Draw it or Lose It

CS 230 Project Software Design Template

Version 1.0

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	03/28/2025	Felix Guzman	First Design Document Created

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

<Write a summary to introduce the software design problem and present a solution. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.>

The purpose of this Software Design Document is to further develop our client's, The Gaming Room, game application "Draw it or Lose it". Currently the application is limited to android and we must develop a web-based application based on its android app that would be able to support a multitude of devices and serve multiple platforms to gain a broader audience. The game is inspired by the 1980s tv show Win, Lose or Draw where teams would compete to guess what is being drawn. Instead of drawings the application will use a rendering puzzle like image that will be completed in 30 seconds. Players must guess what the image is before the time is over. This document will cover how we would help meet our clients expectations and specifications.

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.>*

1. The game will be made into a web-based application that would be able to be accessed by multiple platforms. Compatibility with a plethora of devices and operating systems is a must.

2. Each game should have the ability to have one or more teams involved. This means we must be able to handle multiple players as well as assigning them to their designated teams.

3. Game and team names must be unique. This will allow users to check whether a name is in use when choosing a team name.

4. Once one instance of the game can exist in memory at a given time. This would be achieved with the use of unique identifiers for each instance of a game, team, or player.

5. The application should cater to the game rules and their specifications. This means game rounds should have time limits (30 seconds), and if the time expires the remaining teams should have an opportunity to offer one guess each to solve the puzzle with a 15 second time limit.

Design Constraints

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

1. The Web-Based applications should be able to interact with multiple platforms. This means we have to consider multiple devices, operating systems, and web browsers and how they may interact with our website.

2. The application should be able to support multiple players and teams which means we must consider networking, and web technologies that allow us to implement this system effectively. In addition we must also take in mind related intricacies such as unique names and identifiers.

3. The application is limited to a single instance so our design regarding the development of the website and multiplayer aspect should keep this in mind for our design implementation.

4. We should take common use practices into mind while developing this application. This would include performance, security, and scale.

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

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

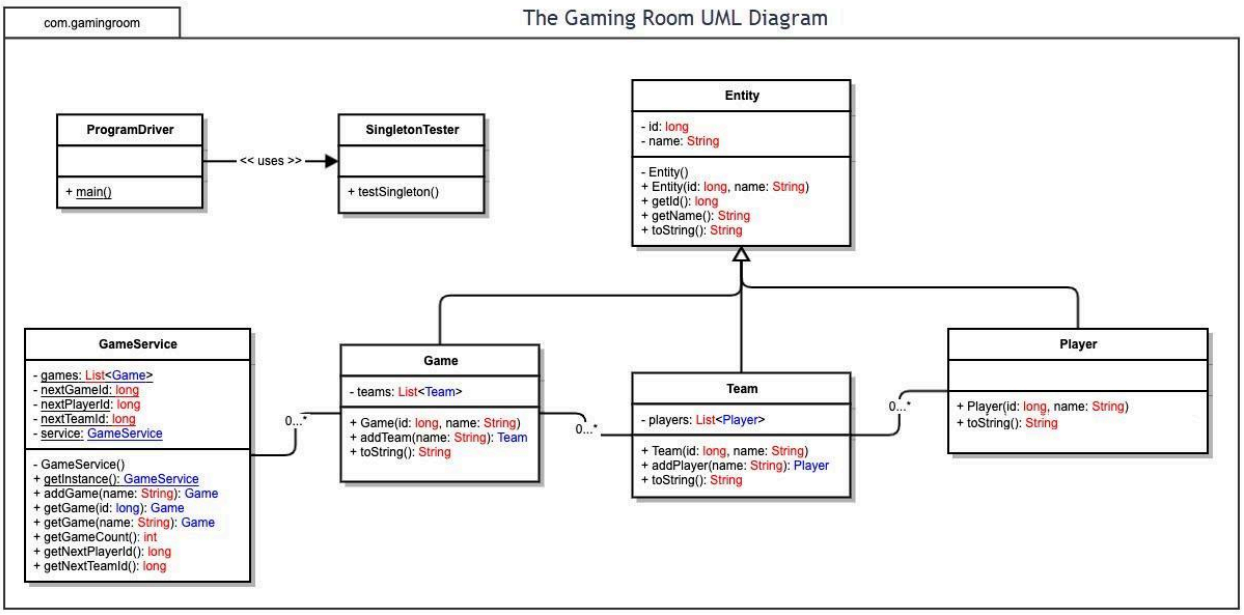
As we can see from the following Domain Model, a lot is taken into consideration regarding the design of this application.

To summarize, the model shows the main parts of the Draw It or Lose It application, all centered into the Entity class. The Entity class is an abstract class that holds all the necessary identification information such as name and ID. The reason it is abstract is so it acts as a blueprint, making sure that no accidental unneeded instances are initiated. Only classes that extend Entity, like Game, Team, and Player, would be instantiated. This is all part of inheritance, an object oriented principle that helps avoid code duplication and follows proper coding practices to make development easier and more efficient.

