



Draw It or Lost It  
**CS 230 Project Software Design Template**  
Version 1.0

## 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>Requirements</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>5</b>

### Document Revision History

Version	Date	Author	Comments
1.0	07/14/2023	Anabel Villalobos	Write up of software design
2.0	07/18/2023	Anabel Villalobos	Complete evaluation
3.0	08/11/2023	Anabel Villalobos	Complete recommendations

## Executive Summary

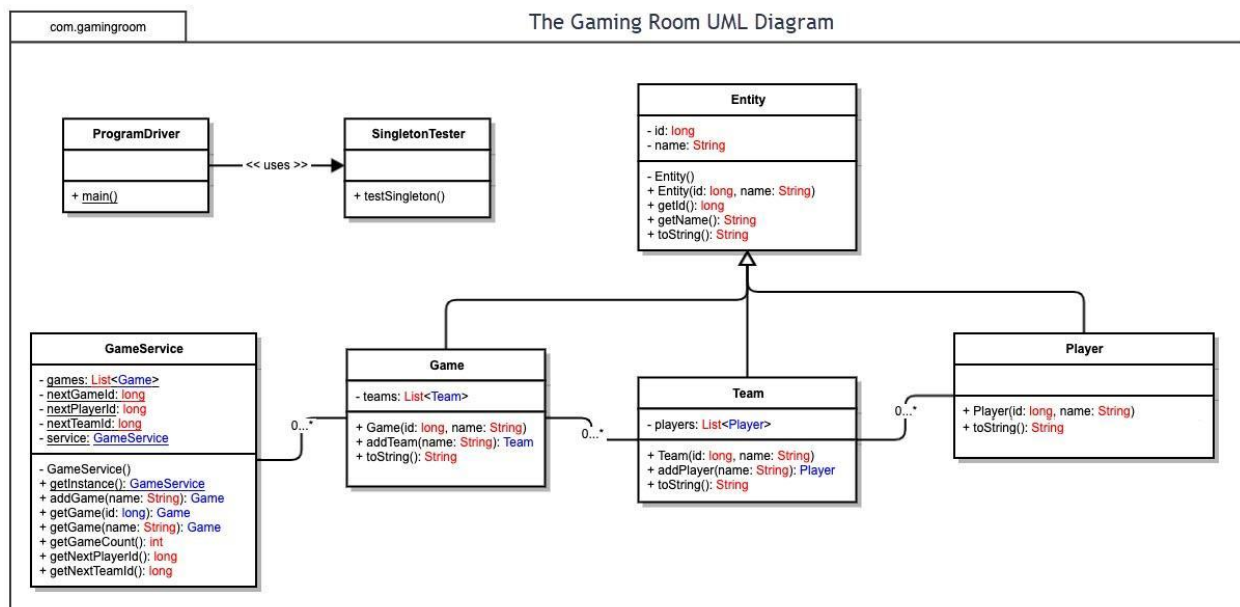
The company: The Gaming Room, wants to develop a web-based game that can run on multiple platforms. The title of the game will be “Draw It or Lose It” and is presently only available on androids. The game consists of four rounds each that last one minute long. The drawings are rendered and are fully complete after 30 seconds. If the team does not guess the puzzle before the time expires, the remaining teams have the opportunity to offer one guess to solve the puzzle in 15 seconds.

## Design Constraints

- One or more teams are involved
- There are multiple players on one team
- Game and Team names must be unique so that the game functions properly
- One instance of the game can exist at any time
- The game must be compatible on multiple OS platforms (other than androids).

## Domain Model

The Entity class creates a relationship between Game, Team, and Player. All three of these classes inherit their methods and attributes from the Entity class. The UML diagram below shows this inheritance. Each class shares common attributes like “name” and “id”. In the UML diagram, we can also see that Game Service has a zero to many relationship with the Game class meaning that the Game class can exist even without the Game Service class. This same situation exists between Game and Team and Team and Player.



