s than other platforms. | Linux is well known for its cost-effective benefits, also it has many different web hosting options for big websites and ecommerce sites. It also shares with mac a good terminal commands for server configuration. | Windows is well known to be used by most web hosting services. It also has the ability to run a lot of software’s competed to other operating systems. Can be used for high traffic game that need a good hosting. | Very hard to be used as a server. It can be highly compatible and cost effective too like the others . Security is big issue for mobile devices. And a specific hardware is required most of the time. |
| **Client Side** | Usually, the same cost as windows for the development part. It needs more time and experience for development on the MAC side. | Linux is famous for being open-source OS, and it relatively low cost to develop for the client side. The time would be dependent if there is an experienced Linux dev team or not. | Windows usually require less time and experience for the dev part which is due to the market share and the popularity that Windows have. Cost would be usually the same as MAC or Linux. | This is the lowest cost platform compared to others, this is because it a simple OS compared to windows or Linux for example. The issue here is more expertise and time usually needed more than Windows or Linux development. |
| **Development Tools** | Swift is more popular now which is used to code/write Mac applications. You code use Atom, visual studio code as a developmental tool. | Atom and Eclipse are usually used to develop programs for Linux. | Atom and Visual studio and JetBrains product are well known tool to write windows applications. | Swift or Eclipse or android studio are common to be used to develop mobile application. Swift is for the iOS, which android studio being more specific to the android phones. |

**These are the major dev tools that can be used for many operating platforms:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **TOOL** | **DESCRIPTION** | **LINUX** | **MAC OS** | **WINDOWS** | | Atom | open source, fast and flexible with its amazing support for many projects in the Windows platform. | YES | YES | YES | | Delphi | Easy to use, cross-platform with single GUI. | YES | YES | YES | | Cloud 9 | supports many programming languages such as JavaScript, Ruby, PHP, C++, Python. | YES | YES | YES | | Embold | work as Find and Fix with many useful plugins that can help fixing issues with code. | YES | YES | YES | | Kwatee | Used for the Agile approach that can be used for the deployment in many servers. | YES | YES | YES | | Axure | Used for it amazing job when used by many teams that can work together, comes from Microsoft. | NO | NO | YES | | Jira | Used for scrum and sprint planning. | YES | NO | YES | | Azure | The famous azure that used to manage cloud application from Microsoft. | YES | NO | YES | | Eclipse | One of the main benefits of eclipse is that is free and it’s also open source. | YES | YES | YES | |

## Recommendations

1. **Operating Platform**: Because of its general low cost and high market saturation, I would suggest the Windows platform to the Game Room. Future prospects for cooperation in this setting are abundant, and it would be simple to adapt it for mobile devices. Choosing a platform with a sizable developer community may also be advantageous since it promotes cooperation, knowledge exchange, and easy access to a variety of resources.
2. **Operating Systems Architectures**: All Windows-based programs rely on the services provided by Windows, which let them access system resources, display a Graphical User Interface, and much more. Furthermore, communications, online services, and graphics and multimedia are all included in these programs. A user account or a dedicated server can be used to receive such applications. While microservice design is ideal for games that need scale and adaptability, client-server architecture is suitable for games that demand immediate communication. On the other hand, games that need effective memory management and powerful processing should use event-based design. Therefore, it is crucial to select an architecture that adheres to the needs and objectives of the game.
3. **Storage Management**: With Windows 10, managing storage is simple because to the user's ability to control files on the hard drive. The GUI makes this hierarchy's navigation and viewing straightforward. To meet the expanding user base, it's also critical to pick a dependable, safe, and flexible storage management solution. For the game to function smoothly and error-free, memory management is also crucial. Memory leaks and other issues with performance that may have an impact on game quality can be decreased by selecting an architecture that enables effective memory management.
4. **Memory Management**:

A large picture-filled database or library will need to be built by The Gaming Room. Memory allocation makes storage simple. This enables you to retain your entire project on your computer in a more secure location. Memory allocation makes storage simple. This enables you to retain your entire project on your computer in a more secure location.

Also, developers may use the game engine to assist offer access to several libraries that have already been established.

1. **Distributed Systems and Networks**: Cross-platform capability and potential teamwork for a mobile game will be made feasible by the Development . This enables the game to be played simultaneously on several platforms using a single instance of the game. It is vital to create a distributed system that can handle game congestion and provide an uninterrupted connection between many users because games will be played on a variety of platforms and devices. A game may manage a high number of people and function smoothly by selecting an architecture that allows load balancing, activated scaling, and tolerance for faults.
2. **Security**: Windows is the best platform for data security because it includes built-in virus protection and a healthy market for extra security products. This cross-platform capability would also benefit from this database security. Making ensuring the game is secure against cyberattacks, data breaches, and other dangers to security is crucial. Game privacy and user data safety may be achieved by using secure coding techniques, data encryption, and user authentication architecture. Additionally, it is advised to run routine security audits and testing to find and address any potential flaws in the game's architecture.