

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 2.0

## 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 |
| --- | --- | --- | --- |
| 3.0 | 04/16/2023 | Hong Luu | Provided recommendations |

**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 (CTS) has taken on a new client, The Gaming Room, to develop a web-based version of their current Android app game, Draw It or Lose It. The game is based on the classic Win, Lose or Draw concept where teams compete to guess what is being drawn. CTS will provide a software design document and begin developing the game application to address The Gaming Room’s software requirements. CTS will work closely with The Gaming Room throughout the development process to ensure that the application fulfills their requirements and expectations.

## Requirements

The client’s software requirements include the ability to have one or more teams involved, multiple players assigned to each team, unique game and team names, and the existence of only one instance of the game in memory at any given time.

## [Design Constraints](#_2et92p0)

The primary design constraint for developing the game application in a web-based distributed environment is the requirement for scalability and reliability. The game needs to be able to manage several users at once while keeping a high level of performance. This means we need to make sure the game is reliable and can handle a lot of users. We will use different techniques to make the game work smoothly, such as making sure the game is hosted on powerful servers and using technologies that help the game load quickly.

## [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 UML class diagram for the game application’s Domain Model consists of four classes: Game, Team, Player, and Entity.

Game class represents the game itself. Team class represents the teams that will play the game. Player class represents the players on each team. Entity class is a base class that has common attributes and behaviors shared by all classes such as id and name.

In the UML diagram, Game class has a one-to-many association with the Team class, meaning that one game can have multiple teams. Team class has a one-to-many association with Player class, meaning that one team can have multiple players. Game, Team, and Player classes are inherited from the Entity class, meaning that they share some common attributes and behaviors such as id and name.

Many object-oriented programming principles, including encapsulation, inheritance, and association, are illustrated in the UML class diagram. Encapsulation is indicated by each class holding its own attributes and behaviors and protecting them from being accessed or changed from outside the class. Inheritance is indicated by the Game and Team classes inheriting common attributes and behaviors from Entity class, preventing code repetition. Association is demonstrated by the one-to-many relationships between the classes, which describe how they communicate with one another in the game application.

**"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** | Macs are a great option for hosting web-based applications because they are well known for being user-friendly and have high quality technology. Macs offer built-in web servers, making the deployment and hosting easier than with other platforms. Macs also have a reputation for being reliable and secure. Apple also provides learning sessions and customer support for users. However, compared to other platforms, Apple hardware and software license fees might be more expensive, which might be a drawback for some customers. | Linux is known for being robust, secure, and flexible, which makes it an excellent choice for hosting web-based applications. Not only Linux is stable and secure, but it also provides a large selection of server-side software packages. Linux is highly customizable and free of licensing fees due to its open-source nature. However, for people who are unfamiliar with the system, Linux may have a higher learning path, and getting support may be more challenging than with other platforms. | Windows is the most popular platform for hosting web-based applications because they have widespread usage and compatibility with many software applications. Windows provides a variety of server-side software packages and has a user-friendly interface, making it simple to use for people who are familiar with the Windows operating system. However, Windows may require more resources than other platforms and the licensing fees may be high. | Mobile Devices such as iOS and Android devices are becoming more and more popular for hosting web-based applications due to the increase in mobile phone usage. Mobile platforms provide a wide range of tools and resources for app development, and they have larger user populations. However, mobile devices may need more development time and experience to make sure it is compatible across devices and browsers. The fees of licensing mobile app development tools can potentially be high. |
| **Client Side** | The software development considerations include expertise in developing for the Mac platform and using Apple’s integrated development environment (IDE). The development team will also need to consider the cost of purchasing Mac hardware and any necessary development tools. In addition, the team will need to test the application on different versions of the Mac OS to ensure compatibility. Also, the team will need to ensure that the application is optimized for performance on Mac hardware, which may require additional development works. | Developing a web application that works on Linux requires consideration for different web browsers like Chrome, Firefox, Microsoft Edge, and others. The development team may need to ensure the application is compatible with these browsers by testing and troubleshooting any issues. Moreover, developing a mobile application for Linux also requires expertise in developing for different screen sizes and device requirements. The cost and time required for development depend on the complexity of the application and the expertise of the development team. | Similar to Linux, developing a web application that for Windows requires consideration for different web browsers like Chrome, Firefox, Microsoft Edge, and others. The development team also needs to ensure the application is compatible with these browsers by testing and troubleshooting any issues. Windows mobile application development involves knowledge of the Windows Phone platform as well as testing for various screen sizes and device requirements. The complexity of the application and the level of expertise of the development team determine the cost and time needed for development. | Developing a web application for Mobile Devices requires consideration for different screen sizes, device requirements, and web browsers. The development team must ensure compatibility with popular mobile browsers like Safari and Chrome. Developing a mobile application for both iOS and Android devices requires expertise in developing for both platforms and ensuring compatibility with different devices. The cost and time required for development depend on the complexity of the application and the expertise of the development team. |
| **Development Tools** | For game development on Mac, popular programming languages include C++, Objective-C, and Swift. Xcode is the primary IDE used for Mac development, and it includes a wide range of tools for game development.  The technical requirements for Mac development may also have an impact on development teams, as many developers may not have experience with Mac-specific development tools.  Additionally, development on Mac may require access to a physical Mac desktop, which can add to the cost and complexity of the development process. Xcode and other Mac development tools are free to use. | Linux operating systems are open source, and many development tools and programming languages are freely available. Popular programming languages for game development on Linux include C++, Python, and Java. IDEs like Eclipse and NetBeans are commonly used for game development on Linux.  Since that many developers may not be as familiar with Linux as they are with other operating systems, the technical requirements for game creation on Linux may have an impact on development teams. However, because Linux is open source, it can be altered to meet certain development requirements. There are generally no licensing costs associated with Linux development tools. | Popular programming languages for game development on Windows include C++, C#, and Visual Basic. Visual Studio is the primary IDE used for Windows development, and it includes a wide range of tools for game development.  The technical requirements for game development on Windows may have less of an impact on development teams, as Windows is a widely used operating system and many developers have experience with Windows development tools. Also, compared to Mac development tools, Windows development tools are typically easier to set up.  Visual Studio and other Windows development tools are available for free. | For game development on mobile devices, popular programming languages include Java and Kotlin for Android, and Objective-C and Swift for iOS. Android Studio and Xcode are the primary IDEs used for Android and iOS development, respectively.  The technical requirements for mobile game development may have a significant impact on development teams since mobile development can require specialized knowledge and skills related to mobile hardware and user interface design. In addition, physical mobile devices may be needed for testing during the mobile development phase, which could increase the cost and complexity of the procedure. Mobile development tools usually don’t require any license fees. |

