

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 | 05/21/23 | Nathan Johnson | Cross-platform compatibility |

## [Executive Summary](#_sbfa50wo7nsh) The Gaming Room has approached Creative Technology Solutions (CTS) with the goal of developing a web-based version of their existing Android game, "Draw It or Lose It." The objective is to create a game that can be accessed and played across multiple platforms. Currently, the game involves teams competing to guess what is being drawn from a library of stock drawings. Each game consists of four rounds, with a one-minute time limit per round. If a team fails to guess the drawing within the given time, other teams have a chance to offer their guesses within a 15-second time limit. To address this software design problem, CTS proposes the development of a web-based game application that enables seamless gameplay on various platforms. By transitioning from an Android app to a web-based solution, the game can reach a wider audience and accommodate different devices. The proposed solution will involve cross-platform compatibility, ensuring that the game can be accessed and played on desktops, laptops, tablets, and smartphones. The gameplay experience will remain intact, with teams guessing drawings within time limits using stock images. Additionally, the application will incorporate a feature to check the availability of team and game names, ensuring that each team and game has a unique identifier. The software development process will follow an object-oriented approach, allowing for modularity, scalability, and code reusability. Overall, the proposed web-based solution will fulfill the client's requirements efficiently, providing an engaging and immersive gameplay experience for "Draw It or Lose It" across multiple platforms.

## Requirements

The client has specified the following business and technical requirements for the web-based game application:

1. Team Involvement: The game should support the participation of one or more teams, allowing multiple teams to compete against each other.
2. Team Composition: Each team should consist of multiple players, enabling collaboration and competition within each team.
3. Unique Game and Team Names: The application must enforce the uniqueness of game and team names. Users should be able to check whether a name is already in use before selecting it for their team or game.
4. Single Instance of the Game: The application should allow only one instance of the game to exist at any given time, ensuring a controlled and organized gaming environment.
5. Cross-Platform Compatibility: The game application needs to run on multiple platforms, including desktops, laptops, tablets, and smartphones. This will enable players to access and enjoy the game from various devices.

## [Design Constraints](#_2et92p0)

While developing the code and software for the game application, there are specific requirements that need to be followed. The objective is to ensure that the game runs on all devices, including Android, as well as other mobile devices and operating systems like Windows, Linux, and Apple. To achieve this, the codebase needs to be extended beyond Android and adapted for compatibility with other platforms. One approach is to rewrite the code in Swift to support Apple devices. Another approach is to leverage existing code and incorporate multiple programming languages to create a unified solution that can run on various devices, using the concept of language inheritance. This allows for the integration of different languages to enhance the strength and functionality of the code. These considerations highlight the need to address application development in a broader context, taking into account cross-platform compatibility and the incorporation of diverse languages to cater to the requirements of different devices and operating systems.

## [Domain Model](#_8h2ehzxfam4o)

The UML class diagram provided demonstrates the relationships between the Entity, Game, Team, and Player classes. The Entity class serves as a superclass, establishing a relationship with the Game, Team, and Player classes through inheritance. Inheritance allows these classes to inherit common attributes and behaviors defined in the Entity class, such as "name" and "id", promoting code reuse and ensuring consistency across the subclasses. Examining the relationships between the classes, we observe that the Team and Player classes exhibit a "has-a" relationship. This means that a Team has instances of Player, indicating a composition relationship where a Team class contains references to Player objects. Similarly, the Game class has a "has-a" relationship with the Team class, signifying that a Game consists of one or more teams. This relationship is depicted as an aggregation (HAS-A) in UML, illustrating that the Game class has a reference to an instance of the Team class. the GameService class is associated with the Games class. This relationship can be described as an aggregation, as the GameService class has a reference to a collection of Game instances. The Games class acts as a container for the Game objects managed by the GameService.

**"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)

Project 2 Revised Chart:

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac offers a UNIX-based operating system, providing a stable and secure environment for hosting web-based software applications.  Mac provides robust development tools and frameworks like XCode and Swift, allowing developers to create high-quality web applications with a focus on user experience.  Mac has a smaller market share compared to Windows and Linux, which may limit the reach and potential user base of web-based applications hosted on Mac. | Linux is an open-source operating system known for its stability, security, and flexibility, making it a reliable choice for hosting web-based software applications.  Linux offers a wide range of distributions and server variants optimized for web hosting, providing scalability, performance, and cost-efficiency.  Linux may require a higher level of expertise and technical knowledge compared to other platforms, which could pose challenges for less experienced developers or organizations without Linux-specific skills. | Windows is a widely-used operating system known for its user-friendly interface and compatibility with a wide range of software applications, including web-based software.  Windows offers extensive support for popular programming languages like C#, .NET, and ASP.NET, providing a rich development ecosystem for building feature-rich web applications.  Windows hosting can be more resource-intensive and costly compared to Linux, and it may require additional licensing fees for specific software components, | Mobile devices, such as smartphones and tablets, have limited computational resources, smaller screen sizes, and operate on mobile operating systems like iOS and Android.  Mobile devices offer the advantage of portability, allowing users to access web-based software applications on the go and enabling developers to target a large and growing user base.  Hosting web-based software applications on mobile devices can be challenging due to limited processing power, memory, and network connectivity, which may impact performance and user experience |
| **Client Side** | When developing software to support multiple types of clients on macOS, important considerations include cost, as Mac development often involves purchasing the necessary hardware and software tools, time, which can be impacted by the need for compatibility testing across different macOS versions, and expertise, as developing for the macOS platform requires knowledge of Apple's development frameworks and guidelines. | When developing software to support multiple types of clients on Linux, key considerations include cost, which is typically lower due to the availability of free and open-source tools, and expertise, as Linux development often requires specialized knowledge and skills. Additionally, time considerations may vary depending on the complexity of the target Linux distributions and the need for compatibility testing across different versions. | When developing software to support multiple types of clients on Windows, cost considerations involve understanding licensing fees for tools and frameworks used, as well as ensuring adequate hardware and infrastructure to accommodate the diverse client requirements. | When developing software to support multiple types of clients on mobile devices, two key considerations are time and expertise. Time is crucial as developing and testing for multiple platforms (e.g., iOS and Android) may require additional effort, resources, and coordination. Expertise is also critical, as each platform has its own unique development frameworks, programming languages, and design guidelines that developers need to be proficient in to ensure optimal app performance and user experience. |
| **Development Tools** | When building software for deployment on macOS, relevant programming languages include Objective-C and Swift. Objective-C has been traditionally used for macOS app development, while Swift is a modern language promoted by Apple. Commonly used IDEs for macOS development include Xcode, Apple's official integrated development environment, which provides a comprehensive set of tools, editors, and debugging capabilities for building macOS applications.  Development Tool/IDE -XCode  Programming Language - Swift (or Objective C) | When building software for deployment on Linux, relevant programming languages include C, C++, Python, and JavaScript, among others. These languages are commonly used for developing Linux applications due to their versatility and compatibility. Popular IDEs for Linux development include Visual Studio Code, Eclipse,  Development Tool/IDE - Eclipse is the most popular.  Programming Language - C is the default language. | When building software for deployment on Windows, relevant programming languages include C#, C++, Python, etc. These languages are widely used for Windows application development and provide access to the extensive Windows API. A popular IDE for Windows development is Visual Studio which offer robust coding environments with features such as debugging, code completion, and project management tools.  Development Tool/IDE - Visual Studio  Programming Language – Visual Basic | When building software for deployment on mobile devices, relevant programming languages include Swift for iOS development and Java or Kotlin for Android development. These languages are widely used and supported by their respective platforms. Popular IDEs for mobile app development include XCode for iOS development and Android Studio for Android development, providing comprehensive tools, emulators, and debugging capabilities.  Android:  Development Environment - Android Studio   Android: Programming Languages - Java, Kotlin.  IOS: Development Tool/IDE -XCode  Programming Language - Swift |

## Recommendations

1. **Operating Platform**: For the operating platform recommendation, I would suggest that The Gaming Room starts by expanding Draw It or Lose It to Windows devices. This choice offers several advantages, including a wide range of available software and tools, making it easier to find suitable development resources. Windows devices also tend to have a larger user base, increasing the potential reach of the game. Additionally, Windows provides a variety of Integrated Development Environments (IDEs) to choose from, ensuring flexibility and options for the development team. Lastly, Windows devices are often more accessible in terms of expertise and cost, making it a practical choice for getting projects up and running efficiently.
2. **Operating Systems Architectures**: Windows operating platform architectures offer a comprehensive set of services utilized by all Windows-based applications. These services enable applications to present a user-friendly Graphical User Interface (GUI) while seamlessly accessing system resources and more. Windows architectures also encompass functionalities related to graphics and multimedia, messaging, and web services. These services can be utilized either through user accounts or dedicated servers, providing flexibility in deployment and usage scenarios.
3. **Storage Management**: An appropriate storage management system for the recommended Windows operating platform is the built-in feature called Storage Sense in Windows 10. Storage Sense offers convenient file management capabilities, allowing users to effectively manage the storage space on their hard drives. With this feature, users can easily identify and assess the space occupied by files, enabling efficient organization and cleanup.
4. **Memory Management**: The recommended operating platform utilizes memory management techniques to support the Draw It or Lose It software. By creating a dedicated database or library to store a large number of pictures, the memory allocation system enables efficient storage of these images outside of the default picture folder. This approach ensures that the entire project, including the pictures, remains organized and securely stored in a designated area on the computer. The memory management system also facilitates seamless integration with the integrated development environment (IDE), allowing for smooth file handling and manipulation during the game creation process.
5. **Distributed Systems and Networks**: To enable communication between various platforms for Draw It or Lose It, a distributed software approach can be employed, leveraging the network that connects the devices. In this context, considering the dependencies between the components within the distributed systems and networks, it is essential to address connectivity issues and potential outages. To mitigate potential problems such as outages or connectivity issues, The Gaming Room should ensure that their servers are robust enough to support a large volume of players.
6. **Security**: Windows operating system includes features such as real-time malware and virus scanning, as well as protection against security threats. These built-in security capabilities provide a basic level of protection for the system and user data. However, for more comprehensive security, it is recommended to utilize additional security software and tools from reputable third-party sources. These solutions can provide advanced threat detection, encryption, secure data transmission, and other security features that go beyond the built-in protections. By incorporating external security solutions, The Gaming Room can strengthen the protection of user information, including sensitive data, login credentials, and personal details, both on the local machine and during data transmission between platforms. Regular updates and patches should also be applied to ensure the security software remains up to date and capable of detecting and mitigating the latest threats effectively.