The Gaming Room

# Software Design Document

Version 2.0

Last updated: 08/20/2022

Author: Erick Herrera Cabrera

## Table of Contents

[Software Design Document 0](#_Toc111573281)

[Table of Contents 1](#_Toc111573282)

[Document Revision History 1](#_Toc111573283)

[Executive Summary 2](#_Toc111573284)

[Design constraints 2](#_Toc111573285)

[Domain Model 2](#_Toc111573286)

[Evaluation Matrix 3](#_Toc111573287)

[OS evaluation summary 6](#_Toc111573288)

## Document Revision History

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/17/2022 | Erick Herrera Cabrera | Added details to all section presented in the table of content. |
| 1.5 | 08/03/2022 | Erick Herrera Cabrera | Added evaluation matrix for Operating Systems |
| 2.0 | 8/16/2022 | Erick Herrera Cabrera |  |

## Executive Summary

This project looks to develop the web-based version of *Draw it or Lose it*, which the company *The Gaming Room* currently has available in the Android app store. The application needs to work for multiple platforms. *Draw it or Lose it* is loosely similar to a TV show game called Win, Lose or Draw, and the game consists of four rounds of play lasting one minute each. Drawings are rendered at a steady rate and are fully complete at the 30-second mark. If the team does not guess the puzzle before time expires, the remaining teams have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit.

## Design constraints

* Currently the application only exists for the Android OS, so the web-based application will need to communicate with the Android version to check the following:
  + Team names are unique.
  + Game instance is unique for each team.
* Application needs to work across all popular web-browsers
* Application needs to run on the cloud to save on cost and make it scalable across multiple platforms.

## Domain Model

Diagram

Description automatically generated

1. Game Service class
   1. The game service class allows the creation of games on the same instance helping managing memory, and stores them in a List of Games. For every game instance created, a Game, Team and Player objects gets instantiated extending from the Entity Object.
2. Game class
   1. This object gets created when the Game Service class gets instantiated and holds the List of Teams in the Game instance.
3. Team class
   1. The team class holds the List of Players in each Team, provides the method to add a player.
4. Player class
   1. The Player class holds the information of each player in the Team.
5. Entity
   1. Provides a the getId()and getName() methods to the objects created for each game instance (Game, Team, and Object).
6. Program driver
   1. Creates the instance of a Java Console Program that will be used to instantiate Game objects using the GameService class. It will also run the singleton class that will Test if there is a Game instance already created with the same name.

## Evaluation Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operating System** | **OS Fitness Criteria** | **Characteristics** | **Advantages** | **Weaknesses** | **Rating**  **(1-5)** |
| Mac OS | Client Side | Proprietary to Apple and is not available for Windows or Linux users. Since is proprietary it will require purchasing a license. | -The most secure option in the market.  -Apple dedicated support. | -Requires apple server  -License for Mac OS is the most expensive in the market | 3 |
| Server Side | Same look and feel of the OS X running on Apple devices. Requires getting a license since it is proprietary to Apple. It is based on Linux, but it is not backwards compatible when it comes to webhosting services that offer Linux. | -If you are already very used to OS X running on Apple devices, it will be easy to use.  -It is designed to work very well with the Apple mini (server).  -The most secure option in the market. | -It is usually not supported by most web hosting services.  -Requires purchasing license.  -It is not open source. | 2 |
| Development tools | Development tools for Mac OS web hosting are very abundant, and there’s multiple ways to circumvent the set up you ae working with. | -Development tools are built to integrate with the UNIX structure.  -Development tools are vetted by Apple | -Most development tools are not open source and require the purchase of a license. | 4 |
| Windows | Client Side | A Windows client will be available to most people since Windows is the most utilized OS. Windows offer the capability of using Linux for the server and Windows OS for the Client, but it will require a license to use it. | -It is available for most web hosting services.  -It is available for Linux users and OS X users  -Most users are familiar with Windows and have it in their devices. | -Requires purchasing license.  -Not the most secure in the market.  -Not open source.  -Cannot run on Mac OS unless developer tools are implemented. | 3 |
| Server Side | Windows server OS is particularly helpful when the app being hosted utilizes the .net framework and the Client also utilizes Windows OS. | -Windows azure cloud integration.  -Can connect server to a Windows, Mac OS or Linux device client.  -Can be deployed with a Linux server | -Requires purchasing a license.  -Not open source  -Mostly used when the scope is only the .net framework, and Azure cloud services are being used. | 3 |
| Development tools | Windows has many developer tools when it comes to server / client | Development tools are vetted and sponsored by Microsoft. Windows server comes with its own set of tools. | Development tools can have malware if not vetted by Microsoft | 5 |
| Linux | Client Side | Linux is an open-source OS with many different distributions. It is the cheapest option in the market, and it can run on both Windows and MacOs. | -Open-source communities that enable different ways of building a client that fits the need of the application being built.  -Cheapest way of running Desktop apps in both Mac and Windows. | -Linux is not widely used as an OS and requires a user to install Linux on their device before being able to install a Linux client. | 3 |
| Server Side | Linux is open-source kernel that is used by the open-source community of developer to create distributions that have different use cases. Making Linux very versatile for server management, and because is open-source it is free to be used by the majority web hosting services. | Open-source community with many active developers with extensive knowledge on the distributions available  Free since it is open source.  Most web hosting services support Linux distributions.  Linux distributions provide in-depth access to the kernel allowing server managers to have full control of the servers and the network.  It also implements a strict user privilege model and offers a selection of built-in kernel security defenses to safeguard against vulnerabilities and attacks. | -Difficult to find vendor support for distributions.  -Apps built by Apple or Microsoft are not supported.  -It is usually more difficult to learn and in the commercial market is not the most popular OS. | 5 |
| Development tools | Development tools for Linux are usually not very user friendly but because they are open source you get all the functionalities at no cost. There is a Linux development tool for every solution that is needed but it requires downloading different tools instead of having 1 tool that does it all like developer tools sold by Microsoft and Apple. | -No cost  -Tools are usually specialized so many are required but each lets you do very in-depth programming. | -Large learning curve for each tool  -Most tools are not very user friendly  -Tools are not available outside the Linux OS. | 3 |
| Android | Client Side | Android is an open-source OS that is optimized to be used by mobile devices. Android was developed by Google, and it is now the most used OS due to its open source. | -Open-source which means there is no cost on using Android OS on the client side.  -Kotlin based on Java has been refactored and optimized for mobile development.  -Kotlin is also supported by web browsers, helping with cross platform development.  -Can be deployed with a Linux server. | -Is not the most secure of clients, and it’s susceptible to malware found on the internet.  -Not supported by the .net framework or MacOS proprietary distributions. | 4 |
| Server Side | Android OS is not a server OS. It is not supported as a server OS by Web Hosting services, and Google opted to use Linux as the OS for their servers since Android OS wasn’t designed for that. | -No advantages | -Not secure  -Not supported by web hosting services  -Does not provide in-depth access to the Kernel and other components of the servers. | 1 |
| Development tools | Android studio is the main IDE for developing Client-side applications using OS for mobile devices. Android studio lets you download emulators and use Kotlin to build efficient and secured mobile applications. This is an open-source IDE since Android OS is open-source. A license to publish into Playstore is needed. | -Free development tools like Android studio are available and many plug ins have been integrated supporting multiple programming languages  -Emulators available for free and Android devices are the most popular.  -Can be downloaded in Windows and Mac OS. | -It depends on the JVM so it needs the appropriate hardware resources to operate. | 4 |
| iOS | Client Side | Apple iOS is used only by Apple mobile devices. It is considered very secure, and it is not open source like Android OS. A license to publish apps for iOS needs to be purchased. Swift is the programming language Apple created for iOS development. | -Consistency across Apple devices.  -Secure.  -Can be developed with a Linux server. | -Only available for Apple mobile devices.  -Requires a license for publishing. | 4 |
| Server Side | iOS cannot be used for server unless an iPad is used as the actual server. Apple offers servers only through the Apple mini which runs an apple server OS with similar feel to ios. | Can use iPads as server units using ios server OS | -Requires having an iPad. | 2 |
| Development tools | xCode is the IDE built by Apple that enables developing an ios client. Swift is the programming language Apple created for ios development. | -The xCode IDE provides emulators for Apple devices.  -xCode IDE enables Swift intellisense assisting development. |  | 5 |

