

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 | 06/04/2022 | Kyle Raja | Software design document to develop web-based game app. |

## [Executive Summary](#_sbfa50wo7nsh)

I am excited to introduce the software design for the web-based game app, “Draw It or Lose It”, for The Gaming Room. I will be going through the technical side of the development strategies to create the app in a web-based environment. My analysis will cover important requirements that will be needed towards building the app and have it running in web environment. “Draw It or Lose It”, will need to have the capabilities of allowing multiple teams as well as players. These software designs will be based on the Java coding language which does have the capabilities to be utilized on other platforms. I will be addressing not only specific software design requirements but also specific operating system requirements for the app.

## [Design Constraints](#_2et92p0)

There are design constraints for this project that need to be addressed before development.

One main constraint is that the application is web-based. Desktop and web-based application differ in that web-based applications need to have the capabilities of being able to be used on various platforms.

Currently, the game functions only on the Android device. For a web app, other programming languages may need to be used such as HTML, CSS, JavaScript, or Python. The app will also need to be able to perform on other Operating Systems like, Macs, Windows, and maybe even Linux. It will also need to have the capability of running on other mobile devices such as iPhones, not only Android.

As far as the constraints of the program while its running are concerned, we see that multiple teams are required for competition. This would include more than one person playing the game. We also need to make sure that one game is played at a time.

## [System Architecture View](#_ilbxbyevv6b6)

This section was skipped due to the instructions. I assume we will come back to this section later.

## [Domain Model](#_8h2ehzxfam4o)

The classes that are most important in the UML diagram are the Entity, Game, Team, Player, and GameService class. An important OOP principle that in the UML Diagram is Inheritance. The three classes, Player, Team, and Game, all inherit from the Entity class. Inheritance helps the developer share important functions or variables between classes in order to not have to constantly write more and more code. By having classes share with one another, tremendous time and energy can be saved. Inheritance also improves organization in the code. Looking at the methods and variables in each class, one can see how they relate to another class. For example in the Entity class, we see by the variables and methods, that we are getting the id and name of the player or team. In the Team class, we user the addPlayer() class to add the player, using the name variable of the Entity class. We see that the Player class also has the id and name variables, which helps Team add the appropriate information of the player. We see the Game class also use the id and name variables from the Entity class. Most of accessor methods are in the GameService class that is used to get the information of the game name, game id, the count of games, and next team id & name. In order to organize all of this information, we need to have lists for the players(Team Class), teams(Games Class), and games(GameService Class). So we see that each class is sharing information with one another while the program is running. The ProgramDriver class contains our main method which represents when the program is executed. The SingletonTester class is used from the ProgramDriver class. This makes sure we are playing one game at a time.