Following this development, we also have the GameService class, which is designed to be a singleton, meaning that only one instance would exist throughout the application. This is where one of our key requirements is fulfilled. By ensuring that the data remains consistent, we can enforce uniqueness by checking through any duplicate names or unique identifiers with an iterator loop and terminating them.

The ProgramDriver is also here, containing our main function, which serves as the starting point of the application, and the SingletonTester is used to verify that everything has been implemented correctly.

Together, all these components work in unison to create a structured program that would help us fulfill all our requirements.



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	<Evaluate Mac for its characteristics, advantages, and weaknesses for hosting a web-based software application.> Mac is Unix-based, making it stable, easy to use, and developer-friendly. However, it is not commonly used for hosting due to higher costs and limited industry adoption. It is also not as scalable as Linux or Windows.	<Evaluate Linux for its characteristics, advantages, and weaknesses for hosting a web-based software application.> Linux is widely used and cost-effective, unlike Mac. It also has extensive customization options and is open-source. However, it lacks compatibility with some software and requires experts to use.	<Evaluate Windows for its characteristics, advantages, and weaknesses for hosting a web-based software application.> Windows has unmatched compatibility and seamless integration with popular tools. It is developer-friendly and extremely well-documented. However, compared to Linux, it can be more expensive.	<Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.> Mobile devices are extremely portable and can access information from anywhere. However, they are an extremely poor choice for hosting due to hardware limitations.

Client Side	<Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mac.> Mac can be extremely intuitive and user-friendly with its sleek design. However, hardware costs are high, and testing is limited to Apple's ecosystem.	<Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Linux.> Linux is extremely cost-effective since the system is free to use and can be distributed across most hardware devices. However, it may require a strong skill set to utilize, which could require experts or training, contributing to costs.	<Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Windows.> Windows has access to the greatest variety of software and unmatched compatibility. It is developer-friendly and familiar to many. However, it can be more expensive than Linux and less secure than Mac.	<Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mobile Devices.> Mobile devices can be easy to use. However, there are many considerations to take into account, including different operating systems and ways to handle input and output.

Development Tools	<Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mac.> Mac has access to its own proprietary tools like Xcode and Swift. They are very developer-friendly and easy to use. Security is also top-notch. However, they are centered around Apple-specific development.	<Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Linux.> Linux has tons of development tools to utilize, from Atom to VSCode. It is often used due to its command-line utilities, cost-effectiveness, and deployment tools. However, it requires specialized skills to utilize.	<Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Windows.> Windows, like Linux, has a diverse toolset, possibly even bigger, with even more documentation. However, it may be more equipped to handle Windows devices.	<Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mobile Devices.> Mobile devices have varying tools depending on the operating system. Usually, Xcode is used for iOS and Android Studio for Android. This creates a big limitation on what tools can be used to develop applications.

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**: <Recommend an appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments.>

After analyzing all the operating platforms provided I would recommend utilizing Windows. With this operating system we have a wide range of software and hardware at our disposal due to its unmatched compatibility. Additionally, Windows provides a stable and well-documented environment with tons of familiarity, which is ideal for hosting all types of applications, including web-based ones. Windows is used widely in both developer and consumer settings and would be ideal for the task.

2. **Operating Systems Architectures**: <Describe the details of the chosen operating platform architectures.>

The Windows operating system architecture is extremely well-built and robust. It is highly scalable and modular, and it comes with excellent documentation. These factors ensure that the system can be easily developed, maintained, and updated as needed.

3. **Storage Management**: <Identify an appropriate storage management system to be used with the recommended operating platform.>

Keeping my chosen operating system in mind, utilizing Microsoft SQL Server would be an effective way to manage storage. With SQL Server, we have access to advanced security features and support for complex queries, making it straightforward to store all types of game data.

4. **Memory Management**: <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.>

Windows has strong memory management techniques formed through its virtual memory system and dynamic allocation capabilities. Much like the Java application developed for Draw It or Lose It, Windows would optimize resource allocation and ensure that the game only utilizes a single instance at any given time thus, preventing memory leaks.

5. **Distributed Systems and Networks**: <Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).>

Hosting the web-based application on a Windows environment would allow us to utilize Windows Server and other associated technologies. This approach facilitates communication and supports the application's modular design through Windows-supported technologies. Even if errors occur or unidentifiable parts are found, data integrity would be prioritized and the service would remain resilient to these issues.

6. **Security**: <Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.>

With Windows, we have access to tons of security features including advanced encryption protocols, regular security updates, and powerful authentication methods. Being the most used operating system in the world, Windows is routinely tested and scrutinized, ensuring that we have an up-to-date platform with plenty of tools, documentation, and capabilities at our disposal to protect user information on and between various platforms.