## Evaluation

<b>Development Requirements</b>	<b>Mac</b>	<b>Linux</b>	<b>Windows</b>	<b>Mobile Devices</b>
<b>Server Side</b>	<p>User friendly terminal commands to configure the server, access, or make changes.</p> <p>Popular in web hosting.</p> <p>Advantages: There are many options for different web hosting requirements.</p> <p>Disadvantages: Less preferred for web hosting services.</p>	<p>More cost friendly.</p> <p>Secured and most preferred.</p> <p>Advantages: It is the most preferred for web hosting services.</p> <p>Disadvantages: It is difficult to find applications to support the web hosting requirements.</p>	<p>More software is available in comparison to other platforms.</p> <p>It is dominant to other platforms.</p> <p>Advantages: Quick loading time, high resource requirements</p> <p>Disadvantages: Hard to debug, susceptible to viruses</p>	<p>More popular, in higher demand, and high portability.</p> <p>Advantages: Will have more reach to more users, cost effective, compatible</p> <p>Disadvantages: Poor security</p>
<b>Client Side</b>	<p>Moderate expertise and time is required. Cost friendly.</p>	<p>Max expertise and time is required. Cost friendly.</p>	<p>Minimum expertise and time required. Cost friendly.</p>	<p>Provides flexibility to clients and developers to see updates at any place. More difficult to implement than other devices.</p>
<b>Development Tools</b>	<p>Programming Languages/Tools - Swift, Visual Studio Code, Eclipse, Adobe UXP, Atom, Delphi, Cloud 9, HTML/CSS/JavaScript, Java, Python</p>	<p>Programming Languages/Tools - Visual Studio Code, Eclipse, Adobe UXP, Jira, Atom, Delphi, Cloud 9, HTML/CSS/JavaScript, Java, Python</p>	<p>Programming Languages/Tools - Visual Studio Code, Eclipse, Adobe UXP, Jira, Atom, Delphi, Cloud 9, Azure, HTML/CSS/JavaScript, Java, Python</p>	<p>Programming Languages/Tools - Swift, Android, Kwater, HTML/CSS/JavaScript, Java, Python</p>

## **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 the outlined requirements, my suggestion leans towards opting for Linux as the server's operating platform. Linux stands as a prevalent choice for server applications owing to its stability, security, and overall performance. It boasts robust web support and can be tailored to specific needs. Considering that Draw It or Lose It operates as a web-based application, Linux emerges as the most fitting and suitable platform.
2. **Operating Systems Architectures:** A multi-tier architecture divides the application into distinct tiers: presentation, application, and data. Considering the requirements of this project, I propose the adoption of a multi-tier architecture. This approach facilitates enhanced scalability and bolsters fault tolerance, making it a valuable choice.
3. **Storage Management:** Given that Draw It or Lose It necessitates the rendering of images from an extensive repository of pre-made drawings, it becomes imperative to employ a high-performance storage remedy to guarantee a delightful user encounter. To effectively handle storage aspects, my suggestion leans towards adopting a solid-state drive (SSD) for the server's storage infrastructure. SSDs, characterized by their rapid data access, would be exceedingly effective in swiftly loading images and enhancing overall performance.
4. **Memory Management:** Solely relying on Linux yields exceptional memory management support, implemented through a paging system for efficient memory administration. Entrusting Linux in this regard could prove advantageous, ensuring optimal utilization of memory resources.
5. **Distributed Systems and Networks:** For establishing connectivity and enabling communication among diverse platforms, my suggestion would lean towards implementing a REST API. These interfaces find extensive application in web-based systems over the internet, functioning through HTTP requests and corresponding responses. Linux, in particular, furnishes robust backing for the integration of REST APIs.
6. **Security:** Security stands as a pivotal element in any web-based application. Draw It or Lose It should uphold this principle without exception. Opting for the suggested operating system, Linux, brings forth robust security support through features like firewalls and access control. Moreover, Linux boasts a well-regarded security reputation, displaying a heightened resistance to breaches compared to alternative operating systems. I would advise to implement SSL/TLS encryption for all communication to ensure confidentiality of data transmitted between the server and the clients.