

# **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 |
| --- | --- | --- | --- |
| 1.0  1.1  1.2 | 9/17/2022  9/31/2022  10/15/2022 | Ryne Williams  Ryne Williams  Ryne Williams | Updated executive summary, design constraints, and domain model.  Update Evaluation of different operating platforms  Update Recommendations. |

**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 Gaming Room seeks to move their mobile game app, Draw It, or Lose It, to a web-based application. The game consists of 4 rounds each lasting 1 minute. Each team will have a chance at guessing the drawing, which is rendered over 30 seconds, before the next team gets a chance at guessing the answer. This will need to operate on web browsers for Windows, Mac, and Linux.

## [Design Constraints](#_2et92p0)

* Must be compatible with Windows, Mac, and Linux based web browsers.
* Each game must support 1 or multiple teams
* Each team will have multiple players
* Teams and players will be required to have a unique name and check to see if a name is already in use in other teams or players.
* Must have only one instance of a game in memory at any given time.
* Each game must have 4 rounds.
* Each round will last one minute
* After the one-minute mark, other teams will be given the chance to answer within 15 seconds if the previous team failed to guess the right answer.

## [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)

In this UML diagram we can see how the code will be laid out for the web-based application. The ProgramDriver class, which is used to “drive” or operate the code uses a SingletonTester class to ensure that only one game is active in memory at any given time. The Game class uses a class called GameService to ensure that only one instance of a game is created and used at any given time, this uses the Singleton design. The Game, Team, and Player classes all inherit from the Entity class which sets up the teams and players for each game.

**"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** | Mac is the most expensive of the options as far as hosting web applications. However, it has relatively secure servers and good support. | Linux is the most secure option for hosting web applications, and it is open source and free to use. The downside is that many applications will need to be altered to be able to migrate out of Linux. Also, Linux can be a bit more difficult to use for those who are not familiar with it. | Windows is the most commonly used operating platform used in web-based applications and most will be familiar with its structure and interface. It also has very good support when needed. While windows isn’t quite as expensive as Mac, it is still expensive and not as secure as Linux or Mac. | Mobile development for web-based applications is possible; however, it is quite impractical as servers are not equipped to properly handle applications used in web browsers. |
| **Client Side** | The cost for Mac would be the highest as the hardware and software components are the most expensive on the market. It would also require a moderate amount of time and money for training for those who are not familiar with its interface and coding languages. | While the Linux operating system is free to use and download, it would require a large amount of time and money to train those who are unfamiliar with its interface as it is the one that requires the most expertise, since there is little to no support for this platform. | Since Windows is the most widely used platform on the market, the cost for training would be minimal. Also, while still cheaper than Mac, Windows can still be rather expensive depending on the hardware requirements needed for web development. | Mobile platforms would be more time consuming in the process of making applications that can run on web browsers, as this is not its intended purpose. However, the cost in time and training would be relatively low since it is a well-known platform. |
| **Development Tools** | Mac uses iCode IDE with the Swift programming language. | The Linux operating system uses the C programming language but also supports Python. | Windows has the largest variety of coding languages at its disposal, but primarily uses Visual Studio for C++. Other languages include Java, Javascript, Python, among others. | Most mobile platforms use the Android Studio IDE for application development. Android Studio uses Java for its programming language. |

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**: Windows seems to be the best operating platform to begin working on the “Draw It, or Lose It” web application. This will allow for minimal code change when implementing the application, since Windows is the most widely used operating system.
2. **Operating Systems Architectures**: The Windows API provides services used by all Windows based applications and allows applications to access system resources, as well as implement a Graphical User Interface and audio into the application. It has thousands of callable subroutines and includes functions such as Base Services, Component Services, Graphics & Multimedia, Messaging, Networking and Web Services.
3. **Storage Management**: Windows comes with Windows Storage Management Provider built in to the operating system and can be used for a wide range of storage configurations.
4. **Memory Management**: Windows allows each process in 32-bit to have its own virtual address space, enabling addressing up to 4GB of memory. Each process in 64-bit is capable of addressing up to 8 terabytes of memory. Threads cannot access memory being used by another process, which prevents memory from being corrupted by another process. However, all threads can access their own virtual address space.
5. **Distributed Systems and Networks**: By utilizing the remote login network operating system users will be able to access their profiles for “Draw It, or Lose It” remotely from their desktops or smartphones, and they will be able to play games with their friends or other random users, as they will all be able to access a remote server set up specifically for that game.
6. **Security**: Windows has a very robust security suite in its operating system to help to ensure no viruses or other malicious software enters the operating system. There are also a wide variety of security software available to Windows computers. The information going to the servers and users, however, would need to be encrypted to ensure that their information and the information of others is not retrieved by intruders or unintended users.