



Draw It or Lose It  
**CS 230 Project Software Design Template**  
Version 3.0

## Table of Contents

<b>CS 230 Project Software Design Template</b>	<b>1</b>
<b>Table of Contents</b>	<b>2</b>
<b>Document Revision History</b>	<b>2</b>
<b>Executive Summary</b>	<b>3</b>
<b>Design Constraints</b>	<b>3</b>
<b>System Architecture View</b>	<b>3</b>
<b>Domain Model</b>	<b>3</b>
<b>Evaluation</b>	<b>4</b>
<b>Recommendations</b>	<b>Error! Bookmark not defined.</b>

## Document Revision History

Version	Date	Author	Comments
1.0	07/14/2022	Marisa Kuyava	Added recommendations for The Gaming Room software
2.0	07/27/2022	Marisa Kuyava	Updated evaluation table
3.0	08/10/2022	Marisa Kuyava	Added Recommendations

## Executive Summary

The Gaming Room would like to take their current game Draw It or Lose It that is currently only available on Android and develop a web-based game that is available on multiple platforms.

## Design Constraints

- The game must have the ability to have one or more teams involved.
- Each team must be able to support being assigned multiple players.
- Needs to have the ability to check if team and game names are already in use as game and team names must be unique.
- Unique identifiers must be created for each instance of a game, team, and player to ensure only one instance of the game is in memory at any given time.

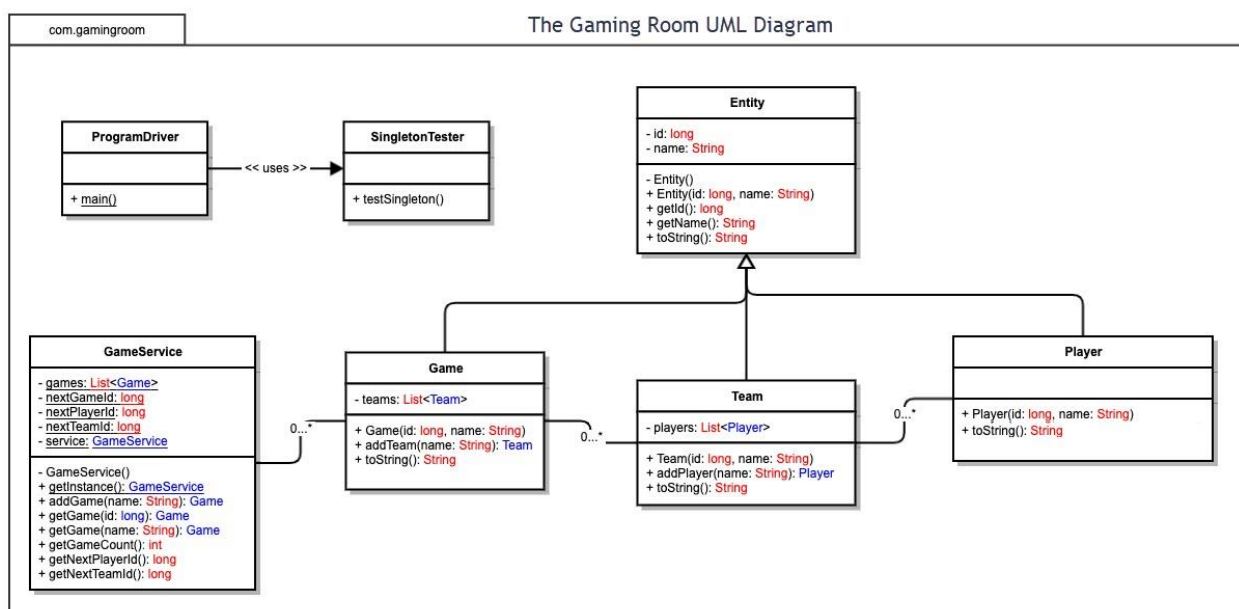
The constraints listed above are the client requested software requirements that need to be met for the game application. The Gaming Room is wanting to develop their current Android game to be available on multiple platforms, so developers will be needed to translate the current code to languages that will work on each platform.

## System Architecture View

N/A

## Domain Model

The Entity class is a Super/parent class and is inherited directly by Game, Team and Player classes. The GameServices, Game, Team and Player class all have a none to many association with each other. The ProgramDriver class holds our main method and uses the SingletonTester Class. Classes are set up to allow for one game to be played with multiple teams each with multiple players.



## Evaluation

<b>Development Requirements</b>	<b>Mac</b>	<b>Linux</b>	<b>Windows</b>	<b>Mobile Devices</b>
<b>Server Side</b>	Apple did have the Mac OS X server; however, it was discontinued in April of 2022. Based on my research I was unable to find a replacement.	There are multiple options available for Linux and most appear to be low cost.	There are multiple options available for Windows, however the cost for most appears to be a bit higher.	It is possible to host on a mobile device, however as they are much less powerful than a computer it likely would not be the best option.
<b>Client Side</b>	Cost would be higher for Mac because it is not open source. Time would be dependent on developer experience, a developer that has more expertise with working on MacOS would require less time than someone who is not familiar with the OS.	Cost should be low with Linux because it is open source. Time would be dependent on developer experience, a developer that has more expertise with working on Linux would require less time than someone who is not familiar with the OS.	Cost would be higher for Windows because it is not open source. Time would be dependent on developer experience, a developer that has more expertise with working on Windows would require less time than someone who is not familiar with the OS.	Cost would be higher because it would not be open source. Time would be dependent on developer experience, a developer that has more expertise with working on Mobile would require less time than someone who is not familiar with the OS.
<b>Development Tools</b>	Objective-C and Swift seem to be the most popular programming languages for MacOS. There are multiple IDEs available, the most popular seems to be Visual Studio.	C/C++, Java and Python are the more common languages for Linux. Atom and Eclipse are two of the more popular IDEs for Linux, however there are many options available.	JavaScript, Python and Java are the more common languages for windows. Visual Studio, Eclipse and PyCharm are common IDEs for Windows, but there are many options available.	Objective-C and Swift would be for iOS with Swift Playgrounds mobile IDE. Java and Kotlin would be for Android with Android Studio.

## 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:** Because Windows seems to be the most widely used operating platform, I would recommend Windows as be the best option.
2. **Operating Systems Architectures:** Would recommend utilizing 64-bit architecture as this would enable the system to utilize far more memory, in turn allowing applications to store and access more memory on a short-term basis.
3. **Storage Management:** Windows has storage management build in. Storage Sense and Disk Clean up can be used to free up space that is being used by unnecessary files, while Disk Management can be used to set up new drives, shrink partitions, extend volume, and modify drive letters.
4. **Memory Management:** Windows has memory management built in as a feature of the OS.
5. **Distributed Systems and Networks:** A robust and redundant network would be needed to ensure all clients are able to reliably connect and interact. As each client application will rely on a single server application, client/server distributed system would be recommended.
6. **Security:** Windows built in Windows Defender can be leveraged for our security system. Ensuring all network traffic is properly encrypted will be required.