

Draw it or Lose it

# **The characteristics, advantages, and weaknesses of various platforms!**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 04/302021 | Michelle Redmond | This is the third draft. This draft is for Draw It or Lose It. (Evaluate the characteristics, advantages, and weaknesses of various platforms) |

**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](#_sbfa50wo7nsh)

Chat Away is the company we are currently working with. They are a social media company seeking to expand their consumer platform. Chat Away has a website, so they want to create an app for both IOS and android users.

## [Design Constraints](#_2et92p0)

System Constraints (Technical)-

This is an obvious constraint because Chat Away wants to use IOS high quality editing and they exclusives use windows. This could cause issues in the performance of the app. This needs to be a user-friendly application.

Business Constraints -

JAD group could be created to ensure the same features and account for the nonfunctional requirements. The process could take longer because they are building one app for two different platform. Both constraints will be costly. This project needs to be completed within budget.

Technical Constraints-

The app should have the same structure as the website. It should include all the functionalities as the current website. There will be many tests runs to ensure it runs without hiccups.

## [System Architecture View](#_ilbxbyevv6b6)

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](#_8h2ehzxfam4o)

The system creates a link between Game, Team, and Player class. This means that they all receive the same message from the system. The UML can show these connections. Every class in the system will share “name” and “ID”. This makes the system become a classified as a super class. Team and Player has a type: Game has a team and Game service has game. The has a refence to an instance to another class. When we look at the system layout Game Service refers to Games. Games refer to Team, and Team refers to Player.

****

## [Evaluation](#_2o15spng8stw)

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** | **Pros –**  Secure security and perceptive malware integration.  Easy integration process for Mac users.  The inference is simple and cleaner allotting for features that are dedicated to multitasking.  **Con’s**-  Only Mac users can use this server.  Limited graphics and software  Mac cannot read windows format.  -  *OS has an adjustable terminal command that build the server, access, and alter.* | **Pros-**  Free open-source solution  Quick and easy to modify.  Available to the public  Only pay for the set up of product and not the operation.  Linux can use any software. Consider more reliable because Linux has fewer resources and is more efficient. It is develop in C language.  **Con’s-**  Linux is a command line-based software.  Does not offer support within the platform.  Not easily managed remotely.  *Mac OS and Linux share the same features, but this is more cost friendly.* | **Pros-**  Windows offer better support solutions.  Windows offers more features than Linux along with a more thorough solution that is built into the software.  Easy to mange remotely and it comes with a remote desktop.  Remote desktop includes graphic user’s inference.  Windows is more featured oriented.  **Con’s-**  Month to month fees  Does not work with any software.  Stability problem  More resources are needed to run windows.  Windows has a pattern of rebooting.  *Windows has more adaptable operating system. This means there is a lot of software readily available for use.* | **Pros-**  Increase in productivity. (Can reduce time and cost)  Portability- You can access vital information anytime (Not restricted to a certain location). Cloud Computing  Easy to Use.  **Con’s-**  Connectivity issues  Security issues (Due to the ability to connect anywhere).  Short Battery life  Limitations on downloading, file storage, and screen size.  *The sever should be immobile, tracking at a single place.* |
| **Client Side** | **Pros’-**  Client would have a longevity and performance. They would experience a little to none virus and security breaches.  **Con’s**-  This would be time consuming.  The clients would have to purchase a MAC. This will be relatively expensive because of specific hardware. They would have limited access to certain software because MacOS is built to only read Mac’s. User would also be confined to the hardware on the computer, you cannot upgrade the system.  *They should have a moderate amount of expertise. The cost is like Windows. They need to make sure the development of the app is compatible all web platforms and mobile devices.* | **Pros’-**  Client would be able to save money because of the flexibility of the platform. Flexibility also allows for a faster completion.  **Con’s-**  Client would have added support.  They would have minimum features and specifications.  *They should have a maximum amount of expertise. They need to make sure the development of the app is compatible all web platforms and mobile devices. The cost is minimal.* | **Pro’s-**  Client would have built in support.  Windows would be easy to manage remotely. The client would also receive more features too.  **Con’s-**  This would be an expensive option, because client would be responsible for integration and services for products while in use. Windows would also take up a considerable amount of space because of the hardware. Windows also has an compatibility problem too, which means this would be time consuming.  *They should have a minimal amount of expertise. The cost is just as much as Mac. They need to make sure the development of the app is compatible all web platforms and mobile devices.* | **Pro’s-**  Client would have portability and cloud computing. This would decrease time and be inexpensive.  **Con’s-**  Client would have connectivity issues and security issues.  Client would also have limitations on viewing, storage, and internet speed.  *This gives consumers more flexibility. While at the same time making updates fast and singular. This is a compressed version so its going to be harder to build than the remote versions.* |
|  | Mac has limited flexibility within its software. This means that only certain servers can be purchased. Teams would need to be developed to ensure the integration process is properly done. There will be licensing fees associated with this platform because windows would need to be intergraded. This system recognizes most languages.  *Macs typically run swift when referring to languages. Mac is not limited to the use of language or libraries.* | Linux is flexible and will not require a team to be assembles for integration. Their will be no extra cost of licenses. This server also recognizes most languages too.  *Linux uses with numerous languages such as eclipse and javas. The system has*  *cool tools easy use like MacOS. Linux is not limited to the use of language or libraries.* | Windows has limited flexibility within its software. This means that only certain servers can be purchased. Teams would need to be developed to ensure the integration process is properly done. There will be licensing fees associated with this platform because windows would need to integrate their software and apps. This system recognizes most languages.  *Windows aim to be user friendly and simple. Eclipse and notepad++ are easy tools to use. You can use all languages on windows its not limited, just like its counterparts.* | Mobile computing would cause for many teams because of the limited software options. The licenses cost would be the most expensive here because of the variety of technology. This server also recognizes most languages.  *Android and Swift can be running by all three machines, making all sorts of apps at its disposal. The app will not be limited to any languages.* |

