

*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 |
| --- | --- | --- | --- |
| 3.0 | 6/15/2023 | Ryan LeChien | Added **Recommendations** section chart |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room has contracted Creative Technology Solutions (CTS) as a technology consultant for the development of a web-based version of their gaming app, *Draw It or Lose It*. Adherent to various criteria and software requirements,

## [Design Constraints](#_2et92p0)

Design constraints for the application include: the ability to have one or more teams involved, each team will have multiple players assigned to it; game and team names must be unique to allow users to check whether a name is in use when choosing a team name; and only one instance of the game can exist in memory at any given time.

## [Domain Model](#_8h2ehzxfam4o)

For this application, there are multiple classes which exhibit traits of an Entity.

**"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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Limited amounts of inferior software for running a server on macOS;  cost of macOS hardware and licensing may be higher compared to other hosting options; easy compatibility with other Apple devices, but may have compatibility issues with Android-based app | Various Linux packages, such as Ubuntu Server and Debian, with great community documentation and support; other operating platforms may have limited support or require additional configuration | User-friendly interface and popular development environment; extensive support for wide range of software frameworks, including Android operating systems; Windows Server 2022 is a flagship server platform, but may have higher licensing costs, especially for large deployments | Inappropriate to host a server on a mobile device |
| **Client Side** | Limited number of developers with cross-platform expertise, resulting in higher cost; incompatibilities within this operating platform may restrict cross-platform interoperability | Through Linux development is more uncommon, access to .NET Framework and other Windows features promotes large number of developers | Ample Windows/.NET developers; access to common interfaces and cross-platform web browsers, ensuring access; WORA languages | Large number of mobile developers for both Android and iOS, keeping labor and resource costs low; may have inexperienced workforce |
| **Development Tools** | The macOS Xcode IDE supports programming languages such as Swift and Objective-C for Apple platform (iOS) development | Linux-supported IDE’s include Visual Studio and Eclipse, which support popular languages such as Python, Java, HTML, and JavaScript, allowing flexible design and functionality | Available IDE’s include Visual Studio, Eclipse, and NetBeans. Access to all programming languages, primarily Python, Java, HTML, JavaScript, and the .NET Framework, allowing flexible front- and back-end design and functionality | Swift is an appropriate language for iOS devices, and Java works well for Android devices; there is no sufficient integrated development environment 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:** The most appropriate operating platform available for hosting the game application is the latest Debain distribution, Debian 12. This operating platform is continually updated and is supported by a large amount of documentation and community support. Furthermore, Debian is feature-rich freeware, shipping with multiple desktop environments and server platforms. This variety of operating platforms within Debian will give the developers flexibility in their approach to availability and accessibility of the game across distributed network servers.

**Operating Systems Architectures:** Debian is a Linux-based operating platform. As stated in the latest release post by the Debian developers, the latest release supports the following CPU architectures: 32-bit PC (i386) and 64-bit PC (amd64); 64-bit ARM (arm64); ARM EABI (armel); ARMv7 (EABI hard-float ABI, armhf); little-endian MIPS (mipsel); 64-bit little-endian MIPS (mips64el); 64-bit little-endian PowerPC (ppc64el); and IBM System z (s390x). Furthermore, the latest release contains 64,419 packages, along with the update of 43,254 packages. These hardware and software specifications and libraries make Debian a scalable and distributable operating platform.

**Storage Management:** The game application will be best stored between distributed network servers and the game user’s local storage. According to the game’s specifications, the application will utilize 200 high-definition image files to choose from, each one approximately 8 megabytes in size. To reduce the bottleneck of unpredictable network latency, these files will be stored on the game user’s device. However, to facilitate a high quality of service for over-the-network play between users, some pertinent application files will be stored on the servers.

**Memory Management:** The game application will be most efficient when developed in the Java programming language. The Java Virtual Machine is a virtual environment that dynamically handles low-level memory operations. To reduce the cost of expertise in memory-efficient programming, the JVM can be utilized to more safely and more cost-effectively manage memory resources, both on the network servers and the users’ local devices.

**Distributed Systems and Networks:** To facilitate communication between various platforms, the distributed software will define an interface which will enable interoperability between the users’ local devices. Additionally, the game servers will be distributed across wide geographic areas to maximize connectivity and minimize latency. By ensuring a common interface and protocols between local users and distributed game servers, a great deal of flexibility can be achieved with regard to scaling the application.

**Security:** To protect user information on and between various platforms, all network traffic will be encrypted using the SHA-256 hashing algorithm. Furthermore, the game application will require the use of a single sign-on linked to a general game account. This form of authentication offers not only security but also the benefit of users being able to access their game data across various platforms.

Reference

“Debian 12 Bookworm.” Debian, 10 June 2023, www.debian.org/News/2023/20230610.en.html.