

<Draw it or Lose it>

Version 3.0

## 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 |
| --- | --- | --- | --- |
| 2.1  2.2  3.0 | 01/17/23-01/26/23  02/03/23  02/14/23 | Brandon Porter | -Updated Requirments and updated Executive Summary Implimented in the documentation  -Diegn Constraints also updated to, Domain Model, changes to header and rivisions.  -Revised on 01/26/2023 for appropriation and submission  -Revised on 02/03/2023  - Added Evaluation documentation to template as Project 2  -Project 3 submission is a copy of project 3 as project 2 had the answers and work for project 3 completed and submitted. |

**Evaluation:**

YES All of the platforms offer a server-based deployment method where the website will be hosted. For licencing and potential cost to the client linux is a free based server it also has open source as well. Which is why it it if cost free. There are other tools like scaling and and programming languages that can all be obtained for free and that is one of the best benefits of using linux. There are also weaknesses and we will go over that. The other option that would be best suited is windows in the sense of server and best reliability. Windows is well know, getting people who are educated or can be educated easily and obtain information on the internet is easy when it comes to windows. As far as linux, there are a lot of do it yourself programers and people who like to tinker and I think there is a fair amount of information out there and support. Going over the pros of linux is open and secure source that is free and cost efficient but it does come with less support . Windows is a more secure way to gom it costs to use windows and can be on the expensive side of things but the support level again is much better than that of linux.

Client side of windows is good we’ve been using Eclipse and that has been rather simple the language is more known we can use VS code and really anything with windows. On the Mac side of things we some people may use xcode (I have never used it, not for any real reason) and the programing languges that is best used based on research is object C or swift. I have used Mac for everything so far and from experience it is not friendly with Maven and is a little more of a pain to use. Rightaway I notice the benefits of Windows. From the Linux side, Eclipse seems to be the most popularly used IDE. C is a defult however. Moving forward there can be costs related to developing tools depending on the process and the platforms being used, Mac and windows cost money as stated above and could be cost impacting and drive down the value to the company. Linux howver would be free with its open source but getting the proper people to build with in it would be more of a challenge that could lead to negative costs. The impact on the team is tough because they need to determain what options are best for them as a team and again using the different lagaunage on various platforms isn’t going to be the easiest, we would want a team and support dedicated to each platform as well as communication between the developers to ensure that what is happen on one platforms corrolates with the other in respect to each separate server side.

A compatiablility test would be required to make sure that the application runs accorss various browsers. This test ensures that the software application is functioning accordingly with different browsers, databases and OS, mobile devices, network and hardware. This derives form a non functioning software test. That means usability, reliability and performance the esures the application and customer both have saticfied results. The two major types of tests are backward and forward compatibility tests. The backwards compatibility is important if older devices are being used to run the program. The company would want someone with a 5 year old Mac or an Samsung s10 to not have the ability to run the program. The Forward monitors the applications success and assessing the software for future versions of both hardware and soft war with new bulds.

With in the testing there is version, browsing, hardware, software, network, device, mobile, and OS testing.

* Version - Ensures that is is compatible with all the versions of the software
* Browsing - Ensures it performs properly across all browsers
* Hardware - Assess the performance around it with other configurations
* Software - Test the Application and verify it is successful, including many other ways
* Network – Test the connection across multiple networks like 3G, 4G, 5G and wi-fi
* Device – Proper performance with different devices
* Mobile – Checks to ensure the software works with all mobile devices
* OS – Software works with all operating systems. AS studied in this template.

There is a lot that goes into this process like the possible defeacts (many to list) and the tools that would be used to do the combatability testing. Again many to list. At the end, there is a lot that goes into having an application that runs on all of the devices and works effectively and one thing to note that is important. If this is a game, it would be best to consentrate on mobile compatability. That will be where this game find the most use. The application needs to be user friendly and the application needs to be effective on cellular devise and that means that this needs to work with Iphoen and and android apppropriatly. The app needs 100% efficiency abroad but the emphasis should come at the mobile side.

## [Executive Summary](#_sbfa50wo7nsh)

Develop a game that that is web based and can run on multiple platforms. At this time the game is only able to run on android. The name of the game is draw It or lose it. Ideally the game will be multiple teams consisting of several people going four rounds at a minute each. After an imagine is pulled from the library one team will attempt to solve the problem till that teams time is up. There will be 15 secons for each team member to answer on the opposite team if the answer is never given.

## Requirements

* *A game will have the ability to have one or more teams involved.*
* *Each team will have multiple players assigned to it.*
* *choosing a team name.*
* *Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.*
* *Names and teams will be special to ensure that there are none similar being used*

## [Design Constraints](#_2et92p0)

* Will run on more than one platform and function.
* The names will be special to ensure it is not being used.
* There can be one instance of the game at any time
* There will have to be multiple teams
* There will be more than one person on each team

## [Domain Model](#_8h2ehzxfam4o)

The entity is suppose to have a relation between the other classes, such as teams and player class. In this situation they are all suppose to get the proper information from the entity program. The objective it to show this when looking at the UML and verifying that there is infact a match going forward. The UML is going to allow us to cross reference the way our code is going to run as well as giving us a literal picture to evaluate from. Looking at theis we see that there is a lot of referencing, Game service will be a reference of games and games that of teams and so on. We know by evaluating the diagram that team as well as players work on a “has a” type.