**"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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Macs offer cost-friendly options for servers on the AppStore. Servers can be configured and secured. Downside may be that more advanced servers and security could be costly for the Mac.  For web hosting and server deployment, I found that the Mac OS X Server offers the ability to host web apps. This could be useful for the video game. Bluehost is also another option that do provide services at a monthly cost. This may be the better option because bluehost can also provide services to other OS. Bluehost is around $29/month and does require a license key. HostGator is much cheaper, starting at 3/month however they have plans that reach to $90/month depending on client needs. | Flexible OS, allowing users to have more control. Linux has more affordable options for server hosting. Strong security options. The problems with Linux may be that not many are familiar with the OS. For web hosting and server deployment, I found that Linux could use HostGator or Bluehost. Each do require a monthly or yearly payment option. It may be more cost effective to have other companies provide the web services since they already have excellent servers, staff, and 24/7 monitoring. Bluehost is around $29/month and does require a license key. HostGator is much cheaper, starting at 3/month however they have plans that reach to $90/month depending on client needs. | Key advantage is others will have more familiarity with Windows. The OS has been around longer and is more widespread among corporations. Cost-friendly and secure. Downside may be that it can become quite complex to set up given the various options. Windows can use either Bluehost, HostGator, or Amazon plus a number of other hosting services. Windows is the most universal so my recommendation is if we go with Bluehost, we can apply it to windows easily. Bluehost is around $29/month and does require a license key. HostGator is much cheaper, starting at 3/month however they have plans that reach to $90/month depending on client needs. | Key advantage would be more cost-friendly. Disadvantage would be performance, given the simplicity of the OS in comparison to the other OS. May not be as secure. HostGator is one of the most popular for Mobile OS webhosting. I still recommend Bluehost because they provide services for all of the OS. We could go with hostgator strictly for Mobile OS to save some money. Bluehost is around $29/month and does require a license key. HostGator is much cheaper, starting at 3/month however they have plans that reach to $90/month depending on client needs. |
| **Client Side** | Advantage would be a nice user interface that the Mac offers. High performance, great graphics, and audio. Disadvantages would be of developers learning how to construct the UI. Compatibility from an Android app to a web-based app for Mac could also prove challenging to transition to.  The cost to setup would likely be quite great in order to properly have the integration between multiple platforms. A Mac setup would be similar to windows so I believe the differences in cost should not be to far different. Time solely depends on expertise and company resources. Higher skilled employees will likely get the job done faster however would cost more. In order to insure that the application can be used on multiple platforms effectively, proper integration must be set up on the configuration and code side. This would require high skill. I recommend using emulators or a cloud based platform in order to accomplish cross platforming. | Developing the user interface on a Linux OS could be quite complicated, more experience may be required. Outside talent may need to be hired which will cost more money. However, Linux User Interface may offer more customizable features.  The cost to setup Linux may be cheaper than the Mac and Windows however the time and expertise may be the more difficult. Because many are not very familiar with Linux as with the other OS, it may require more time and specific experts to set up the game in the Linux environment. Integration will likely be complicated.  I recommend using emulators or a cloud based platform in order to accomplish cross platforming. | The user interface for Windows will be more customer friendly. More people will know how to use the app. The expenses should be average. Less likely for outside hire to be needed.  Likely easier to create.  The cost to set up on Windows OS should be around the same as for Mac. Time will depend on the level of skill that is hired. Because windows is the most popular OS, it is likely that finding high skill for the Windows should come at a lower cost than Linux or even Apple and Mobile OS. Windows is excellent at integration with other OS.  I recommend using emulators or a cloud based platform in order to accomplish cross platforming. | Likely would be the most convenient for customers to use, a game they can play on their phone. User Interface could be more tough to do given the restrictions from mobile operating systems. Outside hire is likely, which could cost more money.  The cost to set up Mobile OS should hopefully not be as expensive as the other OS. Time and expertise will likely be high since Mobile OS is still quite new compared to the other OS. Specific skill will need to be hired. I recommend using emulators or a cloud based platform in order to accomplish cross platforming. |
| **Development Tools** | Macs I believe can utilize web-based coding languages quite well. Languages such as HTML, CSS, and JavaScript. Popular IDE such as Visual Studio can also be used. Cons would be the time it may take for developers to become familiar with Macs if they are use to using Windows.  Macs can also use Visual Studio Code which I believe many employees would be familiar with. They offer a free IDE service that does not require a license. However, to get their main IDE, Visual Studio, that would require a license and payment. I believe Visual Studio for Mac 2022, is the IDE. | More options to customize, more tools to work with. Linux can also use Visual Studio and use coding languages to build web-based applications. Cons would be that developers would need to much more familiar with Linux in order to be able to use it effectively.  Linux can also use Visual Studio Code but can not use the main Visual Studio IDE.  I feel that Visual Studio Code would be best because developers that are familiar with Visual Studio, will also be familiar with Visual Studio Code. There is no cost to use Visual Studio Code. | Much more familiar to use among the population. Most development tools can be used on Windows that I discussed with Mac and Linux. Cons would be not as much options to customize like Linux.  I recommend that for Windows we use the most updated IDE for Visual Studio. I believe it is Visual Studio 2022. It works great with Windows and is excellent with languages such as C# and C++ which are great languages for game development. There is a license and payment required for the IDE. Visual Studio Code can also be used. | Most of the development tools I describes can also be used to create Mobile applications. Cons would be that outside hire may be required to use these tools in order to develop an app for a mobile device. However, many of the same coding languages and IDE can be used.  In order to use Visual Studio for Mobile OS, Visual studio does have an extension called Visual Studio Android extension. This will give the developers an IDE to work on for the game on a Mobile platform. Visual Studio must already be installed so that would mean we would already have a license and made payment for it. |

## Recommendations

1. **Operating Platform**:

Given my own experience working with Windows, Mac, Linux, and Mobile Devices, I would recommend that the company choose Windows as their standard OS. This is because it is likely to be the most hassle free and cost effective. If it’s developed on other another OS, it’s possible that outside hire help may be needed, driving up costs. With windows, there are also many examples online to use to help the development team push the app online.

