

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 |
| --- | --- | --- | --- |
| 1.1 | 07/26/2022 | Richard Wood | Updated this workbook filling out the table required for V1.1 |

**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)

Company name is The Gaming Room. The Gaming Room wants to develop a web-based game that serves multiple platforms based on their current game, Draw it or Lose it which is currently only available on Android. The application will render images from a large library of stock drawings as clues, where teams compete to guess what is being drawn. It will consist of four rounds lasting one minute each, drawings are rendered at a steady rate being fully complete by the 30-second mark. If the team does not guess the puzzle before time expires, the remaining teams have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit.

## [Design Constraints](#_2et92p0)

* Must run on multiple platforms
* Only one instance of the game can exist at any time
* Game and Team names must be unique
* A team must allow for multiple users
* Inherit swift for Apple devices
* Finding a way to use existing code to run across multiple OS

## [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)

Entity is a superclass that has developed three relationships where they inherit its information. The three classes that have a relationship with Entity are: Player, Team and Game. Team and player have a “has a” type relationship. The same relationship happens with Game and team, and Gameservice and Games.

**"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** | Popular choice in web hosting with its upgradeable various options  Less preferred for web hosting services  Flexible terminal | Advantage for Linux is how secured the OS is  Cost friendly  With how secured, it is difficult to find applications to support web hosting required needs. | A high resources most used OS, a lot of software available. A lot of resources, and most user friendly.  Disadvantage is that it is easy to get a virus using Windows. | Most popular as most people have smart phones, making it have wider reach.  Poor security compared to others |
| **Client Side** | Cost is like Windows. The expertise and time needed is mid-ranged | Cost is at a minimum. Max expertise and time required for Linux. | Cost is like Mac. The expertise and time needed is the least in comparison to other OS. | Cost is a minimum, expertise and time needed is maximum. Hardest to implement. |
| **Development Tools** | The most popular language for Mac is Swift. Not limited to languages though, Mac can run all languages. Visual studios is a good tool to use. | Linux can work with various front end languages such as CSS/javascript and HTML, and backend can be Java, Python, C++. Linux can use visual studios as a nice tool | Similar to Linux, Windows can use all types of languages and IDE’s. Windows is easier to use then Linux. Windows can work with front end languages such as CSS/javascript and HTML, and backend can be Java, Python, C++. | Most common for apple devices is swift, and android for android. Both can be run on all three machines and can use all front end and back end languages stated like the others. |

## 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**: My recommendation for the operating platform of choice is Windows. Windows has the most software available, most tools, and requires the minimum expertise. Windows is also a very common OS for games.
2. **Operating Systems Architectures**: Windows has services used by all Windows-based applications that allow a GUI, while accessing system resources.
3. **Storage Management**: The best storage management for Windows is database management system. It is easy to use and runs in multiple operating platforms due to its high compatibility.
4. **Memory Management**: Memory allocation allows for easy storage of pictures outside the default picture folder, this game needs a database with lots of pictures.
5. **Distributed Systems and Networks**: LAN as the networking so it can work as a hub allowing multiple systems to connect at once, allowing all users of the game to play. LAN works if one computer crashes, the game still operates fine.
6. **Security**: Windows is a secure operating system as it comes with a lot of built-in defenders such as windows defender, and malware protection. This keeps users’ information safe