## OS evaluation summary

When it comes to servers Linux is the preferred option because it is compatible with all the different clients that will require The Gaming room to expand beyond the Android application. In addition, Linux OS is the preferred OS because it is supported by the majority of web hosting services in the market, allowing scalability and flexibility. When it comes to Clients, the application will run web browsers if the device is a PC running a desktop OS, and for that reason it would be best to run a Linux web client since it will be supported by all desktop OS.

## Recommendations

* **Operating Platform:** A Linux distribution designed for servers is the recommended operating platform for the servers that will be powering the client application. Linux provides server administrators full control and access to the hardware and the network being maintained. Most server hosting services are compatible with Linux distributions and Linux distributions are for the most part compatible with the main web protocols. Linux based servers are also more secure, and server distributions allow server administrators to scale the network without affecting security.
* **Operating Systems Architectures:** The operating architecture for Linux uses a monolithic kernel architecture. In Monolithic kernel mode, operating system runs in a single address space. Monolithic kernel has all the operating system functions or services within a single kernel. This single kernel will run as a single process in a single address space in memory. Monolithic architecture enables higher performance however less flexible for modifications to add new features or enhance existing features.
* **Storage Management:** For storage management, *Draw it or Lose it* will be using a Cloud Based server where most of the data will be stored, and the clients will be depending on the storage management being used for the cloud server. Since cloud servers do not require purchasing hardware, making sure we are using a reliable cloud platform is key for the application to work across all clients. When it comes to managing data, media, and files, a crucial step to take is making sure that everything is properly named and follow an established naming convention so that information is easy to recall. Another crucial step when it comes to storage management is having a plan for backing up important files. Since the main storage for the application is going to be using the cloud for storing the majority of data, it is recommended to backup essential files, in a physical offline storage which will be automatically synchronized with the cloud storage so that not manual intervention is needed.
* **Memory Management:** Memory management is one of the most critical components when deploying a client-server application using cloud server because scalability will be limited by the server plan purchased to support the application. Applying memory management best practices is crucial, using cache, allocating and deallocating instances from runtime, using garba collection when available, and reducing the amount of processing the server has to use by using separation of responsibilities and leveraging the devices where the clients will be running.
* **Distributed Systems and Networks:**
  + **Architecture**: The *Draw it or lose it* application will consist of many moving parts that are spread out across multiple devices indifferent locations and regions, including client applications that send out updates to the *Draw it or Lose it* API end points, and components of *Draw it or Lose it* that can be scaled up independently of each other to meet load and performance requirements.
  + **Database**: Since most of the components for the Draw it or Lose it application are open source except for the Cloud server that will be utilized, the database recommended to use is MongoDB since it’s well supported by the community and Cloud server systems.
* **Security:** Protecting the confidentiality and integrity of the data is key to ensuring the system is not compromised. Standard ways of securing web-based APIs should be used, such as SSL/TLS (HTTPS). This ensures the clients are connecting to a known server, and the communication between them is encrypted, preventing man-in-the-middle attacks. Another way to build security into the application is by using abstraction, type specific programming, encapsulation of data, and by following the industry standards.