

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 | 03/19/2022 | Jason Buol | Created Software Design |

**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 currently has an android app Draw It or Loose It and they would like to develop a web-based game based off the app to serve multiple platforms. The app and the web-based game will be similar to the popular game show Win Lose or Draw.

**Design Constraints**

* The game needs to be able to run on multiple platforms.
* Each game needs a unique identifier to ensure that there is only one instance of each game at a time.
* Each player and team will need a unique name to ensure that there is only one player and team using the name.
* The software will need to be able to run checks to ensure that there are no duplicates of any game, team name or player name.
* The development team would need to be comprised of people that are familiar with all of the platforms that The Gaming Room wants the game to be run on.

**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**

The Entity class is the super class and the game, team and player classes all inherits from and extends the Entity class. The Player class is linked to the Team class through aggregation, the Team class is linked to the Game class through aggregation and the Game Service class is also linked to the Game class through aggregation. Program Driver contains our main function and is where the command of the program takes place, it also uses the Singleton Tester function. The Singleton Tester function is where each game, team and player are checked to ensure that each instance of these are unique.



**Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| **Server Side** | - Gaining popularity as a host for web services  - Very stable OS  - Good Security  - Higer cost than both Microsoft and Linux  - Most people are more familiar with the Windows Platform | - Free OS  - Very Stable OS  - Very Secure  - Not a mainstream OS so less people are familiar with it | - Most people are familiar with Windows  - Very compatible with 3rd party software making it very flexible  - Expensive  - The least stable option  - Has the most security flaws | At this time mobile devices are not suited for hosting. |
| **Client Side** | - Very stable and secure  - Not very compatible with the mainstream gaming industry  - Expensive licensing fees and hardware costs  - Could be harder to find knowledgeable programmers for this OS possible leading to longer development times. | - Open-source free operating system  - Very stable and secure  - Have to use separate apps to run most games that are not browser-based  - Most people are unfamiliar with the Linux operating systems | - The most popular OS so most people are familiar with it  - Compatible with all gaming software  - Expense is similar to Mac  - The least stable OS  - Least secure OS | - Most people already have a cell phone so cost would be minimal  - There is already an app that has been developed for Android so only the IOS version would have to be developed reducing development time |
| **Development Tools** | - Javascript  - HTML 5  - Lua  - Visual Studio Code  - Atom  - Liveweave  -IntelliJ IDEA | - Javascript  - HTML 5  - Lua  - Visual Studio Code  - Atom  - Liveweave  -IntelliJ IDEA | - Javascript  - HTML 5  - Lua  - Visual Studio Code  - Atom  - Liveweave  -IntelliJ IDEA | - Javascript  - Lua  - Visual Studio Code  - Atom  -IntelliJ IDEA |

**Recommendations**

* **Operating Platform**: Due to overall cost, stability and security I reccommend Linux specificly the Debian distrobution due to it having been established for a long time, its focus on stability and that it is exceptionally secure.
* **Operating Systems Architectures**: The Kernel of the OS is the heart of the system and manages the memory, periperal devices and the CPU. The Daemons take care of the background operations like scheduling, printing, sound, ect. The graphical server and desktop environment take care of displaying graphics and giving the user a visual interface to interact with. Applications are the software that the user runs with the OS to accomplish any number of required processes.
* **Storage Management**: Linux is capable of using the conventional local storage harddrive options that many people use today. There are also many alternitave storage options available for Linux including open-sourced cloud-based storage options like OwnCloud and NextCloud.
* **Memory Management**: Linux uses multiple ways to help manage its memory like virtual memory, page caching, reclaiming pages, compaction and OOM killer.
* **Distributed Systems and Networks**: Linux is horizontally scalable to help ease the burden on any one device. It also can utilize shared memory for familiar read write operations, shared memory has high bandwidth, low latency, and low overhead
* **Security**: From its beginning’s security has been a cornerstone for Linux. Due to its open-source format with many different versions of operating systems, architectures, and components it is hard for malware to sweep through all Linux systems. Another benefit of it being open-source is the large number of developers that are both using Linux and revising its code to make it as secure as possible. With most versions of Linux if a virus does infect a system, it only effects the current user and is not able to get to the root user. This keeps the system more secure and allows the root user to go in and more easily remove the virus.