

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 10/19/22 | Keerut Sodhi | The Gaming Room wants a web-based game developed that can be played on multiple platforms. |

**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 Gaming Room has a mobile application available to Android users called ‘Draw It or Lose It’. The company has requested to develop a web-based game, based off the current game, that can serve multiple platforms. The game will consist multiple game options with the need of one or more teams. Each team created will have the ability to assign multiple players to it. To ensure that only one game instance exists in memory at any given time, each instance of a game, team, and player needs to be unique. When the user inputs a team name, the game will check to see if that name is already in use. ‘Draw It or Lose It’ contains a large library of stock drawings. The staff members of The Gaming Room are unable to execute the setup of the environment.

## [Design Constraints](#_2et92p0)

* This project requires three different platfroms to be served. To perform on each platform, we will need the appropriate software, hardware, and operating system for the game. Using a cloud based hardware is a solution for multiple platform usage.
* The current security for the Android app will not be sufficient for the other platforms. We will need a higher security structure to secure client-server communication and information.
* The game is currently an Android app. To make it available to multiple platforms, the Android app will need to be translated to a web app to make it accessible on other operating systems. Instead of choosing a compatible programming language across the platforms, the use of REST API can be used to communicate this via HTTP.
* To support one or more teams with multiple players, the use of a client-server architecture will need to be implemented to ensure that the server is able to handle multiple players at one time.
* Unique IDs to games, teams, and players will need to be implemented to ensure game and team names are unique and only one game instance exists in memory at any given time.
* Large library of images will be used for the game, requiring more storage and ample memory to allocate these images. Utilizing the Cloud for storage and virtual memory will ensure efficiency and faster performance of the game.

## [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)

From the UML diagram displayed below, the ProgramDriver class contains the main method. The ProgramDriver uses Directed Association with the SingletonTester to test if an instance of GameService already exists. The Entity class is the parent class to the subclasses Game, Team, and Player. These three classes all inherit the Entity’s information about each player in the game. Player class extends Entity to hold information of each player in the game and output that information. Player class has a relationship with the Team class, where each team will have multiple players. Each Team holds a list of Players and can add another player. Game class has a relationship with the Team class, where the Game class holds a list of Teams and can add a team. This relationship verifies that each team name is unique and shows the user if their chosen team name is already in use. With the inheritance of Entity to the three subclasses, this satisfies the clients requirements of each team having multiple players, a game having one or more teams, team names to be unique, and allowing users to check whether their requested name is already in use.

**"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** | MacOS can be used as a server but requires licensing, which is expensive.  You must have a Mac to develop, whether it is a MacBook or iMac.  Web hosting and flexible terminal commands are some of the many characteristics of Mac.  The advantages of Mac include good upgradable equipment, has different options for web browsing, and not easily susceptible to viruses.  The disadvantage of Mac is that it’s not most preferred for web browsing. | For a web-based software application, Linux is well equipped. It is the most popular of the three and the licensing is no cost since Linux is open source.  Characteristics include being more secured and most preferred operating system.  The advantages of Linux include being the preferred web server, the security is much better, an most issues are caught before becoming an issue.  The disadvantage of Linux is that it is difficult to search and use application. | Windows has a server that is very secure and easy to operate and setup. However, the licensing is expensive.  Characteristics of Linux is that it is more dominant compared to the other platforms.  The advantages include less loading time and higher comfort when using,  The disadvantages are that it is easily susceptible to viruses and require extra anti-virus protection than what is already built in. | Though mobile devices can be used as servers, they are not equipped to excel at it. Mobile devices the power for high end but can be used for development.  Characteristics include being portable and easy to use.  The advantages are better compatibility and more cost efficient.  The disadvantage is low security. |
| **Client Side** | Variety of web browsers are supported, such as Safari, Google Chrome, and Firefox.  Mac is the easiest to use for most and the SDKs are great and easy to use.  Time would be minimal as well, but the costs are higher.  A Mac device would be needed as well as someone who is experienced using Swift on Xcode.  macOS has cross browser testing software. | Variety of web browsers are supported, such as Google Chrome and Firefox. Works with every web browser since Linux is open source.  The development time would be the highest and the cost would be less.  You must have someone who is experienced with Python on Linux. | Variety of web browsers are supported, such as Google Chrome and Firefox.  The highest requirement for Windows is the expertise, which can increase project costs depending on how many team members are needed.  Using .NET framework for security and capability would be the most recommended.  Windows can perform cross platform testing. | For Mobile Devices, you will need to find developers who have experience in app development. User interaction and display design needs to be handled differently than on the web. It does provide more flexibility to clients and developers by being accessible anywhere and anytime.  With mobile devices being easily accessible to users, most people will use a mobile device rather than a computer. |
| **Development Tools** | Macs have an app developer called Xcode. Xcode IDE is combined with the Swift programming, in which all coding is done with Swift. | Python is preinstalled on most Linux distributions. A user can use IntelliJ’s Ultimate IDE to code for this. | Visual Studio Code is the best software to code Windows applications. Any programming language can be used with this, but most Windows programs are written in C++. | There are three options that can be used to develop an app for Mobile Devices. For Androids, a specialist in Android Studio will be needed to develop that app. For iPhones, a specialist in Xcode using swift will be needed as well as a Mac device, such as MacBook or iMac. The app can also be developed using Unity, which is a C++ programming language that can be converted to either an Android app or/and iPhone app. However, you will still need a Mac in order to convert it to an iPhone app. |

