

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 | 05/26/24 | Jennifer Rebella | Added to Executive Summary, Design Constraints, Domain Model, Evaluation, and Recommendations sections. |
| 1.1 | 06/04/24 | Jennifer Rebella | Revised Evaluation and Recommendations sections. |
| 1.2 | 06/21/24 | Jennifer Rebella | Revised Recommendations section. |

**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)

Our client, The Gaming Room, wants a web-based game for their game on Android software, Draw it or Lose it, that works on multiple software platforms. The game's object is to the puzzle by the images rendered from the applications the game has four roles; each row lasts a minute, and if one team does not guess the answer, the other can steal within 15 seconds.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

* The game must run on multiple software platforms
* Check if the team and the players name are not in uses
* Only one instance of a game can run at a time

## [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 diagram below shows the Draw It or Lose It application system classes. The class Entity is a base superclass for the application entities. All of the subclasses share attributes. The subclasses Game, Player, and Team extend to the Entity class. These classes are the main entities of the game application. The class GameService has a composition relationship with the class Game; this manages the game instances and holds the references to a multiple of the game objects. The Game class has a composition relationship with the class Team, and the Team class has a composition relationship with the class Player. The class ProgramDriver contains the main function that runs the game. This is like an entry point for the application. In the class ProgramDriver, the singleton instance of GameService is created so that there will only be one game instance at a time. ProgramDriver is also responsible for adding games, teams, and players by using GameService instance. The ProgramDriver is dependent on the SingletonTester. The UML diagram shows many object-oriented programming (OOP) principles, such as, abstraction, inheritance, and encapsulation. The diagram shows the inheritance relationship between the superclass Entity and all of its subclasses, Player, Team, and Game. Since the subclasses inherit from the superclass Entity, they can use attributes and behaviors from the superclass. This helps reduce code duplication. The class of GameService encapsulates its attributes, so it only provides methods to interact with the data. This ensures data privacy by allowing only one instance to be created.

**"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** | Mac has good accessibility, an easy-to-use GUI, and flexible terminal commands. However, it can be very expensive for server use, and larger companies avoid using Mac servers for this reason. | Since Linux is open source, it is very cost-friendly. The command shell is easy for server configuration and accessibility. Linux is easily customizable, which is great for security. Since large companies use it, Linux supports web hosting well. However, using Linux has a big learning curve and can cause issues while learning the operating system. | The server side of Windows is expensive, but it has a User-Friendly GUI and a Command prompt. Windows is widely used, so there is little to no learning curve when using this operating system. Windows has many options for web hosting and lots of third-party support. | Specifications depend on each device, making server-side work trickier. It is possible to use a mobile device on the server side, and it can be useful in some cases, but it is mostly impractical. |
| **Client Side** | Expensive for the end users. Users must have a moderate amount of understanding of the OS is important to use the OS | Users must have extensive expertise to understand the OS. | Easy to operate the OS and to set up. Little expertise is needed. | Flexible for the clients. Lest cost for the end users |
| **Development Tools** | HTML, CSS, and JavaScript are the languages used. Libraries help develop front-end services. Development tools include GitHub and coding environments. | HTML, CSS, and JavaScript are the languages used. Libraries help develop front-end services. Linux systems include Ruby and Python. | Languages used are HTML, CSS, and JavaScript. Libraries help the development of front-end services. Developer tools include coding environments and GitHub. | HTML, CSS, and JavaScript are the languages used. Libraries help develop front-end services. HTML, Python, C++, and PHP are the programming languages for mobile devices. |

## 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 for development as it is the most used platform. It will not have as severer of a learning curve like that associated with Linux, allowing the project to start immediately. Since the application will be web-based, it would be good to utilize Windows backend features. While it is more expensive than Linux, Windows is still cheaper than Mac when it comes to server costs, and the platform is more flexible. It has many options for web hosting and great third-party support.
2. **Operating Systems Architectures**: Windows has a simple GUI for easy use. Windows also allows a wide range of IDEs that can be used to help develop applications; some of the most used ones are Eclipse and Visual Studio. I would recommend having the team use Eclipse so the team is all working with the same IDE.
3. **Storage Management**: For storage management, I recommend the Windows OS manages storage through configuration, which makes it easy for those working with the back-end to navigate through the settings. Though using Windows cloud servers will add more to the cost, it is a good investment since this space can be expanded if the client needs it. Using cloud storage will also help with save physical storage space.
4. **Memory Management**: With Windows, there are two memory management options: physical and virtual memory. Virtual memory would work better for this project since it allows large programs to be handled effectively. An advantage of virtual memory is that, as opposed to physical memory, it has memory protection and helps extend the use of physical memory.
5. **Distributed Systems and Networks**: Common issues with distributed systems and networks include lag, queuing problems, and overloaded servers, which can be solved with a well-communicating team. I recommend using agile and scrum practices like the daily scrum meeting so that all team members are on the same page.
6. **Security**: For security, it is important to have protocols in place and use available features, such as antivirus programs and VPNs. Routine security checks should ensure that users' data is safe and that unauthorized access is not allowed. Data should be accessed on a as needed basis and users should only have access to what they need to complete the project.