Now for the server platform, I would recommend the Windows Servers. Since I am recommending that the companies primary OS be Windows, it would only make sense that the company decide to go with the Windows Servers. Windows Servers are highly compatible with other Operating Systems. In todays age, virtually any platform can work with Windows. Of course, skilled developers must be hired in order for proper integration between the servers and operating systems be performed.

1. **Operating Systems Architectures**:

Windows provides a number of services that could help integrate the android app towards a web-based environment. WebApi is one that I can think of. We also have services such as single sign on and click once.

Now the benefits regarding having a WebApi is that it can be accessed over the web and is layered by certain protocols. API is more easily described as an applications interface over the web. So for example, if one were to go to Facebook, they would see the interface immediately.

Interface can be thought of as the design or layout of the program. The security or protocols are HTTP which stands for hypertext transfer protocol.

1. **Storage Management**:

The company could use some cloud-based storage system for storage management that can integrate with Windows. Windows also has services where you can organize your hard drive such as disk cleanup. The IT department could also have routine cleanups on their servers to manage storage. Windows provides those kinds of cleanup tools and services. Also, upgrading storage space for windows could be easier than upgrading on other OS.

To expand on why a cloud-based storage system is beneficial, the key word I can think of is backups! Cloud is incredible with backups, so no information is lost. It also is more efficient at storage. It is also safer to use cloud storage rather than ones local storage because there are certain security measures such as encryption. Now what encryption is, speaking from cloud, is where readable text is converted into unreadable text before it is placed in the storage. This makes it difficult for hackers to make sense of the data. So basically, cloud storage offers more layers of protection and also provides better accessibility for the user.

1. **Memory Management**:

Well, I recommend our configuration are set with 64-bit which I believe allocates around 16TB of useable memory. More memory will be needed which can be done through servers for temporary storage. Games will likely not need to be saved so temporary storage space could be used for those purposes.

Draw it or Lose it, uses images during its puzzle game for the user to determine the phrase. The images are drawn from a large memory bank, stored on the servers. Certain temporary cache memory can be set up (temporary storage) to only store the images relevant to the phrase as the user gets closer to solving the puzzle. The images can be organized into certain categories and be loaded in the memory cache based on the category. It could increase performance time/ preventing lag between the client and server.

1. **Distributed Systems and Networks**:

First, we will need to consider data storage that will be shared for cross-platforming among the various devices. I recommend having a shared database for all the game information. Windows already has the capability of being a multi-user operating system. We can use Visual Studio, Windows, and servers to distribute the game among different platforms. Our servers will need to be reliable, so I recommend that we have at least two system administrators always monitoring the servers.

Well as stated, Visual Studio could be the main IDE that can be used to develop the game on various platforms even Mobile OS. Servers can be set up to control the various platforms using switches. We can monitor the safety of the data being processed from the network via firewalls.

I recommend we have Windows Servers because they can be easily used on other platforms. One dependency I can think of is having a strong CPU processor, because the servers are going to be doing a lot of work transferring data to the various platforms.

1. **Security**:

Another role for our system administrators is cyber security. More help can be hired if needed. Windows does have security protocols in place which can also be enhanced by anti-virus software. Firewalls can also be put in place between the network and our servers, providing further protection. Main concerns would be viruses and hacking. I assume we will be taking credit card information from customers for purchasing the game, so I recommend a strong antivirus software and firewall be put in place.

Since I am recommending Windows Servers, the potential for strong security is high. Windows Servers offer layers of protection. The configuration tools are quite simple. There are things like key authentication, VPN, file transfer protocols, and password requirements. Single Sign on is available or Two-way authentication. It may be a good idea to hire one cyber security IT administrator in order to improve security measures.