The Gaming Room

# Software Design Template

Version 1.3

## Table of Contents

[Software Design Template 1](#_Toc53360925)

[Table of Contents 2](#_Toc53360926)

[Document Revision History 2](#_Toc53360927)

[Executive Summary 3](#_Toc53360928)

[Requirements 3](#_Toc53360929)

[Design Constraints 3](#_Toc53360930)

[Rationale 3](#_Toc53360931)

[Evaluation 4](#_Toc53360932)

[Memory and Storage: 7](#_Toc53360933)

[Recommendations: 9](#_Toc53360934)

[References: 12](#_Toc53360935)

## Document Revision History

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 09/18/2020 | Ryan Stork | Creation of document and initial implementation |
| 1.1 | 10/03/2020 | Ryan Stork | Added Evaluation and References |
| 1.2 | 10/09/2020 | Ryan Stork | Added Memory and Storage Considerations |
| 1.3 | 10/12/2020 | Ryan Stork | Added Recommendations |

Instructions: Fill in all bracketed information on page one (the cover page), in the Document Revision History table, in the footer, 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 has requested the assistance of CTS to design a web app based on their Android app, Draw It or Lose It. They need our help to create the environment and to facilitate the development of the app.

## Requirements

In this project, we will work with The Gaming Room to develop a web app that serves multiple platforms. Each game will have one or more teams involved. Each team will have multiple players assigned to it. The Game, team, and player names must be unique to allow users to check whether a name is in use when choosing a name. Only one instance of the game can exist in memory at any given time.

## Design Constraints

The first design constraint is that the program must be a web app that serves multiple platforms. This means that the program should be a java-based program. Java can be used by multiple platforms for web services.

The second design constraint is that the game must have at least one team, with the ability to have multiple players involved.

The third design constraint is that game, team and player names must be unique.

A fourth design constraint is that there is only one instance of the game can exist in memory at any given time.

## Rationale

The first design constraint is a business constraint. The client has an Android app, but they would like to make a web app. Java apps can be run on multiple platforms and can be for web apps.

The second design constraint is a software constraint. We could create lists for the games teams and players to make sure that they are all filled appropriately.

The third design constraint is a software constraint. When we pass in a new game, team, or player name, we test through the current list to make sure that isn’t already in service. If it is in service, we will notify the user to pick a new name.

The fourth constraint is a software constraint. This can be fixed by using a singleton. This allows only one instance of the service at any one point in time. If another instance is tried to be created, we will just reference the original service.

## Evaluation

**Server Side:**

We are tasked with making this android game a web-based application that can be run on multiple platforms. This means creating a server-client relationship. Since this is a web-based application, no matter the platform, they will need to connect and transfer data to the server the same way. We are asked to evaluate the hosting the software on different platforms.

Windows is the most common web hosting server. Usually used by larger corporations. This server has better integration with other windows product. It also has remote desktop access if server maintenance needs to be done remotely. Typically, Windows servers tend to also come with paid technical support. The downside of this platform is the cost. Many small and startup companies do not want to pay the higher cost, especially if they aren’t using other Windows products in combination of this server.

Linux is another common web hosting platform. This is popular among smaller companies. There is a cost saving associated with using an open-source platform such as Linux. Linux systems are easier to modify as they do not have the constraints of Windows. Any application can be added or deleted, plus there is the benefit of using other open source programs with Linux. Linux systems generally use less resources. Thus, making them more efficient. If support is needed, usually open source support is cheaper. The downside to Linux systems is that when maintenance is needed, remote access is command line based rather than a Windows remote connection.

Mac also offers a server platform macOS Server. Like Windows servers, Mac uses integration with their own programs. This can be great for a person who has never run a server before and can quickly get his own started. For a company with an IT department, this is less important because a professional can be hired that knows how to set up a server. If the company is using windows devices, having a mac server doesn’t make much sense, and vice versa. The licensing costs of Mac are less than Windows but are still more than Linux.

**Client Side:**

When looking at implementation on multiple platforms, we must have consideration for software development. We must make a cross-browser compatible application. Different browsers render objects differently. We must try to avoid browser-specific code. In order to make sure that our code works on all browsers and mobile devices, we need to test on all browsers and mobile devices. With thorough testing, we can make sure that the application performs on all browsers and devices. This extensive testing takes time and can require expertise with specific browser issues. This is an upfront capital cost to the company, The Gaming Room.

