

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.3

## 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 | 03/20/2022 | Breanna Hodges | Added executive summary, design constraints, and domain model. |
| 1.1 | 04/01/2022 | Breanna Hodges | Added development requirements for Mac, Linux, Windows, and Mobile Devices. |
| 1.2 | 04/17/2022 | Breanna Hodges | Added recommendations. |
| 1.3 | 04/20/2022 | Breanna Hodges | Updated executive summary, design constraints, and development requirements. |

**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)

Our new client, The Gaming Room wants to develop a web-based game that works on multiple platforms. The web-based game will be based on their current game, Draw It or Lose It. This game is currently only available in an Android app. In this game, the application will show images as clues to the teams where they will guess a puzzle. There are four rounds that last one minute each. The drawings are loaded at a steady rate and are complete at the 30 second mark. If the time expires before the team guesses the puzzle, the other teams will have a 15 second chance to solve the puzzle. Requirements include: a game must be able to have one or more teams involved, with multiple players per team; game and team name must be unique; only one instance of a game can exist in memory at any given time.

## [Design Constraints](#_2et92p0)

* Since the game already exists as an android app, we need to develop an identical design for users that already have the game installed. Another option would be to have a completely different design than the one already active, so it feels like a different game altogether.
* We would need to make sure the server can handle multiple players simultaneously.
* We must have a system that checks if a name is already in use, and if so, suggest one like the one they tried to use so the user does not get frustrated about not being able to come up with a unique name.
* A system that assigns unique IDs for each game instance.
* Since the game uses a large library of drawings, we need to make sure we have licenses or copyrights for any of the images used if needed. Theses images also need to be compatible with multiple operating platforms.
* A web server to host the application
* Storage for images
* Quickly rendering images for games
* Security for user’s information

## [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 UML diagram below is used to provide a visual design of the gaming system. We can see by looking at the UML diagram that the Entity class will be a super class that Game, Team, and Player classes will inherit from. In each of the classes, we can see the variables and methods that will be used to develop this game. The inheritance relationship between these classes is shown by the arrows from each of the sub classes to the super class. The 0..\* represents that a game can consist of from zero to many teams. Starting on the left where GameService is, we can see that there is a linear relationship between all of the classes on the bottom row.

The GameService class manages the games currently being played. This class ensures that only one instance of a game can be active at any given time.

The ProgramDriver class uses the SingletonTester class, shown by the arrow connecting the two, to ensure that only one instance of a game exists at a time. The ProgramDriver class is where the game is executed by creating instances of the other 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** | Apple, the company that created MacOS, offers many advantages for hosting a web-based software application. Apple is the most secure, the US Army uses Mac servers. You also get the support of Mac applications. One of the biggest weaknesses is price, not only is the hardware expensive but the cost of web hosting is also expensive compared to the other options. | The biggest advantage of Linux is that it is open source, and it is the cheapest option. It does not get updated frequently so you don’t have to worry so much about the site going down. It is also fully customizable. It also has a learning curve and a lot of people do not know how to use it. So, you would need to find someone familiar with Linux. | Windows has the most usable applications and tons of support. Most developers are familiar with Windows so that is also a plus. As far as hardware goes, it is much cheaper than Apple. You do have to pay for licensing, which can be expensive though. It has a ton of updates which can cause your site to go down occasionally, which is a huge weakness. It is also the least secure option. | Hosting web-based software on mobile devices is possible, but it is not suitable for the work that needs to be done. |
| **Client Side** | The cost can be compared to Windows. According to Apple, “You don’t need to be an experienced webmaster to set up websites and host them  on Mac OS X Server.” Time would depend on the expertise of Mac, someone that has no experience would take a little longer than someone with a ton of experience. | The cost for Linux is the lowest, but the expertise needed is the highest. Therefore, the time would also be more. Since Linux is not commonly used, you would need to find someone familiar with it, or train someone to use it. | Since Windows is one of the most widely used, time and expertise would be the lowest. Most people have used Windows at some point in their life. The cost can be compared to Mac, although the initial investment is the most expensive. | Since mobile devices are so widely used, expertise should not be much of an issue. It could take more time than a traditional OS, because there are so many different platforms used on mobile devices. The cost would be fairly low. It does provide flexibility since changes and updates can be done from virtually anywhere. |
| **Development Tools** | Mac hosts run Apache servers, and JBoss, Apache Tomcat, and Apache Axis are included with Mac OS X. This allows for all basic web code to be used. Including, but not limited to HTML, CSS, JavaScript, Java, Python, PHP, and Ruby. IDEs such as Visual Studio, AppCode, and Xcode are all popular with Mac OS. | Much like Mac, Linux uses Apache servers as well as the same languages, Java, Python, PHP, Ruby, HTML, JavaScript, and CSS. IDEs such as Visual Studio, Eclipse, and IntelliJ IDEA are all popular with Linux to name a few. | Windows can use many different web hosting servers. Any programming language is available to Windows users, and most IDEs as well. Visual Studio, a Microsoft product, Eclipse, PyCharm, IntelliJ IDEA, and many other IDEs are available on Windows. There are also many more tools available for Windows than for Mac or Linux. | As stated above, a mobile device is not suitable as a development environment. |