## Recommendations

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

1. **Operating Platform**: Gaming Room should begin with “Windows” because this is their operating system. Windows also has more software available, and it needs minimal expertise. It will not cost as much if the project is started with this operating system. There will also be numerous IDE’s to work with. I would recommend Linux. It is universal and it will work on most platforms. Plus, it is inexpensive to incorporate.
2. **Operating Systems Architectures**: Although, Windows grants service used by all window-based applications to show Graphical User Interface, while connecting all resources. The graphics and multimedia, messaging, and webservices. These can be used in user account or server specific. But because Linux is the recommended choice and developed in C language. Linux is an open-source OS. The Kernel plays a major role in this system because it allows to run more applications than any other platform. It is governed from a command line (Shell). Linux kernel consists of scheduler, drivers, security, and networking. (Havens, 2018)
3. **Storage Management**: Windows offer a variety of storage options such as storage sense, app specific locations, and the use of a cloud. Linux will use block storage. Block storage is a piece of hardware that can be used to store data. It is fixed sized blocks or chunks of space. This space will act as an extension of the filesystem tree. The disk portion that would be appropriate would be a GPT. A GPT is a form of partitioning system.
4. **Memory Management**: A data base or library should be created. This allows you to keep whole projects together. This includes working with IDE and files that help create the application. Linux has a virtual memory, demanding paging, and memory allocation. It has its own jargon as well.
5. **Distributed Systems and Networks**: Linux is available to ordinary users. It has symmetric multi-processing capabilities. Linux uses DIPC is a software only solution allowing for people to be able to build multiple and program multiple computers. This supports the message passing and the distributed shared memory paradigms of distributed programming, providing more options for the application programmer. (Sharifi, 2021)
6. **Security**: Linux has exceptionally good security features. Linux has a feature called Parrot-Linux. This feature provides numerous security tools. The most important feature is that Linux is an open source.