

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_heading=h.xb80cql7rng6)

[**Table of Contents 2**](#_heading=h.tsdx14drd49a)

[**Document Revision History 2**](#_heading=h.4m6afmllfdx1)

[**Executive Summary 3**](#_heading=h.d5htsz1swlh6)

[**Requirements 3**](#_heading=h.3r0gyq4ctcts)

[**Design Constraints 3**](#_heading=h.8s11ohyhk8bz)

[**System Architecture View 3**](#_heading=h.htymc9r1oycw)

[**Domain Model 3**](#_heading=h.wz9imm4rzhex)

[**Evaluation 4**](#_heading=h.bk39133vu3oo)

[**Recommendations 5**](#_heading=h.5vhp919ewtk9)

## [Document Revision History](#_heading=h.l8gixvxdnrp1)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/20/25 | Clara H. Abraham | The executive summary, requirements, and design constraints were added. |
| 1.0 | 05/21/25 | Clara H. Abraham | The system architecture view (topology) was added. |
| 1.0 | 05/23/25 | Clara H. Abraham | Added the domain model description and evaluation. |
| 1.0 | 05/24/25 | Clara H. Abraham | Recommendations were added. |

**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.ji2bbted1uk0)

Our new client, The Gaming Room, wants to develop a web-based game that has cross-platform compatibility based on their game, Draw It or Lose It, which is currently only available in an Android application.

## Requirements

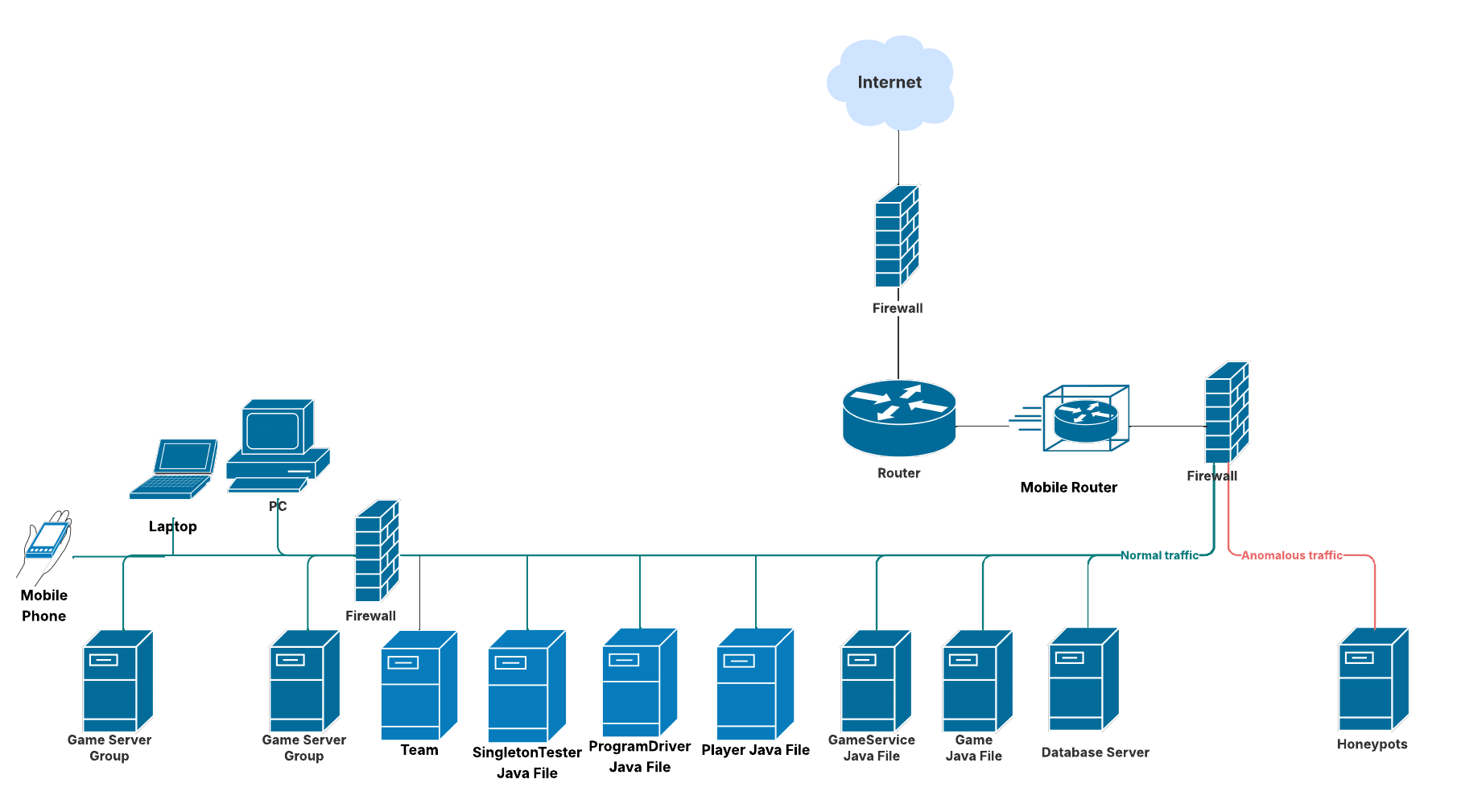
* *The game must use the internet or World Wide Web.*
* *The game must have the ability to function across different platforms and operating systems.*

