

<Draw it or lose it>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc110185019)

[Table of Contents 2](#_Toc110185020)

[Document Revision History 2](#_Toc110185021)

[Executive Summary 3](#_Toc110185022)

[Design Constraints 3](#_Toc110185023)

[System Architecture View 3](#_Toc110185024)

[Domain Model 3](#_Toc110185025)

[Evaluation 4](#_Toc110185026)

[Recommendations 7](#_Toc110185027)

Evaluations

Final thoughts 9

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
|  |  |  |  |
| 1.0 | <07/10/2022> | <Raymond Sauter> | <Brief description of changes in this revision> |
| 2.0 | <07/30/2022> | <Raymond Sauter> | <update made to the Evaluations> |
| 3.0 | <08/13/2022 | <Raymond Sauter> | <edits and final thoughts added Really need this for my degree which I get after I complete this I really tried. Thank you for being a great teacher if you see this> |

**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 company The Gaming Room is looking to develop a browser-based game that can run on multiple different operating systems. The game will be called “Draw It or Lose It” and is already available on Android based systems. The game is based on different teams of players given a four-minute time frame to give an answer on what image they are being shown from a randomly chosen picture from a library of various images. If they fail to answer or answer incorrectly the opposing team is give fifteen-seconds on the timer to counter with their own answer, until the timer runs out.

## [Design Constraints](#_2et92p0)

Multiple teams needed consisting of more then one player on each team

Game and team names given need to be unique so they can be checked against a database of previous users

Only one instance of the game can exist at any one time

Must be cross-platform and easily accessible

## [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 Entity class creates a relationship between the Game, team, and player class definitions. This means that they are the child of the Entity parent class. UML diagrams show inheritance, so, each class shares referenced attributes like “name” and “id”. We can see how UML uses aggregation to show relationships, this means that an instance of one glass can have a reference to an instance of another class, we can see this in the diagram with the Game Services class referencing the Games class, Games referencing the Team, and the Team referencing the 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 own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Flexible terminal commands to configure the server, access, or make changes.  Characteristics It is popular in web hosting  Advantages It is upgradeable, it has various options for different web hosting requirements  Disadvantages It is less preferred for web hosting services | The same goes for mac plus more cost-friendly  Characteristics Secured, most preferred.  Advantages Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  Disadvantages It is more difficult to find applications to support the web hosting required needs. | More software available compared to other OS.  Characteristics It is dominant to the other platforms. Close platform  Advantages High resource requirements, less loading time, high comfortability  Disadvantages easy virus susceptibility, poor tech support | It is better if the server is immobile and can be tracked in a single place. Specifications are better in other devices.  Characteristics More popular, high portability.  Advantages Have a wider reach, better compatibility, cost-effective  Disadvantages It is highly selective to various smart mobile devices Poor security |
| **Client Side** | Moderate expertise and time required. expensive like windows. What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices? | Open source difficult to get used to operating | Minimum expertise and time required. priced similarly to mac. | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | When running languages on macs we can run swift the more popular option. While mixing in nice tools like notepad++. Though Macs can run all languages. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Linux can work with visual studio, eclipse, along with notepad++ for a nice and easy-to-use tool. Along with many more languages and tools. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Easier to use than Linux but can run the same as it. So visual studio, eclipse to name a few of the many languages. And with multiple tools notepad++ is a simple to use the tool. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | You can create countless apps using android and swift. Both languages and software can be run on all three machines. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Rub |

## 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 would recommend the Gaming Room starts on Windows devices working with a very portable framework then move into the other platforms keeping the costs to a minimum
2. **Operating Systems Architectures**: windows environments provide services that enable them to display a GUI while accessing system resources among other things like graphic, multimedia, massaging and webservices.
3. **Storage Management**: Windows comes with a feature called storage sense making it easy to free space up on the system and even connect directly to cloud-based exchange systems. The way the window file system works makes for easy file creation on larger more tedious projects.
4. **Memory Management**: The way this game functions will require the company to maintain a large database of images to make the game unique and surprising every play threw this will require a lot of management and the system explorer makes this easy even within and IDE like studio code
5. **Distributed Systems and Networks**: Because each operating system being different I investigated ways to publish the game to run on all dives. I found Develop 4 which enables cross-platform game creation. It’s an IDE that can be run on any device. Once the game is created you can simply export the game file into the web, iOS, Android, and many more options that will allow cross-play. This will make dependencies even easier to manage and prevent frequent connectivity issues
6. **Security**: windows come with some of the most up to date built in security protection software for free, making it a lot more secure. Comes with a firewall out of the box that is easy to use and manipulate, even comes with a malware and spyware scanner which can be devastating to corporate industry

**Evaluations**

* The web application or game is going to be hosted on a distributed environment which means it will be very network intensive and will operate on the application servers where the input is taken in through the client browser, processed at the server and render back at the client’s screen
* This will operate like a web app that shows html response through a browser window, regardless of what operating system that browser is functioning under every platform has separate development tools, APIs, and programming languages.
* First, we can discuss the server-side implementation and then we’ll focus on the client-side Implementation

Does each platform have a server-based deployment method where the web application will be hosted?

