

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 | 03/18/20023 | Matthew Williams | This shows one of the many design documents that need to be made in order to try to get the Draw IT or Lose it game off the ground and create a web based game for people to play. |

**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)

<Write a summary to introduce the software design problem and present a solution. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.>

The game Draw It or Lose It is a game that they would like to be a web application that works all sorts of devices, there will be players and there will be in teams and the player will be presented with an image from a huge image library and they will have to help their team guess what the image is. The game is supposed to four, one minute rounds and then once the time expires if the team did not get the answer then the other team will have 15 seconds to have one guess at the image in order to get the point. The company would like it so that there is only one unique game name and that at least two teams are in every game always having multiple players.

## 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)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

The application needs to be web based to play so it will have to work with all platforms. It needs to be able to create a game with four rounds and those rounds lasting one minute in length and then it needs to have a round where the opposing team can take a guess lasting 15 seconds in time. They have to have at least two teams with multiple players on those teams that will draw or guess during their turn. There can only be one game with the same name at one time on the server, the team and player names also must be unique. The company does not know how to do any of this themselves, so it is entirely up to you to find out how to make this web based game come to life. The software that is going to be used is not yet decided but will be decided soon.

## [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)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

With this UML Diagram we see that the diver and singleton class are separate from the others the driver only has the main code of the program so that it is executable and the singleton has the code in it to if there is a game name out there with the same name. Then there is the Entity class that holds all the common attributes of the program this class then inherits to the Game, Team, and Player class directly those classes hold the needed code to have a game and then to have a team and then to have players on those teams. The zero to many things mean that there can be one game service and there is many games but that game can have many teams and that team can have many players but it can not be the other way around. In the game service there is the code to actually control the game, Team, and Player 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** | Advantages: they have good developer tools, they allow server based deployment  Disadvantages: You have to have a Mac in order to compile the finished product of your program for mac.  They have a licensing fee that is expensive if you want to develop applications you need to have that | Advantages: does not have memory leaks like windows, this system does offer server based deployment, there is a version that is free to download, has some good security features in the System  Disadvantages: harder to learn and not as widely used as windows so could be harder to find developer that could develop with Linux | Advantages: Windows has a lot of features and tools that you can use, most people already use this System, they do have server based deployment in place  Disadvantages: System comes pre-installed with some things that you may not need/want,  Windows supposedly does not have the best privacy in mind for the user, the licensing cost seems very expensive for this system to the client it is at the lowest 600 dollars, | Advantages: apple devices and android devices offer server deployment, licensing costs seem a bit cheaper for mobile devices then for computers  Disadvantages: mobile devices are good operating system but never have quite the capability that a pc system would have, things take longer to develop on mobile devices |
| **Client Side** | To make sure that it works on everything it will need to be tested on all platforms so you will a macbook and at least a windows computer to test the compatibility.  Time for developing is said to be in the middle for developing on mac. Swift is preferred language but the others are compatible to develop on mac | People say that Linux is hard to get used so it requires some time and expertise to learn but when you do get used to it they say that it is the best system. This is the cheapest operating System to develop on. Python and c are their languages that they are built on the most so requires expertise in those | Cost would be high for higher for this system but it would not take as long for this because almost everyone has probably used at least a windows computer and if not will not take to long to learn how because it is supposedly the easiest to learn how to operate windows, the only thing needed is the correct applications and licenses | I read that mobile development is a bit harder to develop on and requires people to have more skill with programming and that if they have a background in designing apps it helps because the user interface needs to be set up well and easy to understand for the user, so that would mean that the cost would go up and the expertise needed and also the time would go up because if mobile devices are more similar to an app and the interface has to be more specific that would make this set of systems the longest one for development time. Also for this one There could be more than one development team . |
| **Development Tools** | It looks like many different languages can be able to be used on mac, they use Unix based systems so they can have a lot of applications and the needed tools that are able to be used on the systems. You can use languages like Html, python, java, and Ruby easily plus others so the choice is up to the development team for the language. But Apple itself uses XCode and swift which is a language that they created and can be used for a lot of things so that is what they recommend to use for development.  From what I read there is not fees to develop on mac systems you can use the ide’s and application tools for free. | You can use quite a few languages to develop on Linux but it is recommended to use some type of c or python to develop an application on Linux | Windows is another System that you can use lots of languages but most people say use some type of c because that is what it is built on but something like python or java should work fine. The development team probably already has some knowledge of windows so that should not be too bad of a task for them, the cost for the development tools are not too bad | Another that you could end up doing with multiple languages but would definitely need android studio and xcode and Visual studio these are all free ideas to use |

