

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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 |
| --- | --- | --- | --- |
| 1.0 | 1-25-23 | Bret Thex | Initial Design |
| 1.1 | 2-4-23 | Bret Thex | Updated Evaluation Information |
| 1.2 | 2-19-23 | Bret Thex | Updated Recommendations |

**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)

Development of a web-based version of “Draw It or Lose It” for The Gaming Room. The client currently has the app available only on the Andriod app store. The complete the full scope of this project Creative Technology Solutions will need to meet these requirements, providing a web-based environment and a web-based version of “Draw It or Lose It”. The game had 4 key requirements as follows, the game will support one or more teams, each team will have multiple players, team and player names will be unique, and only one instance will exist at any given time.

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

* A game will have 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.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

## [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 Entity class is the parent class to Game, Team, and Player classes. Game, Team, and Player all inherit

Entity’s required attributes. Game Service must only have one instance of each game running at any time. Each Game can only have one unique Team at any time and each Team can only have one of an individual Player at one time.

**"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** | **PROS:**   * Proven Security * Extensive Documentation for macOS   **CONS:**   * Highest Cost * Reliant on Apple For updates * Requires a Mac System with macOS | **PROS:**   * Open-source software available * Lowest hardware requirement * Most cost-effectiveoption * Proven Stability   **CONS:**   * Software Compatibility lacking when compared to windows * Steep Learning curve to operate | **PROS:**   * Largest Range of Software Support * Quick Updates * Hardware Customization * Extensive Documentation for Windows   **CONS:**   * Closed Sourced * Reliant on Microsoft for updates | * Can be used to host small websites * Can manage calls to database * Needs cloud based or physical server to handle anything large |
| **Client Side** | **PROS:**   * Wide Range of Compatible web browsers * Cross Platform Testing   **CONS:**   * Requires Apple product with MacOS | **PROS:**   * Wide Range of Compatible web browsers * Cross Platform Testing * Multiple different versions of open source software   **CONS:**   * Steep learning curve for operating system | **PROS:**   * Wide Range of Compatible web browsers * Relatively easy cross-platform testing   **CONS:**   * Cannot cross-platform test MacOS | * Longer Development Time * Difficult to other environmnets |
| **Development Tools** | **PROS:**   * Eclipse for Java * IntelliJ for Java * Can run Linux or Windows through a VM * Only way to deploy to MacOS   **CONS:**   * Application must be reviewed and approved by Apple. * $99 / year Apple Dev Program | **PROS:**   * No license cost * Open Source Community * Can Run Windows through VM * Eclipse for Java   **CONS:**   * Compatibility issues with some newer hardware. * Less Commercial Software Availability | **PROS:**   * Eclipse For Java * Visual Studio * Can run Linux on a VM * Deployment at any time * More documentationdue to large user base   **CONS:**   * Cannot Run macOS on a VM | * Possible to write code * Not recommended for development |

## 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**: based on our evaluations we recommend the Linux operating system.

Linux is open source reducing costs, also it supports various development tools, and it is flexible

and secure.

1. **Operating Systems Architectures**: Linux OS is constructed in a modular fashion. The heart, or kernel, of the OS, contains only core components while separate modules promote the dynamic expansion of services.
2. **Storage Management**: Draw It or Lose it is a web-based game and we recommend cloud storage and a physical backup. Cloud storage allows for changing storage needs without the cost of physical server hardware. It also decreases the cost for maintenance and security to almost zero as it will be taken care of by the could company. Having a physical backup reduces the chance of a total loss of all data.
3. **Memory Management**: Linux is a versatile and easy-to-customize operating system. For this

game, we will use Java as the main language for the backend development. Java does memory

management automatically. Java uses an automatic memory management system called

a garbage collector. Thus, we are not required to implement memory management logic in our

application.

1. **Distributed Systems and Networks**: Draw It or Lose It is web-based, thus as long as a browser is available on an OS, Draw It or Lose It will be available. We do not need to consider the user OS.
2. **Security**: Since we are using a cloud service for our server and storage we don't have to worry about server security. This will be taken care of by the cloud service provider.