## Recommendations

1. **Operating Platform**: Based on our evaluations of this project, we recommend using the Windows Operating System. Windows has more software’s available at the developers dispense and many IDEs to utilize.
2. **Operating Systems Architectures**: Windows provides services that can be utilized by everyone. Many applications are offered that show Graphical User Interface (GUI) while still accessing system resources. Windows operating system separates into two modes, User mode and Kernel mode. User mode processes are user-faced and influences what the user interacts with. Kernel mode deals with the inputs and outputs, memory management, storage management, and networking. To hold data, Windows uses a directory structure. With Windows supporting multiprocessing of the hardware, it allows the customization of the system.
3. **Storage Management**: The current version of Windows, Windows 10, is supplied with a feature called Storage Sense. This feature allows the user to manage files on the hard drive and analyze how much space it will occupy. You can also utilize an external storage, like the Cloud, to save data. Using the Cloud for storage would allow remote accessibility, enable higher security, and enable faster performance. The built-in storage system allows easy file creation and placement for larger size projects.
4. **Memory Management**: While developing this game, a database or library with space for images will need to be created. The memory allocation allows easy storage of images outside the default image folder on Windows. This will allow you to keep and manage the entire project on one device rather than using an external device. Disc paging and demand paging acts as an extension of the computer’s physical memory and RAM memory, allowing faster and efficient loading from memory. Disc paging reserves part of the hard disk as extra RAM and demand paging separates processes into smaller tasks, loading into memory only when required for instant processing. There are two forms of memory, physical memory and virtual memory. With Windows utilizing virtual memory, it would allow the program to execute efficiently and faster.
5. **Distributed Systems and Networks**: In order to support multiple clients, an API needs to be implemented for clients. REST API exchanges communication between the server and the client while using HTTP to access and use data. The server and client are set up independently, allowing both to understand any messages received. Since REST API uses less bandwidth, making internet usage more efficient, it would be the best client-server solution. RESTful API can be built with JavaScript, which will be used as the programming language for this project. The company must ensure that its servers are powerful enough to support a large player volume and backup power for power outages. This will help prevent problems such as outages and connectivity issues.
6. **Security**: Windows comes with a built-in security protection software called Windows Security. However, Windows is more susceptible to viruses, so it is recommended to install an anti-viral software from another source to ensure user data security. The system scans for any malware viruses and security threats. These scans happen in real time which allows you to be notified immediately of any detected threats. When using a form of encrypted communication called JSON for REST API, it adds extra security to the communication.