What are the potential costs to the client, licensing fees what will this cost The Gaming Room, and how much will the server cost that the web application functions off?

When deciding what server to function off there are two major choices in the industry Linux and Windows.

Linux Server Windows Server

|  |  |
| --- | --- |
| Open Source | Proprietary |
| Secure | High Security |
| Free | Paid |
| Cost Effective | High cost |
| Less support/more knowledgeable | More support/more user friendly |

A Web application is really two apps in one the first runs in some code on the user’s browser the second runs on a server at some remote location and feeds data to all the client terminals. The default languages or “stack” as you will hear it called is the front end or client side which usually consists of HTML5, CSS3, and JavaScript6. They make up the web application in a visual representation, which then needs to relay information back and forth to the server which can be done with PHP or more modern frameworks like React and Angular.JS.

Web applications also make it easier for the developer to feed update into the system without causing mass server outages. The difficulty when it comes to programming web applications is making them super responsive can be difficult since they are relying on a sufficient internet connection to feed the server the information and then for the server to feed it back and then for it to be generated in the browser.

When using windows there can be multiple licensing fees however it is much more user friendly, and you wouldn’t need to spend as much on network engineers to maintain the network. On the other hand, Linux is opensource but can be much more difficult to maintain and operate, some say Linux is less secure as windows however, open-source security applications on Linux outnumber windows security applications. Being opensource the Linux community can be very friendly, and you can find help forums on almost any discussion topic.

Web Servers communicates using HTTP methods between the browser and server to send requests from the browser to server, when you click a link on a web page, submit a form, start a search, or press a button to perform an action the browser sends the HTTP Request to the server. These communications come in the form of a method request which are associated with different actions some of those being FIND, POST, GET, and SEND but there are many more.

**Client-Side**

**1) Desktop Application - Windows**

On windows the most popular IDE is Visual Studio and uses the languages C++ and C# which can require a subscription or a paid for compiler. The most popular browsers are internet explorer, Chrome, and Opera.

**2) Desktop Application - Linux**

On Linux the most popular IDE is Eclipse and is programmed in Java and C. these do not require a subscription or fee but do require a compiler which usually comes pre-installed on the system. The most popular browsers are Mozilla Firefox, Duck-Duck Go, and Brave which offers free crypto to view paid advertisements.

**Final Thoughts**

This game is a great idea and I believe with the right architecture anything in the world of computing is possible. This game would be very system intensive because of the size and number of the library of pictures that we are working with, that is another reason the design elements of a web application make it much less intense on the user’s memory and RAM. It would be ideal to work with a hp server running windows because it is much easier to work with and get customer support when needed. In the server since we want the pictures rendered in a certain amount of time, you would want to be running SSD or even NVME SSDs for storage in the servers for snappy response time. Another important part of the server is the OS architecture, the most popular ones are ARM and X86, which we would want to go with the X86 which is specific to intel and works great with windows. Memory management is another important piece of the puzzle and there are two types of addressing on windows 32-bit which enables addressing up to 4 gigabytes of memory, and then there is 64-bit which allows 8 terabytes of virtual address space. Processes can access this virtual memory, but they cannot access other processes memory to keep them from being corrupted. Network based multiuser interaction systems such as network games typically include a database shared among the players that are physically distributed and interacted with by one another over the network which is how this game will function. We intend to provide a high-level application program interface for the player to interact with one another through so the server will distribute game and player files all over the network therefore we will need a very efficient server or a number of servers to act as the database for the network that will keep expanding over time with new user and pictures being added to the database to keep the users experience new and not repetitive where they receive the same picture a number of times. The security of this server will be a top priority once the game launches because if the database is hit with an attack, then the whole system goes down and no players can even log on to access their player records which could be erased along with all the other files on the system. This is another reason I would go with windows is because of its amazing security features which are much more advances then all the other OSs. Windows running on an HP server X86 architecture would be the best option for the central server which is acting as the heart of the system pumping images through the arteries to all the user consoles.