

Draw It or Lose It

# **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 | 05/22/2024 | Chanelle Harris | Initiate Software Design Template, by updating the executive summary, design constraints, and the domain model to reflect changes at the clients request. |
| 2.0 | 06/09/2024 | Chanelle Harris | Updating Software Design Template, by updating Evaluate Operating Platforms |
| 3.0 | 06/20/2024 | Chanelle Harris | Updating Software Design Template, by updating Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room has developed a web-based version of their popular Android app, Draw It or Lose It. The new version will be compatible with a variety of platforms. This document outlines the constraints that may impact product delivery and provides solutions to ensure that the app can reach the widest possible audience, ultimately expanding the company's market presence.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

For The Gaming Room to function smoothly, the developers at CTS must follow certain design restrictions. Meeting the client's expectations also includes making the game accessible on various platforms. However, since the code currently only works with Android devices, it needs to be adjusted for other platforms, resulting in additional development time. There may also be challenges with cloud storage during the development of the web application or limitations on the number of games, players, or teams due to server 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 application contains a main driver class responsible for initiating the creation of games, teams, and players. This process is handled by the GameService class, which follows a singleton design pattern to ensure that only one instance of it exists in memory at any given time. To achieve this, the constructor of GameService is set to private, preventing its creation. Instead, users must utilize the getInstance() method to instantiate GameService. Before starting GameService, getInstance() checks if it is already running and only runs if it does not exist in memory. Once GameService is up and running, the driver class can use the addGame() method to add a new game. To avoid creating duplicate Games with similar names, addGame() implements the iterator pattern. The newly created Game object is then stored in the List of games.

**"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)

After considering the clients' needs, please find the outline below. We have provided a detailed breakdown of each of our findings for each requirement listed. The table below evaluates the characteristics, advantages, and weaknesses of Linux, Mac, Windows operating systems, and mobile devices.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Webhosting using Mac's server is a favored choice due to its versatility in meeting various hosting needs. However, some may not prefer it for their web hosting services. The accessibility and configuration of Mac's server are user-friendly, offering both a graphical interface and flexible terminal commands. | Linux, being an open-source platform, boasts additional security measures compared to others. Its cost-effectiveness comes at the price of limited support from users, making navigation a bit challenging. However, its command shell makes configuring servers a breeze. | There is a wider range of software options for this platform, and it also boasts faster loading times and a high level of comfort. Additionally, it is widely considered to be the most secure option available. However, there is one drawback - it tends to be more expensive. On the plus side, this platform offers extensive technical support and features user-friendly GUIs as well as command prompt capabilities. | Specifications are increasingly gaining popularity among other devices, as they offer portability and cost-effectiveness. However, there may be concerns regarding poor application security and compatibility with certain mobile devices. Additionally, optimizing specs can prove to be challenging, as they vary according to individual users and their current systems. |
| **Client Side** | Users will incur a significant expense and must possess knowledge and higher skills for navigating the operating system. A considerable amount of expertise and time is needed. The cost is comparable to that of Windows. | Linux users must possess data, a high level of expertise, and maximum cost to use the OS. However, with professional expertise and time put into development, the minimum cost can be achieved. | A certain level of proficiency is necessary. Price comparable, if not slightly lower, than Mac. Uses visual basic as the default coding language. While it may be more challenging to use visual basics, it is still feasible. Costs more for users than Linux but boasts a user-friendly interface and minimal skill requirements. | This feature allows both clients and developers to view updates from anywhere, though it may require slightly more effort to implement compared to alternative devices. |
| **Development Tools** | Mac has the capability to run all programming languages, including HTML, CSS, JavaScript, and Xcode Swift. It also offers front-end development support through PyCharm, GitHub, and Visual Studios. Additionally, Xcode Swift is considered the preferred programming language for Mac's development tool or IDE. | Popular programming languages for creating websites include HTML, CSS, and JavaScript. Linux operating systems are also compatible with popular languages such as JavaScript, Ruby, Python, and PHP. In terms of development tools, the widely used Eclipse is a user-friendly option. The default programming language in Linux is C. | Windows provides a more user-friendly experience compared to Linux. It boasts a wide range of programming language support, including popular options like Visual Studio and Eclipse for HTML, CSS, and JavaScript. In addition, there are also developer tools like PyCharm and command prompt available within this program. | In terms of software and programming languages, Android utilizes Android Studio along with Java and Kotlin, while iOS has Xcode and Swift. Additionally, HTML, CSS, and JavaScript are commonly used for website development. Frontend development is also supported by various IDEs including Python, HTML, PHP, and C++. |

## Recommendations

1. **Operating Platform**: The Gaming Room has set a goal of releasing their Android-exclusive mobile gaming app on multiple operating platforms. This includes making it accessible on Linux, Windows, and Apple iOS devices, in addition to Android. To successfully accomplish this objective, I suggest utilizing a cloud-based operating platform, such as Microsoft Azure. This will guarantee compatibility across various platforms, scalability capabilities, and assist in keeping development and deployment expenses manageable.
2. **Operating Systems Architectures**: The Gaming Room can utilize virtual machines or container services such as Azure Kubernetes Service (AKS) for cross-platform development. Azure by Microsoft is not your typical operating system. Instead, it is a cloud platform that provides an array of services using diverse operating system architectures. Additionally, they can access the "Draw It or Lose It" application through a web-based interface.
3. **Storage Management**: With a library of 200 high-definition images, each approximately 8 MB in size, the Gaming Room is committed to enhancing the "Draw It or Lose It" user experience and preventing image repetition. To achieve this, we turn to Microsoft Azure, which offers two solutions for our expansion needs: Azure Blob Storage for scalable image storage and Azure SQL Database for managing image metadata, game data, and player information.
4. **Memory Management**: The memory management methods used by Microsoft Azure resemble those used in operating systems, such as assigning and compressing memory, swapping data, and overseeing its use. To run "Draw It or Lose It" smoothly on both Linux and Windows environments in The Gaming Room, Azure Virtual Machines will be allocated and controlled dynamically to maximize memory efficiency.
5. **Distributed Systems and Networks**: Among the array of services provided by Microsoft Azure is Azure App Service, a tool designed to support the development and deployment of microservices within a distributed system. Through this architecture, application components like microservices can effortlessly communicate and share data regardless of the platform they are deployed on. Additionally, Microsoft's robust network infrastructure boasts built-in redundancy measures that effectively mitigate the effects of network outages, thus ensuring minimal disruption for The Gaming Room's valued customers.
6. **Security**: Ensuring the protection of user information and The Gaming Room data is our utmost priority. With Microsoft Azure's comprehensive security measures, we address a range of potential concerns with a multi-layered approach. Entra ID, integrated within Azure, offers authentication and authorization for users, applications, and resources. Additionally, our Multifactor-Authentication (MFA) strengthens this already secure position, guaranteeing safe access to "Draw It or Lose It."