<Name-of-Software-Application>

CS 230 Project Software Design Template

Version 1.0

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	<12/15/24>	Reginald True	Added Evaluation info operating system architecture Memory management File Systems

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

The Gaming Room's Draw It or Lose It game is an engaging, interactive game inspired by classic guessing games. The client seeks to expand the platform's reach by hosting the game as a web-based application while supporting cross-platform play on desktop and mobile devices. This document evaluates various platforms and development strategies to guide the expansion process.

Requirements

- **Server-Side**:
  - Host the game on a scalable, web-based platform capable of supporting thousands of concurrent players.
  - Ensure compatibility with Linux, Mac, and Windows servers.
- **Client-Side**:
  - Deliver a responsive, browser-based HTML5 interface for desktop clients.
  - Expand the game's accessibility to both iOS and Android mobile platforms.
- **Development**:
  - Ensure efficient, cost-effective development by leveraging open-source tools and frameworks.
  - Provide seamless integration between client and server components using modern development practices.

Design Constraints

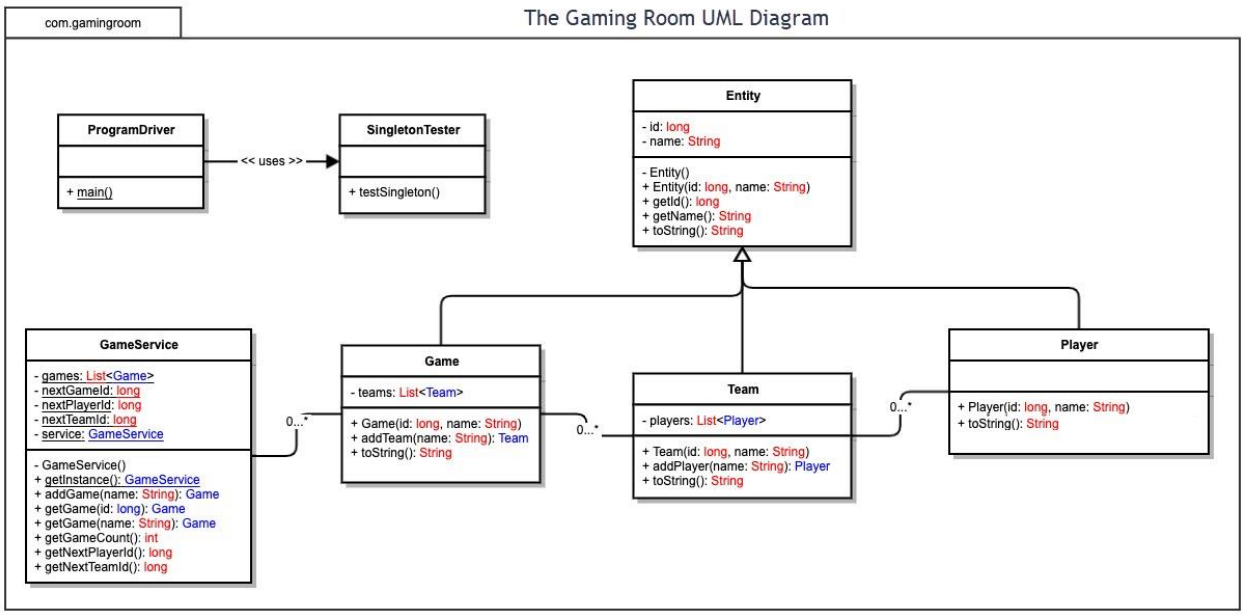
- **Server Environment**:
  - The server must support hosting via Linux for cost-efficiency and scalability.
- **Cross-Platform Compatibility**:
  - The application must run in all major browsers on desktop and mobile devices.
- **Development Tools**:
  - Utilize tools that minimize licensing costs and maximize productivity, such as IntelliJ IDEA Community for Java and React for frontend development.
- **Linux for Servers**:
  - Preferred for its stability, scalability, and open-source community support.
- **ReactJS/React Native**:
  - Chosen for its responsive design capabilities and component-based architecture, enabling code reuse across platforms.
- **Development Tools**:
  - Open-source tools like Maven, Gradle, and Docker ensure efficient backend deployment, while tools like React Native simplify mobile development.

