A close-up of a sign

Description automatically generated

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | 7-15-2023 | Elliot Huh | Original creation |
| 1.2 | 7-29-2023 | Elliot Huh | Updated Evaluation Table |
| 1.3 | 8-12-2023 | Elliot Huh | Updated Recommendations. |

## [Executive Summary](#_sbfa50wo7nsh)

The client company Creative Technology Solutions would like to develop a web-based game that is based off their current android only game. The game is named “Draw it or Lose it”. Help is needed to streamline development and help facilitate the web-based version of the app. The solution is to create a Java based application. With utilizing object oriented programming the application will be at its most effective and efficient state.

## Requirements

* *web based version of the app that can serve multiple platforms.*
* *Game has ability to have one or more teams.*
* *each team can have multiple players.*
* *Ability to use unique names for teams and games and check if name is already being used for teams.*
* *Only one instance of the game is allowed to exist in memory at any given time.*

## [Design Constraints](#_2et92p0)

* Compatibility: the web application should be accessible to as many platforms as possible, there are many different platforms and browsers that may need to be accounted, this can constraint development as it can take resources and extra time to make sure these platforms and browsers are able to run properly.
* Security: safety and precautions are important to consider. Because the nature of the web and game, there will need to be security measures in place to protect users and their data. this can constrain development as it means more time and resources but also more careful decision and planning.
* Performance and quality: the web game is based off a mobile game so there is a performance and quality expectation that is needed to be reach. this is a big constrain as there is a clear goal or bar that must be reached in development. Also, with a growing user base it’s important that performance doesn’t hinder and can handle the demand.
* Network constraints: since this game is online and real time network and online capabilities need to be accounted for, this is a constraint as specific expertise will be needed and will take time and resources, issues such as latency, crashes and outages will all need additional time and effort to develop.

## [System Architecture View](#_ilbxbyevv6b6)

web based distributed environment. There are multiple servers and levels. such as a web server that handles the web pages, application server which handles things like user interactions and manages the game, database server which handles memory and data. This architecture helps the interaction between users and the games in the most efficient way possible.

## [Domain Model](#_8h2ehzxfam4o)

This UML represents the Domain model of the game. There are 7 total classes. The Entity class acts as the base class. it provides common behavior and attributes that are inherited by 3 other classes. object oriented programming principle of inheritance is seen here as we can see Game, Team and Player classes inherit properties and some methods from the Entity class.

We see that Game Service class acts as the central service for the games as it contains much of the necessary functions and methods for the game to run properly and functionally. Game Service holds much of the technicalities for the game to run. The Game class holds information about a game and has some functions that are related to Team, Team holds functions and information about the teams and players within the team, game to team and team to player has instance with the corresponding classes., Player holds data about players. Game Service, Game, Team Player, and Entity are all connected.

Singleton Tester uses the Program class to test singleton, program driver acts as the driver class.

