

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | 05/19/23 | Robert - Lowrey | Initial documentation for The Game Room Project. This will help in the development of a multi-platform web-based game that is currently only available as an Android mobile application exclusive |
| 1.1 | 06/04/2023 | Robert – Lowrey | Second revision consisting of the Evaluation section being completed to discuss what is considered when choosing an operating system to host a we based software application. |
| 1.2 | 06/12/2023 | Robert – Lowrey | Third and final revision on the document for project three. This updated version offers a recommendation to our client The Gaming Room. |

**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 client we are working with is The Gaming Room. The company is wanting their current Android mobile application to be developed as a web-based game that will run on multiple platforms.

## Requirements

Since the game already exists on Android mobile devices, the company wants to expand the platforms by enabling access to the game via the web. The game will have at least one team involved, and each team will have multiple players. Each team name must be unique and cannot be identical to another team name. Each game will go four rounds with one minute intervals. If not guessed in the first 30 seconds the opposing team has 15 seconds to guess.

## [Design Constraints](#_2et92p0)

The client would like this game to run on all devices considering it is already running on Android devices. The need for at least one team to be involved in the game. That way there is some sort of competition between either teams or amongst the individual teams. Another constraint is that each team must have their own unique name. If that team name is already taken then the user is notified to try again. This will encourage the competition if the company would like to use leaderboards of team names in the future. The last constraint is that there are four rounds that are one minute intervals that will give the initial team 30 seconds to guess. If the image is not guessed correctly, the opposing team has 15 seconds to try to guess.

## [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)

As shown in the UML diagram below The Team, Player, and Game glass all inherit attributes from the Entity class, making Entity a superclass. Therefore inheritance is shown in the diagram. Also, there are many references going on as well, hence the one-to-multiple relationship in between the classes. GameServices references the Game class, Games references the Teams class, and Team references the Player class.

**"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** | Some of the characteristics are that macOS has more consistent operating system and better hardware integration.  An advantage is that there are less security vulnerabilities.  A weakness is that it is less used compared to Windows or Linux | It is an open source operating system so a lot of resources and tools are free. Therefore, it is most preferred since the cost is so low.  An advantage is that you can adjust your security settings. Meaning you can give access to authorized individuals. A weakness is you need to download applications to suite your needs. | A characteristic is that is most used platform.  An advantage is less loading time and a well familiarity of the platform. A disadvantage is that the security is average. It loses stability allowing viruses and malware attacks. | A characteristic would be more mobility and less in a stationary spot. However, you would want a stationary platform. An advantage is that it is lower cost, and flexible. A disadvantage is that it has higher security risks, and bandwidth limitations |
| **Client Side** | The cost is moderate and comparable to windows. There is moderate time considerations. There needs to be some kind of expertise when developing software on a macOS. | The cost is near minimal, despite the monthly costs for applications (if any that are costly). A lot of time is needed to adjust to the system if new to it. There needs to be a high level of expertise when it comes to Linux. | The cost is comparable to mac (which is moderate) in order to support multiple types of clients. The time is minimal due to this being a popular choice, and the expertise is minimal as well. | The cost would be medium since the cloud server would already be established. The time would be moderately high, since there is a need to work around updates. The expertise would be minimum. |
| **Development Tools** | The relevant languages can be but not limited to Ruby, Java, and Python. Macs can essentially run all languages, but some IDEs and other tools include Swift (which is the most popular), Visual Studio, and AppCode | The relevant languages that Linux use are PHP, Perl, JavaScript, and Python. Linux can run a lot of languages, but some IDEs that are used to create web hosted software applications include Visual Studio, Sublime, and Atom. | A relevant language that we used was JavaScript in order to develop our application, another language that can by used but not limited to is PHP, and Python. An IDE that we used was Eclipse, but Visual studio and other IDEs can be used as well | A lot of relevant languages are used when using a mobile device for this type of software development. Depending on what mobile operating system that is implemented such as android or mac or cloud. You can use various tools or IDEs such as Visual Studio, swift; However, there are others that can be used that are not listed |

## 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 that the client starts the process on windows operating platform. Since it is the most user-friendly operating system, as well as the cost being so low compared to the other operating systems. Therefore, if the company went with the windows platform, they would save money and increase the likelihood of users interacting with the Draw It or Lose It web-based application.
2. **Operating Systems Architectures**: Windows provides Windows-based services so that the operating system can display graphic images on the application while accessing resources and other components in the system. The user either has to sign into an account or interact with a server in order to use the services of the operating system.
3. **Storage Management**: Windows 11 comes with an added feature called Storage Sense which is located in the settings of the desktop. This feature allows you to maintain files as well as the space of those files on your hard drive. This enables you to save projects to the cloud storage system so they cannot be lost or deleted by mistake.
4. **Memory Management**: When programming the web-based application we would need to create a database that will store the images that will be rendered each round. This memory allocation will allow for pictures to be located outside of the normal “Pictures” folder we have on our local computer. This allows for more compression of the current project and will enable it all to be stored in one place.
5. **Distributed Systems and Networks**: When researching what software development environment to use in order to have the game Draw It or Lose It be cross platform, I found that Unity was a great application development tool. This IDE enables a designer to create different dimensional games that can be ran across multiple platforms. When the application is completed, you can export the file onto the web, and can be accessed by platforms such as: Nintendo Switch, Xbox, iOS, Android, and computer consoles. This development tool also has access to add virtual and augmented reality in case the client wants to add those platforms down the road. This will help with individual dependencies of each platform. When it comes to outages or connectivity, The Gaming Room will need to make sure their servers are adequately conditioned to withstand high player volumes as well as a backup power supply incase of power outages of any kind. This will ensure the game continues to run and does not crash, leaving the current users frustrated that the round and/or game force quit without saving the points the teams have received.
6. **Security**: Windows comes with a built in security algorithm. Therefore, it is already protected in some ways such as malware, viruses, and threats to the security. As long as the user keeps good maintenance of the machine, they will be a little more secured if a security threat does occur. Since threats change, just like strains of viruses, it is better to keep up-to-date on the different types of threats that way you are better protected if one does arise. Maintaining updates will ensure a little more protection on user information.