**Development Tools:**

To develop an application that works on multiple platforms and operating systems, there are different development applications that can be used. Java can be used to develop the web-based application as well as work with android devices. Java is one of the most common languages used today. If Java will be used for those devices, then another language would be needed to develop for Mac and iOS. The most popular is Swift. Swift is an open source language by the Apple corporation for development of iOS apps. Swift can be integrated with XCode, which is used in Mac development. XCode is also free, while joining Apple’s developer program and uploading the app costs $99 per year.

An alternative is the programming language C#. C# can be used to build web applications. C# can also be used with Xamarin. Xamarin is a part of Visual Studio that allows developers to write C# code that is compiled to native iOS and native Android binaries (Chand, 2019). Using this language would reduce the number of teams needed to create an app that runs on all devices. The problem is that the number of Java and Swift programmers outnumber the amount of C# programmers.

Java, Swift and XCode are free to develop an app, but the cost for Apple Developers is $99 per year. For a commercial product using Java, the cost is $25 per month per processor for servers and cloud instances. This pricing started in 2019. The cost of this would be based on the number of servers using the Java web app. If we were to develop in C#, the cost of Visual Studio Enterprise is $250 per month per PC that is developing the program. There would be no cost for the web deployment as with Java. There would still be the cost of $99 per year for Apple. I feel the savings would be greatest for using C# if we have the developers for it.

## Memory and Storage:

When it comes to the management of memory and storage in this game, Draw It or Lose It, we must consider the platforms that we are running on. When we are using a server that is being accessed by different web browsers. For this example, we will discuss one game. All the players in the game will load the same picture into memory from the web page. The game will not begin until all players have the file loaded. Each file is approximately 8 megabytes in size. The average download speed in the US is 50.2 Mbps (Mendoza, 2019). A file would take just a couple of seconds to download at this speed. The global average internet speed is 11.03 Mbps (Hananto, 2019). At this average speed, a file would take roughly 8 seconds to download. Once player in the game has loaded in, the game can begin. This gives no player an advantage over the other. The picture can be rendered in as the program dictates. When the game is over, the file can be unloaded from memory, so that the next game can be played. This would require the least amount of memory for the user.

When using a server-based architecture, the storage for this game would all be on the server. The players will download the file from the server. 200 high definition files at 8 megabytes per picture would be 1.6 gigabytes of files. In relative terms of size for storage, this is not large at all. Depending on the number of players, the company could place virtual servers on one server for players to download the file. This would increase the number of players that could download while keeping the number of servers down. This reduces the cost of the game. If Linux servers are used, then there would be no fees paid to Microsoft as with a Windows server.

If the game is being run on an iPhone or Android phone, the storage may be different. When a game is on the phone, it is usually on the go. The game could have all the files downloaded on the phone and the entire storage could be on the phone. Then the 8-megabyte file could be loaded into memory from storage. There would be no additional downloads needed unless different pictures were added or changed. This would speed up the loading of the game. On the other hand, very few mobile games are that large. The files could be downloaded into memory just like web-based game. This would lower the size of the game.

The difference between memory and storage are typically dealt when the application is closed. If the file is on the hard drive, then when the application is closed (such as the web page or app on a phone), the files would remain on the disk. This storage would reduce the length of time that the game loads because a call from hard drive to memory is faster than downloading the file. Memory is volatile. When the power is shut down, the items stored on memory are lost. But the loading times from memory are much faster than from hard disk storage, which is why these images are either loaded into memory or downloaded into memory from storage. There is also much less memory in a system then versus hard drive storage, which is why we are properly managing memory by only loading in one file at a time. This reduces the overall amount of memory needed.

I would recommend that all files would be downloaded on demand from a company owned server. This would reduce the amount of storage and memory that would be needed from the clients. A game may take up to 20 seconds to load in, but if a progress bar is shown, this would let the user know that the app didn’t freeze and was still being loaded in.

## Recommendations:

These are my recommendations for bringing Draw It or Lose It by The Gaming Room to more platforms as a web-based game. The first recommendation that I would make would be for the operating platform. The operating platform is the system to which the clients would connect to for gameplay by use of web browsers. My recommendation would be to use Linux servers. These servers are great for smaller companies as they provide lower start-up costs to run. Programs can be added or removed at the owner’s need. Also, many open-source programs can be used that can be modified by the programmers at The Gaming Room. If the game’s popularity exceeded capacity, more servers can be added faster and at a lower cost than a Windows server.