**"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** | can offer server-based deployment where website can host using built in Apache server or numerous 3rd party software’s to achieve the same goals. The way the terminal commands are set up makes updates or changes flexible. It’s fairly popular for web hosting although not as popular as Windows and Linux. there are licensing costs, and the costs are higher compared to other operating systems. | is incredibly popular; it offers many features that help it stay secure and stable. offers deployment where websites can be hosted using servers such as Apache or Niginx alongside many others. Its highly customizable and able to tailor towards specific needs, has a great assortment of tools and commands to keep it incredibly efficient although it can come at a higher learning curve but due to its open-source nature, there are no licensing costs and will overall be the cheapest option. | is very popular if not the most popular. it has an ease of use and is incredibly familiar for many users. Windows may have access to the most amount of software application options to help it develop. Offers server-based deployment where websites can host using Apache, Niginx or IIS, or many others. Windows offers an incredibly diverse toolset. Windows does have licensing costs and security may not be the best out of the operating platforms. | Mobile offer tools and development application that are native and unique to mobile devices. IOS and Android devices are the primary mobile devices for development. Mobile devices communicate with back-end web applications for deployment, all servers are usually in the back end to deploy. many applications are hybrid frameworks to deploy across multiple different platforms and devices. |
| **Client Side** | Mac offers moderate expertise as it may not be as familiar to the general Populus although its Ui is incredibly efficient and easy to use. Costs may be higher than other operating platforms, there are numerous licensing costs. There are many robust tools available however there are also many tools that are restricted and not available on mac, there are significantly less options on Mac compared to other platforms such as Linux or Windows this may add in some time. Security is fairly good here thanks to the Apple support and monitoring; Apple Ecosystem adds in a lot of functionality and efficiency can save time through this way | Linux has the highest learning curve and highest expertise needed out of all. Due to its open-source nature this is also the cheapest and most cost-effective platform out of the other platforms. Linux has an incredibly robust tool set and has access to many tools and libraries. with the correct amount of expertise this can be incredibly time effective although getting that expertise may add on a lot more time. But if all those expertise are met the diverse toolset and customizability can potentially make this the most time effective. | Windows is incredibly popular and easy to use, this is the lowest expertise needed. Windows also has the greatest number of applications and tools at its disposal, these together can potentially make this the most time effective out of all the platforms. Costs are moderate, there are many options although many have licensing costs. Overall this has the lowest expertise needed and the option with the most diverse toolset and applications. | Mobile has its own learning curve and expertise will need to be slightly high. Mobile is restricted to only mobile tools, but they have their own diverse toolset that make it effective. Time may not be the strongest however, Cost is more on the cheaper side, and mobile can be developed alongside other platforms. |
| **Development Tools** | Mac has many developments tools although not as much as Windows or Linux, its more restrictive to the amount of tools and applications that it can use. Some of these tools include Atom, Delphi, Cloud 9, Eclipse, Visual Studio, Zoho creater, Embold, Pycharm, IntelliJ DEA Built in Apache. there are many others and numerous third-party applications. licensing costs are dependent on the tool being used, some are free however may have licensing costs. Mac does tend to have higher licensing costs. due to the amount of tools offered, teams may not be needed but can indeed help with production and benefit the project. Mac is able to access many if not all programming languages and has many abilities and potential for development. | Linux has access to many developer tools, it has a wide range of tools including many built in tools that are unique to Linux. It has a wide access to many libraries and is capable of running every programming language. Some of the tools Linux can access and run are LXDE, KDE, Linux Kernal, Kwatee, Axure, Eclipse, Jira, Atom, Visual Studio, Vim, Sublime Text, Seamonkey, Quanta, BlueGriffon, there are many more. Some of these applications may have licensing costs although Linux is open-source based and many applications will be free, and costs will be generally cheaper than anywhere else. Due to the learning curve maybe more teams may be needed although the diverse toolset may boost performance and abilities to teams. | Windows has access to the most amount of applications and tools, some of these tools include Visual studio, eclipse, Axure, Atom, Xcode, Cloud 9, Jira, Delphi, Kwatee. There are many more and can arguably be the most diverse and have the most amount options. There are also many Windows specific tools. Licensing cost are dependent on the developer tool chosen, Many have licensing tools although many are free as well, Windows is on the moderate side of costs. Due to its low expertise needed and diversity and amount of tools available there is no need for more development teams although it can benefit the project, Their can be many positive changes for development changes due to ease of use and diversity of tools | Mobile are more restricted to their mobile development tools, Mobile can potentially be developed alongside other platforms but mobile has their own tools and development applications. IOS and Android are the primary mobile devices and their tools will be split by the two. A few examples for IOS are AppCode, Xcode, Swift, Flutter, Interface Builder, there are many more, A few examples for Android are Unity, Xamarin, AIDE, Android Studio, Ionic and many more. costs tend to be on the lower side licensing cost is dependent on the tool being used. Can potential restrict development teams and more teams may be needed. |

