

Draw It or Lose It

# **Software Design Document for The Gaming Room**

Version 3.0

## Table of Contents

[**CS 230 Project One Software Design Document** 1](#_Toc110932456)

[Table of Contents 2](#_Toc110932457)

[Document Revision History 2](#_Toc110932458)

[Executive Summary 3](#_Toc110932459)

[Design Constraints 3](#_Toc110932460)

[System Architecture View 3](#_Toc110932461)

[Domain Model 3](#_Toc110932462)

[Evaluation 4](#_Toc110932463)

[Recommendations 6](#_Toc110932464)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/24/2022 | Amanda Purnhagen | Version 1.0 was the initial software design document for The Gaming Room’s new web-based application, Draw It or Lose It. The document contained the information delineated in the Table of Contents seen above. The document was awaiting approval from management. |
| 2.0 | 08/01/2022 | Amanda Purnhagen | Version 2.0 was the second version of the software design document for The Gaming Room’s new web-based application, Draw It or Lose It. This version saw an update in the “Evaluation” section on page 4. The document was proofread and optimized for readability. This document was still awaiting approval from management. |
| 3.0 | 08/09/2022 | Amanda Purnhagen | This is version 3.0 of the software design document for The Gaming Room’s new web-based application, Draw It or Lose It. This version includes a substantial update to the “Recommendations” section on page 6. The document has been proofread and optimized for readability. This document is still awaiting approval from management. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room (TGR) has a new contract with our company, Creative Technology Solutions (CTS). They have delegated to us the task of developing a multiplatform web-based game, Draw It or Lose It. Currently, this game only functions on Android devices. The application must get data from a cloud host and various APIs. Currently, CTS’s servers cannot take on the size of the library of possible rendered drawings. The download page for the application must stipulate what the minimum hardware and network requirements are for Android, iPhone, PC, Mac, Linux, etc. devices.

The Draw It or Lose It mobile project must adhere to TGR’s schedule and budget. CTS has assembled a team of developers experienced with writing in the Java programming language. Alice will not work on the project because she is currently assigned to the Park Station Manufacturing project. Bob must work on the project because he is a new hire in need of training.

The creation of graphics and sounds for the application has been outsourced to Lepage Productions (LP), with whom we have a contract. For legal reasons, all graphics and sounds must come from LP, and all relevant data must be hosted by Linode cloud hosting services.

## [Design Constraints](#_2et92p0)

The Draw It or Lose It software must be a web-based application that is cross-compatible. Therefore, CTS has decided to write the application in the Java programming language. We must adhere to our contract with Linode cloud. They will host our large stock drawing library. These stock drawings and any sounds for the application must come from LP. Our employee, Alice, cannot work on the project, while Bob must train on this project.

## [System Architecture View](#_ilbxbyevv6b6)

This is a reminder that, in the next revision of this software design document, CTS must describe the system and subsystem architecture present in the application, including physical components or tiers, required for this project. 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 UML class diagram provided below displays how the various Draw It or Lose It Java application classes relate to each other. For each game service, there can be one to many games. For each game, there can be one to many teams. For each team, there can be one to many players. The Game, Team, and Player classes all inherit from the Entity class which informs its subclasses which attributes and functions they need to have.

The main method is contained in the ProgramDriver class which uses the SingletonTester class to ensure that there is only one instance of any existing game, team, or player at any time. To adhere to the singleton design pattern, the constructors for the Game, Team, and Player classes have been made private. Additionally, the iterator design pattern is used when adding potential new games, teams, and players to ensure they do not already exist.

**"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** | Hosting the web-based software on Mac could give the developers quality of life through the user-friendly interface, but the hardware and OS are not very flexible nor lightweight. However, Mac hardware is relatively powerful. CTS recommends that TGR considers the price and benefits of hosting the relevant data using Sync.com. | Hosting the web-based software on Linux would provide a flexible, powerful, and easy to orchestrate server-side system. However, there is less technical support, and some common software is not as readily available. CTS recommends that TGR considers the price and benefits of hosting the relevant data using Linode. | Hosting the web-based software on Windows would provide a strong closed-source OS, but there would be very little to no customization. It could grant ease of mind through easy-to-access technical support. CTS recommends that TGR considers the price and benefits of hosting the relevant data using Amazon Web Service (AWS). | CTS does not recommend hosting the web-based software on mobile devices. |