**\* ALL** of the Systems in the above table can be used to develop the html web based application and html is said to work well with all of them so the language that they chose to use should not be a problem at all depending on the System may just need to download some extra tools to develop the html website

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

**Operating Platform**:

I think that the best operating system that I came across in all the work for this course is the Linux operating system or one based on Linux. It seems like that this would be the best operating system for this because it is the most adaptable to what ever you need it to do, and it is a powerful system and does things very quickly and does not require too powerful of a computer to work on and should work great for the game application.

**Operating Systems Architectures**:

I think that Linux is a good os system to use for creating this website because from the looks of things it has some good stuff that you can do with Linux and it is an easily scalable platform, it has a strong kernel that can handle input and output very well and is known to be very good at processing things and managing memory well. Also, Linux is known to have a good shell system for the programmer that is also a nice feature that you can configure how you want them to be. I think that a good architectural pattern that the developer should follow is the client server pattern because this works fine with the operating system architecture and it is a good pattern to follow because it is one of the most well known patterns because it is easy to setup and then have a very adaptable architecture that can then be changed whenever the developer needs it to be .

**Storage Management**:

I think that the thing for them to use for storage is a SSD because it will allow for faster times than the use of other ways of storage. Linux manages its storage with something that is called block storage and it just means that it uses a type of storage that is easily accessible to the computer and is referenced with calling the chunk of storage that it needs to find to fulfill the request, making it an easier and quick way of holding and sorting the storage.

**Memory Management**:

The Linux os system is good at managing its memory it is known to use something that is called demand page loading which is a good feature because it can make it so that the system can operate faster than some other ones because instead of loading in the entire program at one time it can do it slowly by only needing to load what is needed into memory based on what is requested and can save time and memory by working with memory this way.

**Distributed Systems and Networks**:

The Linux system from the looks of things should not have many outages and should run perfectly fine since it is a command line os system the client just has to send it requests and it will be able to react to the request that it is given. There could be some downtime for the game server if there was something that had to be changed in the web application but that could just be done during a time of low activity on the website and the developers could go in and make the changes that needed to be made. If you did not want to have any downtime I think that you could have it automatically switch to a different server if some sort of server crash did happen instead if during a sort of system maintenance on it you could change the server so that the new tweaks to the application are built into the new one.

**Security**:

I think that the best way to keep this application secure is to have a role based security system so that only the creators of the game can change it and the regular players can only play the game. Just like the drop wizard package one that we messed around with for a few weeks in the beginning of this term. Then for an extra layer of security there could be a multistep authentication process to prove you are the user that you say you are. So if there was some sort of game sign in to get back into your prior gameplay and stuff that this would protect the users accounts and there could be a sign in also for the developers of the game to sign into their admin accounts in order to pickup where they previously left off with multiple steps in order to get into those accounts so that they can remain more protected.

Before this course I had not really known much at all about this operating system but now after learning more about it and I think that it seems like a very good operating system that I would like to try to figure out how to use. I read a bunch of stuff about the Linux system and while I know absolutely no one personally that uses Linux from everything I read and looked at throughout this term I do not know If I have heard much of anything bad about it and think that is why it would be the best system for this game because it seems cheaper, faster, and better than all of the other operating systems that we learned about. The only downside of it from the looks of things is that it is a little harder to use so something like finding a developer for this entire application would maybe be harder for the company but I think would be better in the end to have it developed and operated on Linux.