

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 | 07/12/2023 | Nii Amatey Tagoe | Description of Executive summary, Requirements, Design Constraints, and Domain model. |

## [Executive Summary](#_sbfa50wo7nsh)

*The Gaming Room is a gaming company with an Android app-only game called Draw It or Lose It to develop a web-based game like Draw It or Lose It that serves multiple platforms. The Gaming Room staff does not have the expertise to develop a web-based version of Draw It or Lose It and the solution to this problem is to create a software application capable of operating on multiple operating platforms.*

## Requirements

*The Gaming Room’s main business requirement is for their game Draw It or Lose It to be available in an Android app and a web-based game capable of operating on multiple operating platforms. The technical requirement is creating a game with the capability of having one or more teams with each team having multiple players assigned to it, the game and team names must be unique to allow users to check whether a name is in use when choosing a team name, and 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 main technical design constraint in developing the game application in a web-based distributed environment is that the game must run on all operating platforms. It will not be an easy task to create a gaming software application that can run on multiple operating systems on different devices. Also, there must be only one instance of the game in memory at a time. Another technical constraint is choosing the right operating system for the development of the gaming app and being sure that the chosen operating system will not produce any compatibility issues in the process. It will take time and effort just to decide which IDE, text editors, and programming language that will be suitable for developing the gaming application, bearing in mind that the application must run on multiple platforms. One common business constraint is the possibility of the high cost of building the application and the number of personnel needed to accomplish the task at hand.*

## [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 UML class diagram provided below depicts the structure of the code and how each class functions and relates to one another. The super class or base class is the Entity Class as shown below with its child classes being the Game class, Team class, and Player class. The Game, Team, and Player classes inherit methods from the Entity Class. The ProgramDriver class has the main method and the SingletonTester class's main purpose is to test a singleton behavior. The GameService class has a singleton service for the game engine, an example of the functions of the GameService class is to give the software application the ability to have only one instance of the game exist in memory, that is generating unique identifiers for each instance of a game, team, or player.

**"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 paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Some well-known characteristics or feature of the MacOS is its well-designed accessibility features. MacOS has a more friendly and less complex user interface which makes server configuration done with ease. The MacOS server which contains Apache can be used to host web-based gaming applications.  The MacOS provides easy integration within the Apple ecosystem, but the downside is compatibility with other ecosystems The downside of using the MacOS is that there is a high cost in terms of purchasing a licensing for every MacOS server instance deployed. | The Linux operating system on the other hand has a less complex user interface in terms of navigation and configuration of the server for hosting a web-based software application but the command-line interface is very tedious to navigate. Unlike the MacOS and Windows operating systems, it is cost-effective because the Ubuntu server is no licensing fees that is it’s an open-source operating system. The Linux operating system is the most suitable option for hosting a web-based application. Linux mostly makes use of Nginx and Apache HTTP has its web hosting. | The Windows operating system is more like the MacOS and even slightly better because of its compatibility with numerous software. Windows also provides good security, but the possible licensing costs to the client, The Gaming Room, for the server operating system is high when compared to MacOS and Linux. Hosting on Windows servers is normally achievable with the use of running Node.js applications or Internet Information Services (IIS) via Windows Server. | Mobile devices have mostly weaknesses rather than strengths in terms of the configuration and navigation of the server for software applications on mobile devices. The server of a web-based software applications can be configured remotely on a device mobile through a browser on the devices but it is not ideal. |
| **Client Side** | In terms of software development considerations such as cost for instance, the MacOS has a high cost when it comes to supporting multiple types of clients. The time needed for software development is above average and the expertise required is medium. One requirement in the application development process for a web-based software is to setup a front-end and back-end web application and RESTful APIs can be implemented to enable communication between the back-end and the front-end. Using JavaScript, HTML, and cross-browser CSS. The application must be compatible with the safari browser. | The Linux operating system on the other hand incurs less cost but more time is needed in software development. In terms of expertise in software development, maximum expertise is needed. Also, a requirement in the application development process for a web-based software is to setup a front-end and back-end web application and RESTful APIs can be implemented to enable communication between the back-end and the front-end. Developers can use JavaScript, HTML, and cross-browser CSS. The application must support multiple browser such as chrome, Firefox, Microsoft Edge, and others. | Again, the Windows operating system is similar to the Linux operating system, but it is not cost-effective, and the same level of expertise and time is needed because the application must be compatible with all web browser platforms and mobile devices. Similarly, a requirement in the application development process for a web-based software is to setup a front-end and back-end web application and RESTful APIs can be implemented to enable communication between the back-end and the front-end. Again, the application must run on browser that are compatible with the windows operating system. | Software development considerations that are necessary for supporting multiple types of clients as they pertain to Mobile Devices is very similar to the regular three operating systems. The duration to create a web-based software application for iOS and Android is shorter meaning it has less cost and minimum expertise is needed. |
| **Development Tools** | Visual Studio code is one of the best and most popular IDEs available on MacOS for web-based software applications. Visual studio code offers a range of tools for JavaScript development which is one of the programming languages relevant for building a web-based software application. Other languages include HTML and CSS and frameworks such as Angular and React. IntelliJ IDEA is also preferred. Multiple development teams are needed in other to finish a deployable application and one impact is figuring out which language will be best, and tools needed to ensure effective communication. | Visual Studio code on a Linux operating system is an ideal IDE to use while developing a web-based software application. Again, JavaScript, PHP, CSS, C#, Java, and Python are all preferred programming languages. | The Windows operating system is the most compatible operating system, Visual Studio code, Eclipse and Sublime Text 3 are development tools, IDEs and text editors that can be used to develop a web-based software application with ease. Visual studio code for instance has tools available that uses AI to learn from the edits made while writing code to finish the lines of code. Again, the programming includes Python, JavaScript, HTML, Java, and C++. | <HTML, CSS, and JavaScript, Swift or Objective-C for iOS and Java for Android are programming languages used to build web-based software applications for deployment on mobile devices. IDEs and text editors include Visual studio code and Sublime Text 3. |

