

<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 | 11/08/23 | Nathan Book | Draw It or Lose It currently is only available on Android platforms, The Gaming Room would like to expand the platforms in which their web-based game is accessible. |

**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)

Draw It or Lose It is currently only available on Android platforms and The Gaming Room would like to expand their client base by offering the game on additional platforms while addressing specific functionality requirements. To resolve this, we will begin developing the web-based application for Windows which will be integrated with its current version on Android.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

The system must be designed to handle a large number of games and players, it must be scalable. The system should also be secure and only permit authorized users. The game must be accessible at all times throughout the day, 24/7.

## [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 classes Game, Team, and Player inherit from the Entity class. This means those three classes will share the name and id references. The association between objects in the GameService, Game, Team, and Player is called aggregation. GameService has an association with Games, Games an association with Team, and Team an association with Player. This is identified in the UML diagram with the solid line connecting the classes. The 0..\* notation indicates that there could be 0 to many interactions between those classes.

**"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** | Easy to use commands for server configuration, access, or amendments. Mac servers are scalable, has various options for web hosting, but considered the least preferred web hosting environment. | Most secure web host. Security flaws are identified quicker. Even though Linux is the more preferred choice, it is more difficult to find applications that support Linux web hosting. | Most widely used OS. Higher than other OS’ resource requirements, less load time, and seamless usage. Susceptible to viruses and inadequate tech support. | Portability, cost-effective, and better compatibility. Specific to one mobile device OS unless developed for various mobile platforms. |
| **Client Side** | Moderate level of expertise and time. Cost is similar to Windows. | Maximum level of expertise and time required. Low cost. | Minimum level of expertise and time required. Cost similar to Mac. | Maximum level of expertise and moderate amount of time required. Moderate cost. |
| **Development Tools** | Macs can run all programming language IDEs.  HTML, CSS, and JavaScript required for web development. Eclipse and Visual Studio IDEs. | Linux is mostly written in C.  Visual Studio Code will be used as the IDE. | Windows can run all programming language IDEs. HTML, CSS, and JavaScript required for web development.  Eclipse and Visual Studio IDEs. | Visual Studio IDE.  C and C++. |

## 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 is a cloud-based server. With a cloud-based server, The Gaming Room can meet all their client’s needs while maintaining the scalability and cost-effectiveness of the project. There are a few options for cloud services, Google Cloud Platform, Microsoft Azure, and Amazon Web Services. Either of these services would be an ideal choice and gives the stakeholders options to work within their budget.

**Operating Systems Architectures**:

The recommended operating system architecture is a multitier client-server architecture. This is where the presentation, application processing and data management functions are physically separated. This type of architecture allows for scalability, security, and flexibility.

1. **Storage Management**:

The appropriate storage management system that should be used in compliance with the architecture is Microsoft SQL Server. This is a relational database management system or RDBMS. These management systems efficiently permit the storing and retrieval of data.

1. **Memory Management**:

This operating platform will use virtual memory management techniques. This will allow for the memory to be stored to the computer or device for ease of access. This will optimize memory usage and improve performance.

1. **Distributed Systems and Networks**:

The communication between various platforms can be accomplished by using an API Gateway. It acts as the “guard shack” for all entry requests while also handling the authentication, security, and load balancing. This will address the security, reliability, and back-up concerns. The networks that interconnect these systems are LAN, WAN, and Wireless, all of which are compatible with the recommended architecture.

1. **Security:**

The Gaming Room is not responsible for the security in place on the individual devices in which the application will be installed, but there are methods to assist in the user’s information security. The platform will have built-in security features like data encryption, authentication, and secure network communication. There are also several coding practices that provide security by design.