

Draw It or Lose It!

# **CS 230 Project Software Design Template**

Version 4.0

## 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 |
| --- | --- | --- | --- |
| 4.0 | 10/22/22 | Trayven Nixon | Updated Executive Summary, updated Design Constraints, Updated System Architecture View, Updated Domain Model, and added Recommendations. |

## [Executive Summary](#_sbfa50wo7nsh)

Our new client The Gaming Room wants us to develop a web-based game that will be available on multiple platforms. This game will be based on their current game, Draw It or Lose It, which is currently only available for Android currently. The best option for us to pursue would to be use a distributed application architecture so that multipole players on platforms other than android can be authenticated and connect to a centrally located game using the server client architecture. Then the server can be managed securely.

**Requirements**

**[REQ\_1]** A game will have the ability to have one or more teams involved.

**[REQ\_2]** Each team will have multiple players assigned to it.

**[REQ\_3]** Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.

**[REQ\_4]** 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 Game Room needs to have a web-based game application. This means that a web application needs to be built and the appropriate hardware server or cloud-based machine must be implemented. This machine will also need an OS that will host the website.
* The server must have large amounts of storage for the image library of drawings.
* This is a quick game so timing is imperative. There must be large amounts of RAM within the server to cache so that the game application can quickly receive the data needed.
* The Game Room will need an Administrative app to be able to allow games, teams, and players to be defined, associated and coordinated to the client-side applications.
* There will need to be some form of login to allow the app to differentiate each player from the rest of the players who are currently playing the game.

## [System Architecture View](#_ilbxbyevv6b6)

Diagram

Description automatically generated

## [Domain Model](#_8h2ehzxfam4o)

There are 7 classes shown in The Gaming Room UML diagram. Class names are: ProgramDriver, SingletonTester, Entity, GameService, Game, Team, and Player. The ProgramDriver class uses t SingletonTester and has the Main Method that executes the application. Game, Team, and Player class show Inheritance from the Entity class. This means that those classes acquire properties from the Entity class i.e. parent/child association. GameService and Game show association, Game and Team show association, and Team and Player also show association. Each of these class association have a none to many composition, meaning that A has o or more instances of B but B cannot survive if A is disposed.

