

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 | 01/26/2024 | Matthew Cohen | First version |

**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**

Creative Technology Solutions (CTS) is tasked with developing a web-based version of The Gaming Room's game, Draw It or Lose It, currently available only on Android. This game, akin to Win, Lose or Draw, entails teams guessing phrases from rendered images. Our proposed solution includes designing a system allowing multiple teams with unique names and ensuring a single instance of the game in memory at any time. By leveraging web technologies, we aim to expand the game's reach across various platforms efficiently.

**Design Constraints**

Developing the web-based Draw It or Lose It game poses several design constraints. Ensuring uniqueness of game and team names demands robust database management to prevent conflicts. Scalability and performance optimization are crucial for rendering images and managing multiple teams concurrently. Security measures must safeguard user data and prevent unauthorized access. Additionally, seamless integration across platforms necessitates adherence to web standards and compatibility across browsers and devices.

**Domain Model**

Inheritance: Game, Team, and Player all inherit from entity and utilize its methods and variables.  
Encapsulation: Entity has private variables which are only able to be changed externally via Entity's getters and setters.  
Polymorphism: There are several different classes with different behaviors which all inherit from entity. (this is somewhat closer to absraction, however, the entity class was not marked as abstract)



**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 systems offer stability and ease of use but may have limited scalability for hosting web-based applications. | Linux provides robust performance and scalability, making it an ideal choice for hosting web-based software applications. | Windows offers comprehensive development tools and compatibility but may suffer from security vulnerabilities and licensing costs. | Mobile devices present challenges in hosting web-based applications due to resource limitations and platform fragmentation. |
| **Client Side** | All of these will require additional resources, knowhow, and testing in order to support clients on multiple platforms. This will require knowledgable workers, more money spent on licenses, and a longer development cycle. |  |  |  |
| **Development Tools** | Mac development necessitates proficiency in Xcode and Objective-C/Swift. | Linux development relies on open-source tools and libraries. | Windows development often leverages Visual Studio and .NET framework, requiring familiarity with Microsoft technologies. | Mobile development demands expertise in platforms like Android Studio and Swift/Xcode for Android and iOS respectively. |

**Recommendations**

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

* **Operating Platform**: Linux stands out as the recommended operating platform due to its cost-effectiveness, stability, and security features.
* **Operating Systems Architectures**: Linux architecture offers modularity, reliability, and extensive community support, ensuring scalability and flexibility for future enhancements.
* **Storage Management**: Utilizing relational databases like MySQL on Linux ensures efficient data storage and retrieval, supporting concurrent user interactions.
* **Memory Management**: Linux employs efficient memory management techniques like virtual memory and swap space, optimizing resource utilization for Draw It or Lose It.
* **Distributed Systems and Networks**: Implementing RESTful APIs and WebSocket protocols facilitates seamless communication between platforms, ensuring real-time updates and synchronization.
* **Security**: Linux's built-in security features and community-driven updates mitigate vulnerabilities and safeguard user information during transit and storage. Encryption protocols and access controls enhance data integrity and confidentiality across distributed systems.