## 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 recommend Windows for the operating platform. It is one of the most expensive options, but it has the largest userbase and it is popular for game development. The need to train someone on how to use it is very unlikely. Windows allows developers to design and develop apps using a suite of Microsoft products. There are many cross-platform IDE’s that can be used on Windows, such as Xamarin, React, or Cordova.
2. **Operating Systems Architectures**: Firstly, Windows has two OS architectures, a 32-bit and a 64-bit. Windows is separated into two modes, User-mode and Kernel-mode. User mode consists of system services, system library, and explorer/GUI. This is the part that affects what the user sees and interacts with. Kernel mode consists of drivers – networking, graphics, devices, etc. – executive layer, and hardware abstraction layer. Windows supports multiprocessing and modularity of hardware, allowing for a more customizable experience.
3. **Storage Management**: Cloud storage will be the best option for the Draw It or Lose It. You don’t have to worry about space as you would with physical storage, and the startup cost is less. As the game grows, they can easily add more storage without the hassle of buying extra physical hard drives. Azure, offered by Windows is a great option. They have competitive prices, continuous updates, and great customer support. Within Azure there are multiple options such as, Azure Disk Storage, Azure Blob Storage, Azure Files, and storage containers.
4. **Memory Management**: For the 32-bit, 2GB of memory is allocated to the kernel and 2GB to app memory. For 64-bit, 8TB of virtual address space is given to each the kernel and user application. Virtual memory is much like cache memory but provides more space. Provides a default heap for each process and file mapping. File mapping works great with websites without requiring the entire file to be mapped to memory. Of course, as time has progressed, Windows has updated their memory management to be faster and more efficient at loading.
5. **Distributed Systems and Networks**: Since the game is web based, we do not need to consider which OS the user has. All computers and smartphones can browse the web with an internet connection. We do need to make sure the game is mobile friendly, since most things are scaled down from a computer to a phone. The most important thing is to make sure that the game can be accessed by any web browser. There could possibly be complications with internet connectivity or lagging, which cannot be controlled by developers or testers.
6. **Security**: Windows has a ton of new security features on Windows 10. Biometrics authentication – facial recognition, iris scan, or fingerprint – a new API, Windows defender, windows SmartScreen, trusted and measured boot – identifies and blocks malware during the boot process – device encryption, and a device guard feature where only trusted code can be run, along with many other security features. Windows has taken steps to ensure their clients are protected with extra security measures. While Windows has taken these steps, it is still important to implement routine security checks to ensure none of the user’s information is exposed, as well as making sure developers are writing secure code and keeping user information encrypted.