The Gaming Room

CS 230 Project Software Design Template

Version 3.0

Table of Contents

CS 230 Project Software Design Template 1

Table of Contents 2
Document Revision History 2
Executive Summary 3
Requirements 3
Design Constraints 3
System Architecture View 3
Domain Model 3
Evaluation 4
Recommendations 4

Document Revision History

Version	Date	Author	Comments
3.0	12/10/2024	Christian Sebo	Updated with Project Three details, including operating platform recommendations, storage management, memory management, distributed systems, and security considerations.

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* application is expanding from its Android-only setup to support multiple platforms. This document evaluates Linux, Mac, Windows, and mobile platforms for hosting, development, and compatibility requirements in a distributed environment. The goal is to deliver a scalable, secure, and responsive application that works seamlessly across desktop and mobile platforms, aligning with the client's requirements for a modern web-based solution.

Requirements

The client requires a game application that can operate across multiple platforms, ensuring compatibility, security, and scalability. Key technical requirements include: (

1. Support for server-side operations.
2. Efficient handling of multiple users simultaneously.
3. Strong security measures to protect user data.
4. Cross-platform compatibility for Linux, Windows, Mac, and mobile devices.

Design Constraints

Developing *Draw It or Lose It* for a web-based distributed environment comes with several constraints:

1. **Server-Side Hosting**: Hosting must handle scaling for thousands of players while maintaining cost-efficiency.

2. **Development Compatibility**: Development tools must ensure the application is compatible with desktop and mobile web browsers.

3. **Licensing Costs**: Tools and platforms with high licensing fees may impact the project's budget.

4. **Team Expertise**: Developers must have skills in multiple programming languages and platforms for successful deployment.

System Architecture View

This section will focus on the web-based application architecture with a client-server model. The game logic and storage are hosted on the server side, while the client interface is delivered through responsive web browsers on all supported platforms.

Domain Model

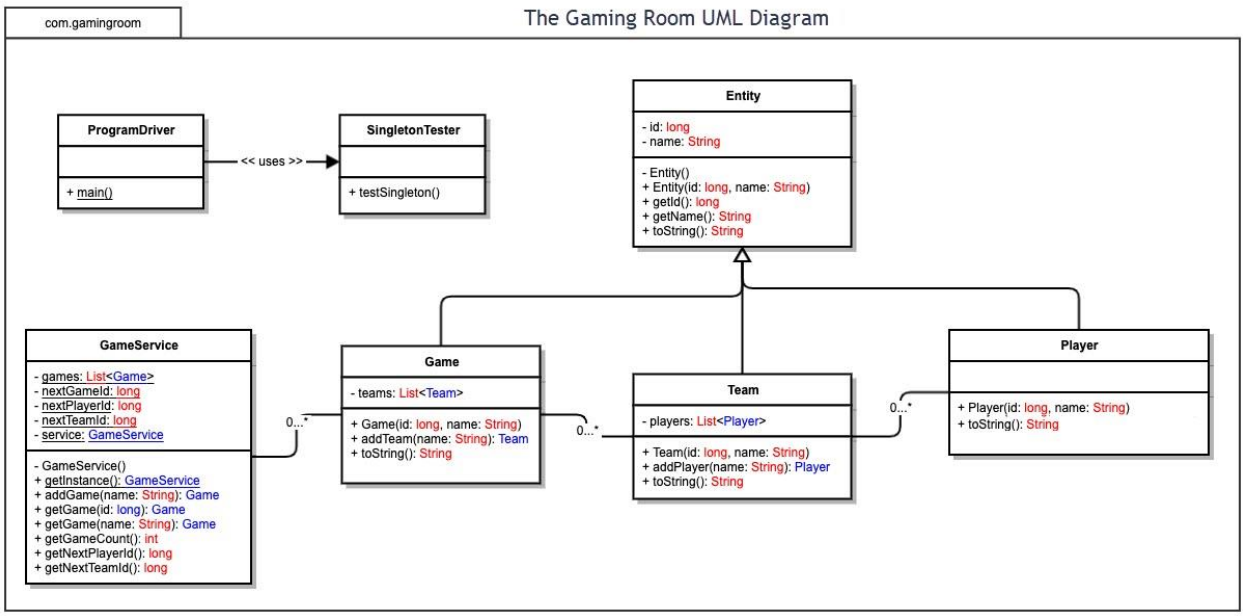
The provided UML diagram outlines the structure of the application.

