

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 |
| --- | --- | --- | --- |
| 1.1 | 04/1/2023 | Sean Toon | First edition of “ Draw it or Lose It.” |

**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)

Creative Technology Solutions has taken on a new client; The Gaming Room. The Gaming Room has reached out to CTS for help to develop a web-based game that is based off their currently owned game “Draw It or Lose It”.

## Requirements

The Gaming Room wants a game that will be playable on multiple platforms while meeting all their software and future hardware requirements. CTS will need to set up the correct project environment and facilitate the web-based version of the gaming app.

## [Design Constraints](#_2et92p0)

The software constraints:

* Game must have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to prevent duplicate team names when choosing a team.
* Only one instance of the game can exist in memory at any given time.

Hardware constraints:

* Hardware requirements will come later because of software application development decisions.

## [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 package in the UML Diagram is com.gamingroom. There are 4 different classes that inherit from the class Entity. Entity is the class that is used for all attributes. The Team class uses the Player class, the Game class uses the Team class and the GameService class uses the Game class. There is a program driver class that is used for the main() method which is where all of the objects will be created. The SingletonTester is used to check for a value of the GameService, to make sure that there is only one instance of GameService. The Team class will allow multiple teams to be created with multiple players assigned to each team and each will be involved in the finished game.

**"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** | Characteristics:  Mac uses the macOS operating system and is based on Unix.  Mac systems have high quality hardware components.  Advantages:  Mac has great security features.  Mac operating systems are user-friendly.  Mac operating systems have high performance due to top notch hardware components.  Weaknesses:  Mac operating systems are not compatible with some software applications.  Mac operating systems can be costly.  Hardware maintenance can be more challenging than other OS. | Characteristics:  Linux is open source.  Linux has robust security features.  Linux is known for its stability.  Linux is highly scalable.  Advantages:  Linux is cost-effective because it is free to download and use. Linux has low and inexpensive hardware requirements.  Linux can be configured to meet different needs such as web-based software application.  Linux has large community support which helps developers find solutions and troublshoot issues.  Weaknesses:  Linux can be less user-friendly than other operating systems.  Some software may not be compatible with Linux.  Linux has a shortage of formal technical support options. | Characteristics:  Windows provides many services and tools for web-applications. Windows is known its user-friendly interface.  Advantages:  Windows has Internet Information Services (IIS) which is a web server that supports a wide range of web tech and is customizable.  Windows also supports ASP.NET.  Windows provides many security features like built in firewalls, and user- authentication.  Weaknesses:  Windows can be more expensive than other operating systems such as Linux.  Windows has higher hardware requirements than some operating systems. Windows is less secure than some operating systems. Windows can be less stable than some operating systems such as Linux. | Characteristics:  Mobile devices usually have smaller screen sizes than desktops and laptops.  Mobile devices offer a range of input methods such as touch screens, voice commands and keyboards.  Compared to desktops and laptops mobile devices has limited battery life, limited storage and limited processing power.  Advantages:  Mobile devices are more portable than laptops and desktops. Activated mobile devices with cellular network are always connected to the internet.  Weaknesses:  A mobile devices small screen size can cause display issues with web-based software applications.  Mobile devices have limited input methods.  Mobile devices limited battery may affect usability.  Limited storage can constrict a user’s ability to save data.  Limited processing power lessens the performance of mobile devices. |
| **Client Side** | Cost:  Additional hardware components and software tools may need to be purchased to ensure compatibility.  Time:  Can be time consuming because different versions of applications may need to be created to ensure compatibility.  Expertise:  Developers will need knowledge in multiple programming languages, Game Developing frameworks, User interface design, cross- platform development and more. | Cost:  Linux is very cost-effective when considering the cost of hardware, software and maintenance. Linux is free to use although there may be additional costs for upgrades.  Time:  Linux’s time – efficiency is based on the amount of experience that the user has with the operating system.  Expertise:  Developers will need knowledge in multiple programming languages, Game Developing frameworks, User interface design and more. | Cost:  Compared to other operating systems, the price of maintaining and using a Windows operating system is about mid leveled. New tools and upgrades may need to be purchased.  Time:  Extra time may be needed for training to ensure hardware and software development compatibility with multiple clients.  Expertise:  Developers will need knowledge in multiple programming languages, Game Developing frameworks, User interface design and more. | Cost:  Developing an application with or for mobile devices can be costly because the application must be tested and optimized for multiple platforms. Additional services and tools may need to be purchased.  Time:  Because applications will need to be optimized for multiple platforms, development cycles will also increase greatly.  Expertise:  Developers must have experience and knowledge with iOS and android and other platforms. |
| **Development Tools** | IDEs and other tools used to build web-based software applications on MAC operating systems:  -Languages  \*HTML  \*CSS  \*JavaScript  -IDEs  \*Visual Studio  \*Qt | IDEs and other tools used to build web-based software applications on Linux operating systems:  -Languages  \*C  \*C++  \*Python  \*Java  -IDEs  \*Eclipse  \*Visual Studio  \*Qt | IDEs and other tools used to build web-based software applications on Windows operating systems:  -Languages  \*HTML5  \*JavaScript  \*CSS  \*TypeScript  -IDEs  \*Visual Studio  \*WebStorm  \*Atom  \*Sublime Text | IDEs and other tools used to build web-based software applications on mobile devices:  -Languages  \* Java/Kotlin  \*Swift/Objective- C  \*JavaScript/ TypeScript  \*C# |

## 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 as the best Operating Platform for creating the web-based game Draw It or Lose It.
2. **Operating Systems Architectures**: Windows is a commonly used Operating System, so the amount of training and outsourcing needed for extra expertise will be reduced. Many programing languages and IDEs can be used to develop the software. Software development tools and hardware will be relatively cheap and efficient.
3. **Storage Management**: There are many storage management systems that can be used with Windows but the most common system that is used is the default file system used by Windows , NTFS or NT (New Technology File System).
4. **Memory Management**: Because the web-based game “Draw It or Lose It” will be used across multiple platforms and will hold a large database, significant amounts of memory will be used. One technique that can be used to improve memory performance is virtual memory. Other memory management techniques that can be used is memory paging, memory compression and memory allocation and deallocation.
5. **Distributed Systems and Networks**: For an application such as the game “Draw It or Lost It”, to be able to communicate between various platforms the software can be designed as a client- server application. This will allow the app to be run on various platforms.

Network connectivity must be strong so that all the components can communicate effectively. Two ways that can ensure connectivity of the system is load balancing and redundancy.

The system should also be prepared with keeping things such as possible power outages , hardware failures, software crashes, or natural disasters. Practices such as replicating data, load balancing and redundancy can reduce outages.

The system should also be designed to minimize security risks such as hacking, data breaches, and malware attacks.

**Security**: The system should also be designed to minimize security risks such as hacking, data breaches, and malware attacks. This can be done by encryption, access control and data backups. Windows operating systems have security features such as user account creation, password protection, Firewall, antivirus and anti-malware software, encryption, secure boot, and BitLocker.