| **Client Side** | Clients who choose to use the web-based application on Mac will have a user-friendly experience. Mac device users are supported since the application is written in Java and is accessible in a web browser. | Clients who choose to use the web-based application on Linux will have a customizable and flexible experience with more of a learning curve. Linux device users are supported since the application is written in Java and is accessible in a web browser. | Clients who choose to use the web-based application on Windows will have a user-friendly experience. Windows device users are supported since the application is written in Java and is accessible in a web browser. | Clients who choose to use the web-based application on a mobile device will have a user-friendly experience. Mobile device users are supported since the application is written in Java and is accessible in a web browser. |
| --- | --- | --- | --- | --- |

| **Development Tools** | Eclipse will be used to write the web-based application in Java which will ensure that the software is cross-compatible. We will need our development team to be proficient with the MacOS operating system. If not, we will need to allocate some time and resources to training our development team to work efficiently with MacOS. | Eclipse will be used to write the web-based application in Java which will ensure that the software is cross-compatible. We will need our development team to be proficient with the chosen Linux distribution. If not, we will need to allocate some time and resources to training our development team to work efficiently with Linux. | Eclipse will be used to write the web-based application in Java which will ensure that the software is cross-compatible. We will need our development team to be proficient with the Windows operating system. If not, we will need to allocate some time and resources to training our development team to work efficiently with Windows. | Eclipse will be used to write the web-based application in Java which will ensure that the software is cross-compatible. However, CTS does not recommend hosting the web-based software on mobile devices. |
| --- | --- | --- | --- | --- |

## Recommendations

1. **Operating Platform**: CTS officially recommends that TGR uses a Linux distribution of their choice for their operating platform for the Draw It or Lose It game. TGR should take advantage of the security, stability, flexibility, and scalability that comes with utilizing Linux.
2. **Operating Systems Architectures**: CTS officially recommends that TGR uses Linode, Terraform, and Docker to create their operating systems architectures. TGR can have their private server hosted by Linode, a reliable cloud-based hosting company that can cater to TGR’s specific memory, storage, and CPU needs. CTS recommends that TGR allocates the work of managing the Linode cloud infrastructure to Terraform. Terraform allows TGR to avoid configuring their data through hardware configuration and instead uses the cloud’s definition files. Docker will allow TGR to bundle its software for easy client consumption. This circumvents the need for clients to build the application’s entire infrastructure on their device.
3. **Storage Management**: CTS officially recommends that TGR uses the Linode cloud for their storage management needs. CTS also recommends the Alpine Linux distribution for its servers because it is lightweight and effective. A bonus is that Alpine is also very secure. CTS highly recommends that TGR uses Docker to provide “recipes” of files for devices to download and build the Draw It or Lose It application on various operating systems. TGR can save space by opting for a lightweight distribution and allowing Docker to do the heavy lifting. Additionally, CTS recommends that TGR opts for a serverless microservice architecture. This will streamline and minimalize TGR’s use of storage.
4. **Memory Management**: CTS officially recommends that, in order to use less memory, TGR does not use sessions for logins. Instead, they should use proper authorization and API endpoints which can be managed with Docker. This would help with memory because your server doesn’t have to manage many sessions simultaneously. Additionally, CTS recommends that TGR opts for a serverless microservice architecture. This will streamline and minimalize TGR’s use of memory.
5. **Distributed Systems and Networks**: CTS officially recommends that TGR picks reliable companies to provide the component services for the Draw It or Lose It game. Each company must meet or exceed industry-standard uptimes. CTS recommends that TGR uses Kubernetes to manage Docker in order to assist with load balancing and server--side supply of specific Docker images needed for download by clients. These are reputable and dependable services that will provide necessary data to the server and clients as necessary with exceptions including severe outages.
6. **Security**: In order to protect user information on various platforms, CTS officially recommends that TGR’s servers for Draw It or Lose It meet certain standards. CTS recommends that the login system has proper authorization and authentication protocols. We also recommend that TGR does not allow web directory indexing. The servers should also be able to load balance in order to prevent DDOS attacks. TGR should not allow users to inject JavaScript or HTML in input text boxes. Firewalls should be active at all times when and where appropriate.