## 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**: When looking at the available operating platforms there are 3 clear choices. The issue with mobile devices is that the game is already available on android. Whilst the expertise and resources may be abundant for the client, there are many advantages and benefits that will be missed by staying on mobile devices for further development. This leaves us with 3 options. Out of the three windows is the best and the one I would recommend. Whilst mac can be incredibly efficient with its own slew of benefits the restrictions on the applications and the limited choices make it a hard platform to really justify hosting a server for the game. Linux is a great option and is seen as the one of the most if not the most popular ways to host a server, it offers many helpful tools at incredibly low costs, but it can require some expertise. The thing about this project is that the game and the subject may not really need to utilize those special tools and those expertise costs may not truly be justifiable for this project. Windows offers an incredible number of approaches to this project and there is a truly diverse way to build up this server. There are many options and costs are negatable considering the low expertise needed. There are numerous tools in here to help serving a website. With amazing hosting tools such as HostGators and Ionos there truly is a lot to be had here and an amazing amount of benefits that can be utilize very well for hosting a website.
2. **Operating Systems Architectures**: Windows offers a multi-layered system. There are two different modes that can be utilized, User mode and Kernel mode. User mode is a private virtual address space that has more limited access but more safety for device and applications whilst kernel mode is on a public virtual space that offers more access and more possibilities however less security and more chances for damaging systems and applications. These two modes can be switched between each other and prove useful. These two are the main two layers that are able to be utilized and are critical to the windows software systems architecture.

For the game it’ll be important to utilize a server-client based architecture. This offers the most effective way to host the type of game that’s being served. Due to its design, scalability will be strong and be able to be further developed to handle growth of new users and be flexible enough to meet up with demands and needs as they come. The centralized structure can make this easy to maintain and upgrade. Its real-time benefits and capabilities will prove useful for this game being on multiple platforms and support the ease of access live games service.

1. **Storage Management**: There are many areas where storage can be improved or utilized better. First step is to utilize development techniques that condense or make the game less taxing on storage. Windows offers a huge selection of hardware choices and potential tech choices. There are many options and brands that are available. I would recommend trying to stick to SSD drives for hardware storage options. They offer many benefits on top of the original HDDs, such as lower load times, less energy to run, etc. utilizing good file explorer organization and declutter drives may prove useful. Due to the game’s cross platform nature a cloud drive may be crucial for the players progress to transfer over onto other devices. There are many options to consider on windows including their own Microsoft options but utilizing will be important. There are also some built in features from Windows platform that may prove useful, these can include deduplicating. Storage sense, and numerous tools to help troubleshoot and manage storage.
2. **Memory Management**: Similar to storage management options there are many ways to improve and help memory management in similar yet very different ways. Development approaches and techniques really help here as well. Utilizing a more condensed overall scripts and codes will help memory run faster. Some specific techniques such as pre-loading or streaming will reduce the load on ram and improve memory, these help loading and get things loaded up faster without using as much memory. cache mechanisms will be critical here as well, often used assets that are ready and having temporary memory will all help take the load off of memory. Hardware upgrades and options are available as well, upgrading RAM will greatly improve the memory however most of the decisions made during development and in the software application will be more important and have a bigger impact on the client as a game that requires too high a need of a RAM that’s too high can be extremely inconvenient and bothersome. Thanks to Windows there is a wide variety of RAM and other hardware options available to help with RAM. There is also numerous tools built in the system such as paging, memory mapping and other tools to help with memory management.
3. **Distributed Systems and Networks**: Windows has may tools and features that help support the deployment of systems and networks and many tools to help build these systems and networks to distribute. One of the supported models on windows is RPC model (Remote Procedure Call). It’s a technology that helps develop distributed client-server-based programs. It’s a type of system where devices can communicate with other devices through connected networks online.

There are numerous third-party options available here as well. The numerous IDE’s and programming tools that are all able to help develop and add onto these systems as well. There are numerous third-party applications and tools that are available on windows that can get applications communicating with the web. Windows has a very diverse number of applications and tools.

1. **Security**: Security may come down to the user and may most importantly be about the individual’s decisions and knowledge but there are many things the developer can do to ensure safety. First off windows offer many security measures such built in anti-virus software and other anti-virus applications, security support options, and built in security measures and tools. There these measures can be useful to detect and manage harmful or malicious threats on the game. There are some important measures that should be considered when developing the game. One feature that should be added is having a game account with a password. this ensures identity within a user and adds a layer of distinction between users. on top of that there are numerous options to further protect users. 2-factor authentications can ensure that user locks down their account. Adding in a firewall will be important to help detect threats. The handling of user data is critical and ethically important.