

Zach Irvin

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_heading=h.gjdgxs)

[**Table of Contents 2**](#_heading=h.30j0zll)

[**Document Revision History 2**](#_heading=h.3znysh7)

[**Executive Summary 3**](#_heading=h.2et92p0)

[**Requirements 3**](#_heading=h.tyjcwt)

[**Design Constraints 3**](#_heading=h.1t3h5sf)

[**System Architecture View 3**](#_heading=h.4d34og8)

[**Domain Model 3**](#_heading=h.2s8eyo1)

[**Evaluation 4**](#_heading=h.17dp8vu)

[**Recommendations 5**](#_heading=h.26in1rg)

## [Document Revision History](#_heading=h.lnxbz9)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.2 | 06/022/2024 | Zachery Irvin | Updated recommendations for the gaming room application. |

**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](#_heading=h.35nkun2)

The Gaming Room is creating a web-based game called "Draw It or Lose It," which is currently available only on Android. The game features several teams, each made up of multiple players, competing in four rounds, each lasting one minute. In each round, a team tries to guess a picture from a library within the given time. If they don't succeed, members of the opposing teams get 15 seconds each to provide an answer. The goal is to make the game accessible across various platforms.

## 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*

## [Design Constraints](#_heading=h.1ksv4uv)

- Involves multiple teams, each with several members.

- Requires unique game and team names to ensure availability.

- Supports only one active game instance at a time.

- Needs to be compatible across multiple platforms through the use of the web:

- Android

- Windows

- Linux

- Apple devices

## [System Architecture View](#_heading=h.44sinio)

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](#_heading=h.2jxsxqh)

The following shows the relationships among the Game, Team, and Player classes within a software model, with all these classes inheriting from a common superclass, the Entity class. Inheritance is illustrated using Unified Modeling Language (UML), showing that each class shares common attributes such as "name" and "id". The relationships are described as "has a" types, signifying aggregation in UML. This indicates that an instance of one class (e.g., GameService) contains references to instances of another class (e.g., Games), creating a hierarchical and interconnected structure among the classes.

**"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](#_heading=h.z337ya)

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** | Not used as often for server development .  Does provide a macOS server app that can be added to the standard macOS to provide the server functions.  Cost has the possibility of being substantial as macOS run only on  Apple hardware. | Used often for web servers. Provides good security and performance as well as having various versions of the OS being opened source  Open source versions have zero licensing cost for their use. Red Hat Distribution requires a subscription of 349 per year. | Offers a very well rounded operating system for server hosting.  The cost for licensing depends on the needs of the client. The cost can vary from 900 a year to upwards of 6000 a year. | Limited capabilities and not well suited for server side requirements. |
| **Client Side** | Not used for games as much as Windows, however does provide excellent software for more creative endeavors such as art and design.  There isn’t any cost for licensing for the software, however as it must be an apple device, the cost of the hardware is more expensive. | Not used for gaming as much as windows.  There are several distributions that are open source and therefore free to use. | Widely used by gaming enthusiasts and is the most viable option on the client side.  Licensing cost can be from 139 to 199 depending on the version of windows that one chooses to install. | As the name implies, it provides for mobility and updates from anywhere. Needs some expertise for implementation. |
| **Development Tools** | Supports the web programming languages of html, css, and javascript. Tools include general programming languages as well as multiple ide’s and text editors. | Supports web programming languages. Has multiple text editors and ide’s that make for a productive work environment. | Supports web programming languages as well as the Visual Studio ide. | Has mobile specific tools. Most notable is Swift, Java and Kotline. |

## 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**: It is advisable for The Gaming Room to initiate their project on Windows devices, as these platforms offer a wide range of software options and require minimal expertise and investment to begin. Additionally, there is an abundance of Integrated Development Environments (IDEs) available, ensuring no shortage of development tools.
2. **Operating Systems Architectures**: Windows provides various services that are essential for all Windows-based applications. These services enable the creation of Graphical User Interfaces (GUIs) and grant access to system resources, among other functionalities. Windows applications also incorporate features related to graphics, multimedia, messaging, and web services, accessible either through a user account or via a dedicated server.
3. **Storage Management**: In Windows 11, users can benefit from the convenient tool called Storage Sense, which provides comprehensive management of files stored on the hard drive. This feature allows users to inspect and organize their files efficiently while keeping track of storage usage. Furthermore, Windows 11 offers the flexibility to designate specific storage locations for applications, enhancing accessibility and organization. Windows supports seamless integration with cloud storage solutions for data backup and storage. With its integrated storage system, it simplifies file creation and organization for substantial projects, ensuring their security and safeguarding against accidental deletion.
4. **Memory Management**: In Windows 11, memory management techniques are crucial for efficient resource utilization in the "Draw It or Lose It" software. The operating system employs virtual memory to extend RAM capacity when needed, dynamically allocates memory for various components such as variables and graphical buffers, and manages memory deallocation through garbage collection. Efficient buffer management ensures optimal storage of temporary data, while memory paging techniques enhance memory access by loading necessary pages into physical memory and swapping out less frequently accessed ones to disk. Memory protection mechanisms safeguard against unauthorized access and security vulnerabilities, contributing to the stability and reliability of the software on the Windows 11 platform.
5. **Distributed Systems and Networks**: This system would involve interconnected components like a central server, client applications across various platforms, and potentially cloud-based services for storage and synchronization, relying on network protocols like TCP/IP or HTTP(S). Dependencies include network connectivity, potential outages, latency, and security. To address these, mechanisms for detecting and handling connectivity issues, redundancy and failover strategies for outages, optimization of data transfer protocols to mitigate latency, and robust security measures including encryption, authentication, and access controls must be implemented. Overall, careful consideration of these dependencies is essential to ensure seamless communication and reliability across the distributed system and network.
6. **Security**: Due to the high number of threats out there for windows operating systems, it is highly recommended to use additional security solutions including mcafee and / or Norton on top of the pre - installed security software that comes with the windows operating system. These can provide better detection and elimination of security threats. A positive aspect of the Windows operating system also included automatic security updates that are downloaded and installed as new threats and vulnerabilities are assessed. A combination of the security that comes with windows as well as third party software will help ensure a secure platform for the users thereby protecting their information and data as well as the company’s.