## [Design Constraints](#_heading=h.av2d6drgw56l)

* Internet connectivity: Web-based games are reliant on a constant internet connection and an unstable network connection can disrupt gameplay. Web-based games cannot be played offline.
* Varying browser sizes: Web-based gameplay can be affected by different browser layouts since the screen sizes are different for computers, laptops, and mobile phones (the main concern here is the game is bigger that the size of the device’s screen/browser, therefore the user cannot play the game correctly).
* Testing and quality assurance: Testing a cross-platform game can require repetitive testing on various platforms such as Windows, macOS, Linux, Android, and IOS to ensure complete functionality. Some examples of devices that would be used for testing the game are Windows computer, IOS computer, macOS laptop, Linux laptop, Android mobile phone, and IOS mobile phone.

## [System Architecture View](#_heading=h.dmiuclt1sb9y)

A logical topology of the communication and storage aspects is provided to understand the overall architecture.



## [Domain Model](#_heading=h.upozz7gmn75o)

The class named, Entity, is the parent class with Game, Team, and Player as its child classes. This is because Game, Team, and Player inherit from Entity. Game and the class named, GameService, have an association relationship meaning that a property of a class holds a reference to an instance of another class. ProgramDriver and SingletonTester have an implementation relationship because the interface has a relationship with the implementation class.

**"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.1d4wt6pyaihd)

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** | Pros: Built in program called BootCamp that allows you to install windows, Linux, and any other operating system.  Cons: Can’t play high-end games. | Pros: Less susceptible to viruses, malware, and other security threats.  Cons: Complex for beginners. | Pros: Has a troubleshooting system feature.  Cons: Has too many updates. | Pros: Platform is cost effective.  Cons: Has poor security management. |
| **Client Side** | Cost: Expensive for users.  Time: Moderate amount of time.  Expertise: It will take less time if the user is mac experienced and more time if the user is not mac experienced. | Cost: Feasible  Time: Needs more time compared to other platforms  Expertise: Even if the user is Linux experienced, it would take a lot of time because Linux is complex. | Cost: Moderate level of expense  Time: Less time is required because the setup is easy  Expertise: Requires a minimum level of experience since it is easy to operate. | Cost: Depends on the device  Time: Needs more time than other platforms  Expertise: Easy to use and understand regardless of experience with mobile devices. |
| **Development Tools** | Visual Studio, Python, Java, C++, IntelliJ IDEA. | Visual Studio, Python, Java, C++, IntelliJ IDEA. | Visual Studio, Python, Java, C++, IntelliJ IDEA. | Visual Studio, Python, Java, C++, IntelliJ IDEA. |

## 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**: Windows operating platform is recommended because it is highly secure without long load times and is not expensive.
2. **Operating Systems Architectures**: “Operating system kernel” is responsible for taking system requests from system services. The kernel translates those requests into instructions for the computer hardware.
3. **Storage Management**: Windows has an optional feature called Windows Storage Management that includes the capability called Storage Optimization which optimizes storage by freeing up space in your hard drive.
4. **Memory Management**: Windows has built in features to manage RAM efficiently.
5. **Distributed Systems and Networks**: The communication between the variety of devices is going to be efficient. The efficiency of communication depends on the following factors: network connectivity and cellular data consistency.
6. **Security**: Windows has built in features called Microsoft Defender Antivirus, Secure Boot, and TPM 2.0 Requirement that can provide protection against malware, ensures trusted software is loaded during the boot process, and safeguard sensitive data.