THE
GAMING
ROOM

Draw It or Lose It
CS 230 Project Software Design
Version 1.0

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	12/01/24	Andre Rowe	Initial Draft

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 current game, Draw It or Lose It, operates exclusively on Android. To expand its reach, this project evaluates how to develop a web-based, cross-platform version of the game. The proposed solution focuses on hosting the application on reliable server platforms, ensuring compatibility with desktop and mobile operating systems, and using cost-effective development tools. This document highlights the characteristics, advantages, and challenges of Linux, Mac, Windows, and mobile platforms and recommends the best approach for the client's needs.

Requirements

The Gaming Room requires a responsive web application to:

Host on a scalable server that supports thousands of users.
Ensure compatibility across Linux, Mac, Windows, iOS, and Android platforms.
Use modern development tools to minimize costs and maintain high security.

Design Constraints

1. Scalability: The server must handle a high volume of simultaneous users without performance issues.

2. Cross-Platform Development: The application needs to run seamlessly across desktop and mobile operating systems.

3. Budget Limitations: The solution must minimize licensing and development costs while maintaining quality.

4. Security: The platform must secure user data with robust encryption and authentication measures.

Evaluation

Development Requirements	**Mac**	**Linux**	**Windows**	**Mobile Devices**
Server Side	Characteristics: Uses macOS Server with Apache. Advantages: User-friendly; strong Apple ecosystem support. Weaknesses: High hardware and licensing costs.	Characteristics: Supports Apache/Nginx. Open-source. Advantages: Cost-effective, scalable, secure, and customizable. Weaknesses: Requires skilled administrators	Characteristics: Built-in IIS for hosting. Advantages:Strong enterprise integration and robust support. Weaknesses:Expensive licensing and potential scalability issues.	Characteristics: Requires external hosting. Advantages: Scalable cloud hosting solutions available. Weaknesses: Limited hosting capabilities on native mobile platforms.
Client Side	Higher development costs due to Apple-specific tools.	Cost-effective open-source tools but requires expertise.	Uses enterprise tools like Visual Studio, which are costly but efficient.	Responsive design frameworks like Flutter ensure compatibility across devices.
Development Tools	Swift, Xcode, JavaScript.	Python, JavaScript, VS Code.	C#, .NET, Visual Studio.	Flutter, React Native, Java/Kotlin.

Recommendations

Operating Platform

Linux is recommended as the operating platform for hosting Draw It or Lose It. It offers excellent scalability, robust security, and cost-efficiency, making it ideal for a cross-platform gaming application. Additionally, Linux supports a wide range of development tools and has extensive community support, ensuring flexibility for future expansion.

Operating System Architectures

A Linux-based server architecture with Docker containerization is suggested to enhance scalability and maintain isolated environments for different components of the game. This approach allows seamless updates and deployment of new features while ensuring system stability. The lightweight Linux kernel ensures efficient multitasking and resource allocation, vital for handling multiple users simultaneously.

Storage Management

AWS S3 is the recommended storage management solution. It provides flexible, scalable, and secure data storage, essential for handling the game's dynamic data requirements, such as user-generated content and session data. Its redundancy and availability features ensure minimal downtime and data protection.

Memory Management

To optimize performance, caching mechanisms such as Redis or Memcached should be implemented. These technologies reduce server load during peak usage by storing frequently accessed data in memory, thus minimizing delays in gameplay. Linux's virtual memory and demand paging further enhance efficiency by dynamically allocating resources as needed.

Distributed Systems and Networks

To enable communication across platforms, the game should use RESTful APIs for handling requests and WebSocket for real-time updates, ensuring smooth gameplay interactions. The network architecture should incorporate load balancers to distribute traffic evenly and maintain high availability. Dependencies such as internet connectivity and fallback mechanisms should be addressed to mitigate potential outages.

Security

Security is paramount for protecting user data and maintaining trust. The following measures are recommended:

- Encryption: Use TLS/SSL for encrypting data in transit and AES for securing data at rest.
- Authentication: Implement OAuth for secure user login and enforce multi-factor authentication (MFA) for additional protection.
- Platform-Specific Security: Leverage Linux Security Modules (LSM) such as SELinux for enforcing strict access controls, ensuring data integrity across the platform.