

The Gaming Room

# **CS 230 Project Software Design Template**

Version 3.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/23/2022 | Afshin Ahvazi |  |
| 2.0 | 04/06/2022 | Afshin Ahvazi | Added evaluation for operating systems |
| 3.0 | 04/20/2022 | Afshin Ahvazi | Added a recommendation page |

**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* will provide consultation to *The Gaming Room* regarding transition of their game application *Draw It or Lose It*, which is currently only available for android devices. Upon completion of the project, the game will be web-based and accessible by all platforms. This document explores different aspects of the project such as design constraints, the UML overview of the program, and a comparison of different operating systems to run the application, in the sections that follow.

## [Design Constraints](#_2et92p0)

Since the game will be hosted on the web, clients will need access to the internet. The program must implement data monitoring to ensure conservative use of data so that users on mobile or otherwise limited data plans will have the best experience.

## [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)

Below is the UML diagram for the program. The main class, GameService, is responsible for keeping track of the Games. Each Game consists of Teams, and each Team contains Players. GameService implements the singleton design pattern to ensure only one instance of the GameService class may exist at a time. The classes Game, Team, and Player inherit common traits (fields and functions) from the abstract class Entity. These classes employ iterators to loop around their internal data structures.

**"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** | MacOS is not as common as Windows and Linux. Support for web development is not prevalent. | Linux is free and open source. It is  safe and secure. It’s also light and fast, with many distributions specialized for different tasks. | Windows is expensive. This OS is more prone to attacks and viruses. However, it has native support for major frameworks like .NET. | Hardware wise, mobile devices are the most expensive, and offer little processing power. There is minimal support for hosting web servers. |
| **Client Side** | Compared to other OS, Mac requires moderate expertise. Its Unix terminal provides relative ease of use, and Apple offers support. Many Apple devices run the iOS, therefore we can target many devices as clients. | Linux requires the most expertise and provides the most ease of use for developers with the knowledge. Since it is open source, it is the best option to learn and take advantage of the operating system. | Most common system that is widely known of. It is by far the easiest to use, with user-friendly GUI. Cost is high and similar to MacOS | Mobile devices cover a variety of operating systems. From Android and Linux to iOS and Windows. This broad range makes it difficult to cover all platforms as clients. |
| **Development Tools** | As mentioned, Apple offers developer tools. There limited IDEs. Examples are IntelliJ (free community edition) | Linux is open source; there are myriads of IDEs and SDKs available for this platform, almost all of which are free and open source. | Windows is known for program availability; it has the most tools available for any line of programing. | Since mobile devices are the most diverse in this group, there are many tools to select from. Availability consists of the previous 3 systems. |

## 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**:

Most webservers are hosted on Linux. As discussed previously, Linux is free, secure, and fast. It is the most optimal operating for this project.

1. **Operating Systems Architectures**:

The Linux architecture consists of 3 main components: the kernel (including the kernel modules and system libraries), the hardware (external/internal physical devices), and the shell interface. The kernel acts as an intermediary between the software and the hardware such as the processor, memory, and other I/O devices. Through the shell, the user can interact with the kernel and therefore the hardware. It is an API by the operating system.

1. **Memory Management**: <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.>

The Linux kernel must ensure that memory is allocated and deallocated so that new processes and applications can even begin to run. In addition, different processes must not interfere with each other. Since memory, especially continuous available blocks of memory, are scarce, the operating system implements additional techniques like virtual memory management. The kernel would temporarily store unused memory into disk storage, and constantly swaps information in between memory and storage, thus making room for new processes.

1. **Distributed Systems and Networks**: <Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).>

A distributed system is one where multiple systems compute different components of an application and transmit the results in between each other. For our application *Draw it or Lose it*, the necessary amount of computing power doesn’t extend a distributed system. A single server would be sufficient where clients connect to via internet protocols.

1. **Security**: <Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.>

Linux is as secure as it can be. Firstly, users accounts are separated by username and password. The Linux permission system controls user access to directories and available administrative commands. In addition, since Linux is open source, its source code is inspected by myriads of developers. Even if an exploit is found in the operating system, it will not go unsolved for too long.