## 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**: The appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments is the Windows operating system. The Windows operating system currently holds more than 87% of the existing desktop or laptop OS market. This fact is an advantage for the Windows OS on the client side. On the server side, one of the best cross-platform app development frameworks that is Node.js is available on the Windows OS, and again, hosting on Windows servers is normally achievable with the use of running Node.js applications or Internet Information Services (IIS) via Windows Server. The most notable downside is the cost of licensing. Windows OS has multiple development tools that can be used to develop a cross-platform web-based application and an example is the use of programming languages such as CSS, HTML5, and JavaScript. I believe Windows Operating is the best operating system suitable for The Gaming Room to magnify Draw It or Lose as a web-based application for multiple computing environments.
2. **Operating Systems Architectures**: Generally, the system architecture of the Windows operating system is centered on a structured design that comprises of numerous subsystems and sections, these sections include file systems such as NTFS (New Technology File System), device drivers, the Windows Kernel, user mode, registry, Win32 API, a networking stack featuring DNS and DHCP. The system architecture also includes a Graphical user interface. In terms of the operating platform architecture or pattern, I would recommend a three-tier client-server architecture. This client-server architecture features a first tier which is the client, and it is in communication with two or more servers which is also in direct communication with a database. The use of a three-tier client-server architecture will bring about scalability, high performance, improve security, and ease maintenance to the Draw It or Lose as a cross-platform web-based application.
3. **Storage Management**: In terms of storage management is recommended the use an SSD (Solid State Drive). The Windows operating system architecture features an optimized drives utility which is responsible for maintaining the high performance of storage devices such as an SSD by optimizing the arrangement of data on the storage device.
4. **Memory Management**: The Windows operating system uses multiple concepts and mechanisms to ensure efficient memory management. For instance, Windows OS uses the concept of Virtual Memory. Virtual memory basically permits the processors to use more memory than physically available Random Access Memory. Windows OS provides memory management application programming which also allocates, protects, and prevents memory leaks efficiently.
5. **Distributed Systems and Networks**: The use of RESTful APIs will be relevant at this point. A RESTful API is ideal for communication between various platforms. Windows OS is compatible with RESTful APIs. The main purpose is a RESTful API is to enable communication between different software devices over a network using the HTTP (Hypertext Transfer Protocol). A REST API can be accomplished following the six guiding principles of the RESTful architecture which later results in well-distributed software and the network that connects to multiple devices. A RESTful API must have a uniform interface that is the API must allow for the unique identification of resources and provide the necessary information requested by the servers. RESTful API must be layered systematically, stateless, and cacheable.
6. **Security**: The Windows OS offers security such as User Privileges, Firewall, and Anti-Malware also known as Windows Defender. On the client side, we can use make of multi-factor authentication or strong passwords in order to counteract security attacks. We can also educate users on how to prevent and avoid phishing attacks. We must also encourage users to install any software updates deployed. On the server side, we can perform server hardening that is Disable unnecessary roles, services, and features to minimize security attacks, encryption, strong authentication, firewall, and network segmentation, and enable security auditing which is tracking and recording every log system event and login attempt.