
Creative Technology Solutions

**CS 230 Project Software Design Template**

Version 1.2

**Table of Contents**

**CS 230 Project Software Design**



**Document Revision History**

| Version | Date | Author Comments |
|---|---|---|
| 1.0 | <03/25/24> | <Brian Kremer> <Initial creation of the document> |
| 1.1 | <04/07/24> | <Brian Kremer> <Updating the evaluation section> |
| 1.2 | <04/19/24> | <Brian Kremer><Updating the Recommendations section> |

**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

Our client, The Gaming Room wants to expand their popular Android game called "Draw It or Lose It". They'd like to turn it into a web-based application accessible across multiple platforms. This move should broaden the game's audience and enhance its accessibility. Ultimately allowing teams to conveniently compete in guessing puzzles based on a library of stock drawings. Creative Technology Solutions (CTS) plans to beat this demand by proposing developing a scalable, web-based version of the game. They could do this by leveraging modern web technologies to ensure compatibility across various devices and platforms. The solution will prioritize user engagement through intuitive design, ensure the uniqueness of the game as far as the names of teams. We'll also maintain a singleton instance of the game in memory to manage resources effectively.

## Requirements

< Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client's business and technical requirements in a clear and concise manner.*>

## Design Constraints

- Web Based Environment: This is a design constraint because in multiplayer mode multiple people from around the world can play one game online. To effectively pull this off we must have efficient use of servers to handle multiple simultaneous game sessions. The data needs to be transferred between all players in real time. And the same game should be cross compatible for all users
- Uniqueness: When choosing a name we want to make sure that we arent going against another team with the same name. This means we must have a system in place to check the name library for uniqueness.
- Accessibility: This somewhat piggybacks on the web based environment but solely focuses on the cross compatibility and general accessibility of the game. It should be easy to obtain and easier to play.

## System Architecture View

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.

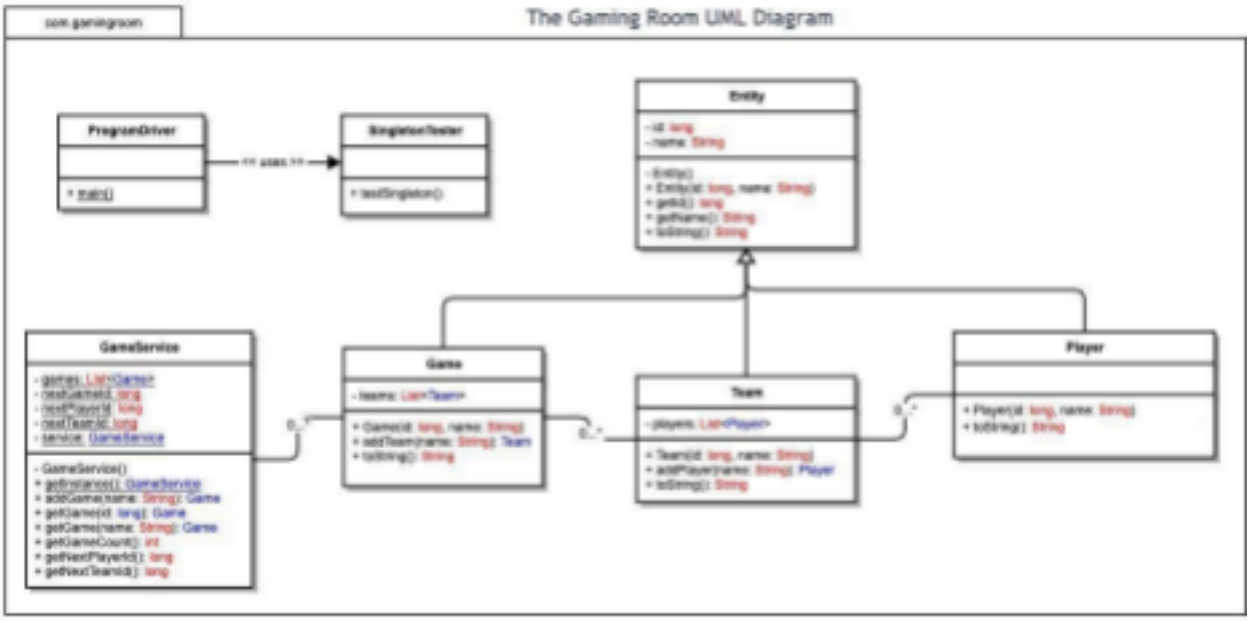
## Domain Model

The model below illustrates essentially everything I'm describing here. The game runs on 4 main classes that each have their own purpose. Our entity is where all the different classes go to work, where the user starts. Our game class contains everything for the game, for example the rules. Our Team class contains everything surrounding the teams, like their name. And our Player class contains everything surrounding the player like their ID. As mentioned above everything comes together in the entity class,



you could almost call this class the game. It brings in our unique classes like player and team to coincide

with the game and game service classes. Overall splitting up the sections of our game makes it easier to inherit options in the future and keep certain coding separate.




