

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 2.0

## Table of Contents

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

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

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

[**Executive Summary**](#_sbfa50wo7nsh)2

[**Design Constraints**](#_2et92p0)2

[**System Architecture View**](#_ilbxbyevv6b6)2

[**Domain Model**](#_8h2ehzxfam4o)2

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

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

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/27/2022 | Victor Vieira | Current Revision |
| 2.0 | 04/10/2022 | Victor Vieira | OS Evaluation/Domain addition |
| 3.0 | 4/23/2022 | Victor Vieira | Final Recommendation |

## [Executive Summary](#_sbfa50wo7nsh)

Creative Technology Solutions (CTS) is looking to design a browser version of the mobile app game Draw It or Lose It. The game requires one or more teams with multiple players, but no overlapping team names, along with only one game at a time.

## [Design Constraints](#_2et92p0)

In order to address these issues, we must look at the technology at hand.

* One or more teams involved
  + Create a list that can store the available team names, so that no overlap occurs
* Multiple players
  + Create a list that can store all the players unique names
* Only one instance of the game
  + This can be accomplished by writing a Singleton class to iterate over the lists and verify that no more than one instance has occurred

## [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)

The base class Entity has been built off the 3 classes Game, Team, and Player. These three classes attribute to one another primarily with the player’s ID and name, with a list of multiple team names and player names held in the Game and Team classes. The purpose of the Entity class is to hold one unique instance of the Team, Game and Player, it prohibits multiple uses or ambiguity within gameplay. The GameService class is built to hold the available fields to ensure no overlap may occur and output the game to the user.

**"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** | **Pro**: Highest level of security and encryption  **Con:** Many server tools now depreciated and built for small business and school  **Con:** Expensive | **Pro:** Security issues often non-issue  **Pro:** Minimal hardware requirements  **Pro:** Free/minimal cost  **Con:** Some distros do not have LTS | **Pro:** Up-to-date drivers  **Pro:** Automized system updates  **Con:** Vulnerable to cyber attacks  **Con:** High licensing costs | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.> |
| **Client Side** | **Pro:** Mac supports all major browsers.  Since the game will be web based, there are no cons in this list | **Pro:** Linux supports all major browsers except Safari officially, therefore there are no cons in this list | **Pro:** Windows supports all major browsers except for Safari, therefore there are no cons in this list. | **Pro:** Since the game is already built as an app, mobile development is taken care of. However, they would need to scale it for mobile browser.  **Con:**  Not enough resources/memory on mobile devices to run as standalone services. |
| **Development Tools** | **Pro:** Mac is known to be the more friendly option for developers given their superior hardware components and attractive GUI. They are faster at processing data than Linux or Windows.  **Con:** Initial price component may hinder scalability | **Pro:** Open source allowing for co-op work with safety net protection  **Pro:** Many built in tools along with customization  **Con:** Steep learning curve within the terminal  **Con:** Command line operation (perhaps not a con for many) | **Pro:** GUI for aesthetic ease of development  **Pro:** Long-term support and updates  **Con:** Script languages limited to VBScript, ASP.NET  **Con:** Server/Client needs to be installed and configured | **Con:** Developing on a mobile device is not advisable due to their specific device allocation. |

## 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**:

My recommended operating platform is Linux OS. The reason behind this recommendation is first, Linux is free and open source. It is a very customizable operating system that allows for fine tuning, as well as scalability.

1. **Operating Systems Architectures**:

The Linux Shell operates in a way that that acts as a messenger to the Kernel, or the underlying

Systems of the OS. The user will write the commands to the shell, which act as instructions to

the kernel. Although it may take a brief amount of time getting familiar with the Shell, the

tradeoff in price and customization pay off in the long term.

1. **Storage Management**:

My recommendation for storage management on Linux is the use Logical Volume Management (LVM). LVM manages storage volumes across different physical hard disks. This process allows for greater scalability for future games/databases.

1. **Memory Management**:

Linux has a few unique features that it utilizes to deal with memory management. First, virtual memory is used by the Kernel to allow programs to make a memory reservation. It will allocate a specific amount of memory that no other application can then use. However, if the actual amount of memory used begins to the approach the limit, there are parameters that are able to be put in place to kill the application such as /proc/sys/vm/overcommit\_memory with a specific value to provide direction.

1. **Distributed Systems and Networks**:

Since we are utilizing a client/server model, the client will be loading the game from our servers. Therefore, the applications can be modified to suit the specific needs of the client’s system. Since we are expecting to host thousands of players, a reliable server network must be in place to reduce bottlenecking and bandwidth issues.

1. **Security**:

Linux offers several security features and workarounds to deal with security issues. The first is to update often. This can be made automatic so the system to check on regular intervals. The systems engineer should create a privileged user to avoid using root, which grants complete access to the system. Although Linux is not prone to viruses, it is vulnerable to server attacks just as Windows or Mac. Fail2ban is an application that will review the system logs for anomalies, and if it finds recurring attacks, it will automatically adjust the firewall protocols to block the IP address from the service.