- **Entity**: The base class for `Game`, `Team`, and `Player`, encapsulating shared properties.

- **GameService**: Ensures centralized control of the application, using the Singleton pattern for efficient management of resources.

- **Game**: Contains a list of teams and manages game-related logic.

- **Team**: Organizes players into groups for gameplay.
- **Player**: Represents individual participants.
- The application uses inheritance, encapsulation, and the Singleton pattern to manage data efficiently and meet 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	Mac is stable and user-friendly for hosting but comes with higher costs for hardware and licensing.	Linux is highly scalable, free, and ideal for hosting web applications, but it requires experienced developers.	Windows is widely used and supports many tools, but licensing costs and higher resource usage can be drawbacks.	Mobile devices are not suited for server hosting but serve as clients to connect with the back-end.

Client Side	Developing for Mac requires Xcode and Swift, which increases costs but offers smooth integration.	Linux development requires open-source tools and expertise but is cost-effective.	Windows is easy to develop for, using Visual Studio, but the tools may add licensing fees.	Mobile development requires frameworks like Flutter or React Native to ensure compatibility across iOS and Android.
Development Tools	Swift and Xcode are key for Mac, offering seamless development but with licensing fees.	IDEs like Eclipse and IntelliJ are free and flexible for Linux, making it cost-efficient.	Visual Studio, C#, and .NET are widely used for Windows, but tools may require licenses.	Tools like Flutter, Kotlin, or Swift are essential for mobile apps, with moderate licensing costs.

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**: To expand **Draw It or Lose It** across various computing environments, **Linux** is recommended as the operating platform. Linux offers stability, scalability, and flexibility—key attributes for managing high-demand, multi-user gaming environments. Its robust security features, cost-effectiveness, and strong community support provide an excellent foundation for seamless, secure operations across multiple platforms.

2. **Operating Systems Architectures**: The **Linux architecture** is ideal due to its modular kernel design, which supports efficient scalability and customization. The architecture handles concurrent game instances effectively, ensuring optimal performance. Furthermore, Linux's compatibility with a variety of hardware configurations, from physical servers to cloud-based deployments, makes it highly versatile for implementation and maintenance.

3. **Storage Management**: To accommodate the substantial storage demands of **Draw It or Lose It**, including a vast library of high-definition images and other assets, **Network Attached Storage (NAS)** is recommended. NAS enables high-capacity, high-throughput storage solutions, which are crucial for frequent read/write operations in gaming. To bolster redundancy and backup capabilities, **cloud storage solutions** such as **Amazon S3** can be integrated, ensuring data availability and integrity during peak loads or system failures.

4. **Memory Management**: Linux employs advanced memory management techniques suitable for the gaming industry. Features such as **paging**, **virtual memory**, and **demand loading** allow efficient allocation of memory resources. These techniques minimize latency and ensure responsiveness under high server loads, enabling **Draw It or Lose It** to provide a seamless gaming experience with optimal resource utilization.

5. **Distributed Systems and Networks**: To support seamless communication across platforms, adopting a **microservices architecture** is recommended. This architecture allows components of **Draw It or Lose It** to be deployed across diverse platforms while ensuring scalability and fault tolerance. Inter-component communication can be managed through **APIs** and **message queues**, providing real-time synchronization and enhancing gameplay. Additionally, incorporating **Content Delivery Networks (CDNs)** will optimize the delivery of game assets, reducing latency and improving the user experience.

Security: Ensuring user data security is paramount in a multi-platform environment. Recommended security measures include:

6. **SSL/TLS encryption** for secure data transmission.

7. **OAuth** protocols for user authentication.

8. Firewalls and intrusion detection systems to prevent unauthorized access.

9. Regular security audits and compliance with data protection standards such as **GDPR** or **CCPA** to enhance trust and reliability.

- These measures will protect user data across platforms, providing a secure and trustworthy gaming environment.