## Evaluation

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 | Characteristic is that mac's popular in web hosting. Piggybacking on this an advantage is their options for different web hosting. Another is that the OS is upgradable. A disadvantage is the limited scalability for large environments | A good characteristic is that it has a secure environment. Advantages are that security mistakes are usually discovered before they become an issue. A disadvantage is finding supporting applications that accept the hosting requirements. | A characteristic is that more software is available compared to the other options. It's also the most popular OS. An advantage is the high resource requirements. Low loading times are also a huge plus. A disadvantage could be since they're popular they're more susceptible to viruses. | A characteristic is that they're popular and portable. An advantage is the wide reach portability gives to mobile devices. They're also relatively cost efficient. A disadvantage is the poor security and level of attacks they're targets within. |
| Client Side | Here an average | Here we're | Here the cost is | Although slightly |

| | | | | |
|---|---|---|---|---|
| | expertise would be required to operate. The cost is close to windows and the time requirements are moderate. | required the most expertise and time to execute. On the flip side though the cost is minimal. | similar to MacOS. Minimum expertise is required and that goes for time as well. | more difficult to implement than other devices, here flexibility is provided to clients. This flexibility is even available to developers to see updates. |
| Development Tools | Mac will run all languages while supporting libraries to support their frontend and GenPurp. Languages. Mac has advantages of being able to use tools like notepad++ in addition to what you're already doing. | Here Linux will operate with visual studio, eclipse, and notepad++. This makes for easy to use tools. Here most languages are supported to include HTML, CSS, Javascript etc. | Here windows will have more ease than Linux. The nice thing about that is they run the same in the sense of all the tools you use to support the front end libraries like Python, PHP and Ruby. | Numerous applications can be made on swift and android. Both the software and languages can be run on the machines. The languages consist of HTML, CSS and Javascript. |

**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**: The Gaming Room will use a server operating system that's scalable and reliable. One good option would be the Microsoft Windows Server 2016. The server is a powerful and versatile operating system that can host a plethora of applications if need be. Draw it or Lose it will be one of the available applications. This server is also highly scalable making it easy to adapt to meet the needs of The Gaming Room as it expands.

2. **Operating Systems Architectures**: The Microsoft Windows server 2016 operating system has a multilayered architecture. These three points are the main layers to this architecture. The kernel: This is the core of the operating system. It manages the hardware and provides basic services to all other layers. The User Mode: This layer is where the application is run for the 'User' . It is isolated from the kernel layer to keep the kernel from receiving malicious code. Finally we have Drivers: These softwares provide a layer of abstraction between the operating system and the hardware. They allow the system to talk to the hardware without having to know the details of that specific hardware.

3. **Storage Management**: This Microsoft Windows Server 2016 operating system supports a variety of devices to include, disks, tapes, and optical disks. It could also support a variety of storage devices such as RAID, volume shadow copies, and deduplication.

4. **Memory Management**: The Windows Server 2016 operating system uses a variety of memory management techniques to keep track of memory usage and to ensure that applications have access to the memory they need. These techniques include virtual memory, memory mapping, and paging.

5. **Distributed Systems and Networks**: The Windows Microsoft Server 2016 operating system is designed to support distributed systems and networks. It provides a variety of features that make it easy to develop and deploy distributed applications. These features can include Remote procedure calls (RPC's), Distributed file systems (DFS), and Web services. RPC's allow applications to talk to each other across a network. The DFS systems allow applications to access files that are stored on remote servers say for storage purposes. Finally there's Web services that allow applications to talk to each other using the web.

6. **Security**: The operating system behind the Microsoft Windows Server 2016 provides a comprehensive set of security features to protect data and applications. These features could be Intrusion detection systems (IDSs), Firewalls, User accounts and passwords,

and Data encryption. First off, IDS's can be used to detect and alert administrators to security threats. Firewalls are used to block off unauthorized access to the network. User accounts and passwords are used to authenticate users/clients and control access to resources. Finally we have Data encryption that’s used to seal sensitive data from unauthorize access.