System Architecture View

The game's system will be set up in three main parts. The first part is the Frontend, which is what players see and use on their devices, like phones, tablets, or computers. It will be designed to work well on all screen sizes and browsers. The second part is the Backend, which handles all the game rules and keeps track of things like teams, players, and game progress. It also makes sure only one game runs at a time. This part will run on a server that can handle many players playing together. The last part is the Database, which stores all the important information, like team names, player details, and game sessions. It will make sure no two teams or games have the same name. These parts will talk to each other securely, so the game works smoothly and safely. This setup will also make it easier to fix problems and add new features later.

Domain Model

For the domain model, it shows how the parts of the game work together. The Game class is the main part that keeps track of the game and the teams playing. Each Team has its own name and a list of players. The Player class represents the people playing the game and connects them to their teams. All the parts share some basic features, like IDs, through a special class called Entity, which helps save time by reusing code. This setup uses programming ideas like keeping information safe in each part and sharing common features. This makes the game easy to run, follow the rules, and add new features later.



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	Can host, but not as common for servers.	Works great, free, and very reliable.	Works well, but costs more.	Doesn't apply to servers.
Costs	Expensive hardware and tools.	Free to use. Some versions may cost money.	Needs paid licenses.	Free to make apps, but stores charge fees.
Scalability	Not as flexible for big growth.	Very scalable and works well for big games.	Can grow but costs more money to do so.	Needs strong APIs to keep up with players.
Time and Money to Build	Slower and costs more for tools.	Fast to build with free tools like Maven.	Some tools like Visual Studio cost money.	React Native lets you build for both phones.
Works on Many Platforms?	Doesn't work as well for big apps.	Java helps it work on all kinds of computers.	Good for enterprise use.	React Native or Flutter helps with this.
Security	Safe but less flexible.	Very safe with lots of tools for protection.	Safe but may cost more for extra features.	Mobile devices have built-in safety tools.

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**: I recommend using Windows as the operating platform. Windows is one of the most popular operating systems in the world and works on a wide range of devices, from personal computers to gaming consoles. It's easy for developers to create software for Windows, and it supports a large number of gaming peripherals and hardware. Since Draw It or Lose It will need to reach many players, Windows offers a great foundation for the game, allowing it to run smoothly on both home computers and gaming PCs.

2. **Operating Systems Architectures**: Windows uses a hybrid kernel architecture, which combines elements of both monolithic kernels and microkernels. This means Windows is both efficient and flexible, able to handle many types of software and hardware. For Draw It or Lose It, this architecture would allow the game to work well on different devices, from basic laptops to high-performance gaming PCs. The hybrid kernel ensures that the game can run smoothly, even with lots of players and complex game features.

3. **Storage Management**: For storage management, NTFS is the best choice for Windows. NTFS is the default file system for Windows and is known for its ability to handle large files and complex directories. It also includes security features that help protect game data from corruption. For Draw It or Lose It, NTFS will allow the game to store user profiles, drawings, and game progress securely. It supports large amounts of data and can easily be backed up, making it a reliable choice as the game grows.

4. **Memory Management**: Windows uses a method called virtual memory to manage the computer's memory. This allows Windows to use more memory than what is physically installed by swapping data between the computer's RAM and the hard drive. For Draw It or Lose It, this means the game can run smoothly even when lots of things are happening at once, like showing images or handling multiple players. Windows also manages memory for each program separately, preventing one part of the game from using too much memory and slowing things down. This ensures that the game can handle many players without crashing or freezing.

5. **Distributed Systems and Networks**: To make sure Draw It or Lose It works across different devices, Windows can support distributed systems. The game can use a client-server model, where the server holds all the important game data and the players' devices connect to it to send and receive updates. For this to work well, the game will need to use a strong internet connection and a server that can handle many players at once. Windows supports TCP/IP networking, which will allow the game to send and receive data between the server and the clients. If there are network issues, Windows also supports caching to temporarily store data on the player's device until the connection improves.

6. **Security**: Security is very important to keep players' personal information and game data safe. Windows has built-in tools to help protect data, such as BitLocker for encryption and Windows Defender for virus and malware protection. For Draw It or Lose It, Windows can use SSL/TLS encryption to protect data sent over the internet, ensuring that players' information is safe while they play. It also supports user account control which helps prevent unauthorized changes

to the game or system. Using multi-factor authentication will add an extra layer of security, ensuring that only authorized players can access their accounts.