The users will connect to the servers via internet browser or mobile platform. Once connected, they will join a game with other players or start a new game in which other players can join. Each player will have to enter a unique name to play a game. The architecture of the Linux Operating System is typically administered as a command line interface. Linux is comprised of four main layers: applications, operating system, kernel, and hardware (Havens, 2018). The kernel is the main brain of the operating system. When a request comes in from the users, the kernel takes that request, allocates the necessary resources for it to be accomplished, and provides that information back to the user from the network. When a network administrator would need to make updates to the server, they can take down the network interface while they update the applications and shared libraries, like the library of pictures.

The storage management for the game will be located on the servers that the users will connect. There are 200 high definition photos at approximately 8 megabytes per picture in size. This means that there are 1.6 gigabytes of files. It would be a bad idea to force the users to download the entire picture library when they connect to the web server while on a PC. When the game is running, the picture would be loaded into memory from the server. Working with a file from memory is much faster than from storage. PC gameplay is typically stationary, which means a stable, fast connection. This would allow the user more gameplay and less time waiting for downloads. When the game is over, the file can be removed from memory and a new file can be loaded in, if the users continue to play another round. I would allow the mobile version of the game to download the library of pictures onto the mobile device when the device is connected to WIFI. Typically, mobile is meant to be played on-the-go, which means that download times may vary. This would reduce the downtime of playing for a mobile player. If the picture library changes, then the user will be notified that they must update their library in order to play.

The storage management would also affect the memory management. In order to decrease the demand on memory, only one high definition image would be loaded in at a time. If multiple images were loaded at the same time, that would be unnecessary to the user because they are only going to see one image at a time. Decreasing the amount of memory usage in the picture allows the game to reach a greater amount of people who may not have access to a system that has a large memory demand.

The Gaming Room would like for this game to be interconnected through multiple platforms. This can be done by using distributed systems. A distributed system is one where the users are connected to each other through the server, where multiple servers are located strategically in the world to reduce latency and downtime. This connection will allow the users to communicate with each other. The dependencies between these different systems would rely on the server. If each user can communicate to the server, that information can be relayed to the other users. If a user is disconnected from a game, they would be removed, and the updates would be sent to the other users. It would also be important to have servers across the world. This would reduce the ping time for the users and make the game more playable. If a server went down for maintenance, then the users would be rerouted to their next nearest server. This would allow time for server maintenance, lost connectivity with a server, or fixing a crashed server, should the need arise.

Security is of upmost importance to all parties involved. The information stored on the databases would be only information that would be must have information. Personal information would be behind a password. We would not store the password in our system, but instead a hash of the password. When the user transmits their username and password, the system encrypts the password with a hash. If that hash matches, then the user can continue their gaming process. If the password doesn’t match, the user will be notified. If the user forgot her password, the network will send a password reset to their email on file. Any credit card information would be stored in the system under AES encryption. This would help protect from any malicious attacks. Direct access to any servers that store credit card information or transactions will be limited to database administrators that need access to that information. They should also have a background check to make sure that they can be trusted with that level of access.

## References:

Chand, M. 2/21/2019. *Best Programming Language for iOS App Development*. Retrieved from: https://www.c-sharpcorner.com/article/best-programming-language-for-ios-app-development/

Gewirtz, D. 4/8/2019. *Windows, Mac, or Linux? We compare the pros and cons of these computing platforms*. Retrieved from: https://www.zdnet.com/article/windows-mac-or-linux-we-compare-the-pros-and-cons-of-these-computing-platforms/

Hananto, A. 7/12/2019. *Rank of Countries with the fastest (and slowest) Internet in the World 2019*. Retrieved from: https://seasia.co/2019/07/12/rank-of-countries-with-fastest-and-slowest-internet-in-the-world-2019

Havens, K. 1/4/2018. *Understanding the architecture of the modern Linux operating system*. Cumulus. Retrieved from: https://cumulusnetworks.com/blog/linux-architecture/

Mendoza, N.F. 11/12/2019. *US cities with the fastest and slowest internet speeds*. Tech Republic. Retrieved from: https://www.techrepublic.com/article/us-cities-with-the-fastest-and-slowest-internet-speeds/

Volico. 4/17/2017 *Linux vs. Microsoft Windows Servers*. Retrieved from: https://www.volico.com/linux-vs-microsoft-windows-servers/