## 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**: For expanding Draw It or Lose It to other computing environments, I would recommend using Linux due to its robustness, security, flexibility, and affordability. Developing a web application for Linux requires consideration for different web browsers and testing the application to ensure compatibility. C++, Python, and Java are popular programming languages for game development on Linux, and IDEs like Eclipse and NetBeans are frequently used. Linux is cost-effective because there are no license fees, making it a good choice for The Gaming Room. Despite possible technological restrictions, Linux can be a dependable platform for their purposes with the correct tools and experience.
2. **Operating Systems Architectures**: Based on the Unix operating system, Linux is an open-source operating system. It is widely used in servers, embedded systems, and supercomputers and is very adaptable. It is intended to be reliable, efficient, and secure, and it supports a variety of hardware platforms. Due to Linux’s kernel architecture, the kernel serves as the operating system’s core and manages all communications between the hardware and software. Additionally, it supports virtualization, enabling the use of several operating system instances on a single machine.
3. **Storage Management**: For Linux, one good option for storage management is the Btrfs file system. Btrfs is a modern file system that offers many advanced features such as built-in data redundancy, snapshotting, and support for large file sizes and multiple devices. It is also suited for solid-state drives and capable of handling large amounts of storage. Btrfs is a wonderful option for both personal and business use because it performs well and is made to be simple to manage.

1. **Memory Management**: A storage management system for Linux I would recommend is Logical Volume Manager, or LVM for short. It allows us to break the hard drive into small logical volumes that can be adjusted or transferred without losing data or needing to shut down the computer. LVM also features a snapshot function that simplifies data recovery. It is straightforward to use since it comes pre-installed with most Linux operating systems. It is also adaptable and may be utilized with many storage technologies, such as drives and hard disks.
2. **Distributed Systems and Networks**: A distributed software system can be utilized to enable communication between different platforms in Draw It or Lose It. The game is divided into smaller components by this system, which uses APIs to connect them. The right network protocol should be adopted, and backup plans should be implemented in case of any network issues, to ensure that the system is fast and reliable. Both RabbitMQ and Apache Kafka are suggested tools for this use case on Linux. It is essential to make sure that every part of the distributed system functions effectively and can connect with one another without any problems.
3. **Security**: Linux operating systems provide a range of built-in security features that can be leveraged to protect user information. For example, Linux provides various authentication methods such as passwords and biometric identifiers to authenticate users, and these methods can be further strengthened by using two-factor authentication or multi-factor authentication. Linux also comes with firewall and encryption features that can be used to protect data while it’s in transit and at rest. Linux has an advanced file and folder permissions system that makes managing user access and authorization simple. Finally, updating Linux operating systems frequently is necessary to guarantee that the most recent security patches are applied, reducing the chance of security breaches.