**"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** | OS X Server is a server OS designed specifically for Apple based server machines. Mac systems were developed based on BSD UNIX systems starting in the early 1980’s.    **Advantages:**  General administration is easier than Linux or Windows. It has an easy to navigate GUI. Mac has a lower TCO. It is upgradeable and reliable. There are many hosting options for the Mac OS x such as Bluehost, SiteGround, and WPEngine. A website hosted on a Mac server will work on any operating system. Mac servers are considered extremely safe when it comes to security.  **Disadvantages:**  Server hardware is costly being that is proprietary. Cloud service providers are difficult to find as well as also being high priced compared to windows or Linux.  Unlimited client license price is $999. Enterprise scaling is difficult. Often software packages are not certified for MacOS. Main uses is for Mac clients. | Linux was also based off UNIX systems, developments started in the early 1990’s. Linux is the only OS of the 4 that is open source. Meaning that its source code was made available on the internet for free.  Linux is the most preferred sever side OS. Is the most commonly used server to host websites.  **Advantages:**  Hardware can be less expensive than Windows or Mac. Hardware requirements are also less than Mac or Windows.  Is more secure than windows but less secure than Mac. Less updates means more stable and more up time. Because Apache Linux is open source it is a free OS, these means it will have no license fee.  **Disadvantages:**  Linux has a higher learning curve than Mac or Windows, while Apache is free if you want tech support a RHEL is needed which costs between $349 and $1299 a year.  Lacks software support. | Development for Windows began in the late 1980’s as a propriety OS and is the most widely used OS at this time when considering server and client OS’s. Windows uses Windows VPS for web hosting.  **Advantages:**  Hardware costs are moderate, while it can be more expensive than Linux it is much cheaper option than Mac.  Most people are familiar with Windows. Windows has the most compatibility for software applications. Windows allows the use of Linux as well.  **Disadvantages:**  Regular updates are required, this affects the uptime of the server.  Least secure OS compared to Linux and Mac.  Windows Server 2022 Licensing fees range from $501 to $6,155  With the standard licensing costing $1069. | Mobile Devices are not an option for the server side of the application due to the load the server will need to process. If it was a static website with a small amount of needed resources this would be different. |
| **Client Side** | Most expensive hardware option of the four due to the proprietary hardware and OS. There are less developers familiar with Mac than Windows. Most Developers using Mac us Swift or C++ not Java. | This can be the least costly option if the developer is familiar with Linux. Linux also will require the most time and expertise from a developer due to the complexity of Linux and less GUI and more use of the terminal. Some Developers using Linux will be familiar with Java. | Windows has a moderate cost but much less than Mac. Most developers are familiar with Windows OS so this will require the least amount of time and expertise. Some Developers using Windows will be familiar with Java. | While Mobile devices will be needed for testing, a mobile device would not be a good development platform due to the lack of hardware power and usability. |
| **Development Tools** | Mac does not need a third-party tool to SSH to a server, this is done through the terminal.  Eclipse IDE for developing in Java is available and viable for use on Mac OS. Mac OS does have many tools for visuals within the game and is the choice for graphic designers. While the default browser is Safari, you can also use Firefox and Chrome for testing. Supports GitHub Desktop.  REST Assured can be used within the java code to test and validate REST services. | While a server can be access by using the terminal, the VNC application is a common tool that also access a remote server.  Eclipse IDE for developing in Java is available and viable for use on Linux. Default browser in Linux depends upon the flavor that you use. Most flavors can still use Firefox and chrome for testing. Can access GitHub through command line as well REST Assured can be used within the java code to test and validate REST services. as a browser. | Severs can be accessed using Remote Desktop Connection that comes with Windows, but many people choose to use Putty. Eclipse IDE for developing in Java is available and viable for use on Windows. The default browser is Edge, you can also use Firefox and Chrome for testing. Windows can also run TestComplete for automated testing at an additional cost.  Supports GitHub Desktop. Notepad ++ is available. REST Assured can be used within the java code to test and validate REST services. | Mobile devices will be needed for testing. To test the web-based application, most mobile platforms can use Chrome or Firefox for testing. IOS has a default browser, Safari, but can use other browsers. Android default browser is Chrome but can use other browsers as well. |

## 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**: I recommend the use of the Windows OS for the development of The Gaming Room’s Draw It or Lose It application. Reasoning for this is that most developers are familiar with the Windows OS as well as not as technically inclined management. Windows allows the easy use of the Eclipse IDE to develop the application in Java. For the cloud server that the application and data will be stored on I recommend the use of Apache Linux due it being fast, reliable, and highly secure.
2. **Operating Systems Architectures**: I recommend the use of a server / client architecture using the REST API. This allows for the browser on any device (the client) to send and receive data quickly and efficiently to and from the cloud server. Using a cloud server will allow the gaming room to size the storage and memory on a as needed basis for the user load on the application. To make the cloud service more redundant and reduce the points of failure a redundant three-tier architecture should be used.
3. **Storage Management**: Using a cloud-based server that is using Zonal Balanced Persistent Disks with object storage(cheaper than SSD persistent disks but suitable for most general purpose applications)The Game Room team will be able to size the storage of the server quickly and easily. This also decreases the cost being that you will not be paying for extra storage that is left unused.
4. **Memory Management**: The Machine used should be a memory optimized, this allows for medium to large caches in the memory of the server.
5. **Distributed Systems and Networks**: Since the server provider will be in the cloud of the server goes down a new server can be spun up quickly, sometimes automatically, to make sure that the server up time is as good as it can get. The downside is that the client device will always need to have a connection to the internet, this could be through WIFI or mobile data. If the client does not have WIFI or cell service, the game will be unplayable.
6. **Security**: Using Windows as the development OS and Apache Linux as the Server OS makes managing the security much easier. Because Windows is patch monthly for security threats and Apache Linux is also managed by a large community because it is open source this limits the attach vectors available to threat actors. The server will also be using a firewall to help stop inbound threats. The users of the application will have a log-in page to verify their account, this account information will be stored on the server in a hash form so that not of the user’s information will be in plain text. I also recommend to use of two factor authentication. 2FA add another layer of security because the user will need more than just a password to authenticate themselves.