**"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

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | When accessing or making changes the terminal is better suited. In terms of web hosting it Can be upgraded, making it more preferred. There are other options for other types of hosting  Macs are not really preferred for hosting even though they are widley used in management roles this does not make them great on a hosting stand point | They catch any security issues before it becomes A large scale problem, Linux is more secure which can make it desireabel to use.  Trying to find applications that you can use with it would be a downsidem its not as popular as windows or mac even with its benefits. | Microsoft has a tremendous amount of software in comparison to other Operating Syetems, it is the more prominent platform to use. It has many resources the loading time is not bad and because of its widely used it has a larger following.  Tech support has been noted to be worse and windows does seems to be more pron to virsus, this is likely due to its wide range of use abroad. | Mobile devices are more popular than most other devices as many people have mobile devices. This means they are more redily available and everywhere in comparison to that of a standard computer. This means it could be better used for profitablility and customer base. The security isn’t very good, most people don’t have a back up antivrus to help protect it and most people phones are not secure. |
| **Client Side** | Mac is an amazing platform, it has a large following and is simple to use. Very compatible with apple products and very suer friendly. Supprot is great when needed and most anything you need is on the internet. Gmaing is a no go for the most part, maybe one day we’ll see a change in that. | Linux requires experience and understanding and patience learning. Cost is cheap, appears to be compatible with just about every platform you can use. The hardest thing is familiarity. The support side is likely limited as well | Windows take sno time at all to get use to. Great for everyday user as there is a large following (the largest) and you can find anything you need on google or youtube to support a user in their efforts on any platform and coding. Arguably the best option | Not easy to implement, greta in the sense of user ability and functionality but when talkimg about mobile we have to remember there are other operating systems as well that range from no experience needed to good experience needed to unlock the potential. |
| **Development Tools** | Swift is a more popular option to run though I’ve never done it. Mac also has decent tools that can be used as well. As you dig into the tools you need experience. Typicallly the tools are usles for a beginner but eventually the mac tools become useful. Mac is great for anything although python and java are the only things I have used. Mac runs HTML, CSS, JavaScripts Python, Java and many more like PHP. As a mac user I have never used it for its full potential and I love it. | Linux seems to work with all the platfor, has many tools and that is what makes it such a great system to use but this system is for people that are advanced, people that have mastered windows or mac and wanted to try something else and have more capailites with less limitations, it is my understanding that all of the basic platforms run great on it. Great for front end developing like mac, probably has more abilities and tools. runs HTML, CSS, JavaScripts Python, Java and many more like PHP and Ruby | Windows from my understanding and what I have found is a more widly used sytem that has similar benefits as linux, there is a lot of free range that you can do, there are apps for just about anything and windows always has access to thoughs apps making it one of the best systems to use. No matter whre you go there is a windows being used and the ability to find thinsg on the web when trouble shooting is easy. Windows runs HTML, CSS, JavaScripts Python, Java and many more like PHP and Ruby and is not limited. Like linux, you can do about anything. | You can do a lot with mobile devices including creating apps that use swift and android. The software can also be run on all three types of machines and that make sit very versatile. Languages are The same as the other machines. |

## 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**: The Gaming Room should start on windows devices as it has more software available along with minimum experience needed and cost less to get projects going. The use of windows is more widley used and the program can be worked later for all platforms.
2. **Operating Systems Architectures**: Windows has better services and they are used by all windows based apps that can show the GUI while accessing the system and the resources with in the system.
3. **Storage Management**: All of the devices have great storage management system that are very easy to access, according to my research, windows could be the better fit because it has other features for storage that assit. In my experience I have only used mac, storage and location have never been an issue while using mac and I would argue that all the systems equally have a great storage management syetem
4. **Memory Management**: The project is going to require memory space and with in the librarys to access the game, There is not winner on memory manegment. They all can have picture folders that make accessing easier and there is no winner for this situation, only preference.
5. **Distributed Systems and Networks**: Modern distributed systems have evolved to include autonomous processes that might run on the same physical machine, but interact by exchanging messages with each other. Benefits Unlimated scaling, adding machines when needed, having machines that are closer to the user will reduce tinme to serve a user. If there is a fault and a server goes down other could still serve the users devices. The negative is data integration, network and communication faults and the amount of management over head.
6. **Security**: Security protection software is standard with windows. The system should have other security added to help it along. The system has the ability to scan for threats regularly but Mac would be better fit for security from my ready. Mac’s generally don’t get attacked as much because they are less used, however the security in them does seem supireor and the support is good as well. We can also add more protection to the mac.