

# The Gaming Room

# **Draw It or Lose It**

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 | 05/20/21 | JonKayla Pointer | Initial |
| 2.0 | 06/06/21 | JonKayla Pointer | Evaluation Review |
| 3.0 | 06/15/21 | JonKayla Pointer | System Architecture |

**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)

The Gaming Room wants a web-based game. This game needs to serve multiple platforms and it is based on a current game, “Draw It or Lose it.” The current game is only on Android OS devices.

## [Design Constraints](#_2et92p0)

The Gaming Room would like to expand their gaming application to multiple platforms. This application is web based. There must be a server-style configuration for hosting the website and it must have scalability. There will be different client side applications to account for the players on iOS and Android mobile platforms. The application should have uniformity and look similar across all platforms. There must be a modern responsive HTML interface running for the desktop clients (Linux, Windows, and Mac). Each application will be able to9 communicate with the web server.

## [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)

Entity is the parent, or Super class of the classes Game, Team, and Player. Game, Team, and Player inherit the attributes of the parent class Entity. Therefore, they all have the attributes id and name. The instance of one class has a reference to an instance of another class. Game Service has reference of Games and Games has a reference of team. Team has a reference of Player. So, Game Service can have multiple games, each game has teams, and each team has players. This portrays aggregation.

****

## [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** | MAC OS servers provide easier creations of different factors for the servers and mac clients. It is easier than Window group policies. The GUI is simple to use. It is more difficult to scale up into larger environment. | Linux is reliable and provides stability. It is less susceptible to cyber-attacks. It is easy to install, and it has better scalability than  MAC OS Server. The power command prompts make it easy to install different repositories and packages. Linux is open source and requires more advanced skills to use. | Windows Servers are easy to use and configure. Windows Server’s has great support. It has great scalability. The GUI is easy to use. The license cost can get expensive. | After programming the backend code, you would need to find a provider to host the mobile application. |
| **Client Side** | Clients are limited on using MAC OS devices. MAC can be more expensive than other OS. | Linux is an open-source OS. It is powerful but for advanced users. It is not restrictive with the OS for clients. | Windows makes it easy for developing websites and Microsoft has many applications that work well on Window OS machines. | There are different OS for mobile devices. Right now, the gaming Room only has an application for Android OS mobile devices. Coding for other mobile OS such as iOS would expand the field of users. |
| **Development Tools** | MAC OS uses C++, C, Swift, Objective C, and Assembly Language. MAC OS pairs well with XCODE IDE because it is also Developed by Apple. | Linux is an open-source Unix-like Operating System. It was written in C and assembly language. Linux is power OS and used by a lot of Network Admins and penetration testers. JetBrains is a good IDE for Linux. | Windows OS was written in Visual Basic, C , and C++. A great IDE for windows is Visual Studio. Visual Studio was created by Microsoft. | iOS devices use SWIFT and C, C++. A great IDE to develop iOS applications is XCODE by Apple.  Android OS devices was written in Java, C, and python. Many applications are written in java. A great IDE to use is Eclipse or Visual Basic. |

## 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**: I recommend the Gaming Room to use Linux servers. I suggest creating a mobile app for iOS device to expand the user field.
2. **Operating Systems Architectures**: Although Linux is a little bit more advance its cost-friendly, reliable, and it is opensource so they can do more with Linux. It has great scalability.
3. **Storage Management**: Linux does not require much for storage. I recommend implementing IaaS. IaaS provides “high-level APIs used to dereference various low-level details of underlying network infrastructure like physical computing resources, location, data partitioning, scaling, security, backup etc.” IaaS is a type of cloud computing service. It is great for storage management. Microsoft Azure is a great cloud computing service.
4. **Memory Management**: Linux is fast and powerful. The memory management is great. Using IaaS is also great for memory management. This is because it provides storage, backup, and recovery with simplifies the management of the systems.
5. **Distributed Systems and Networks**: It is important to use different programming languages to ensure that the applications for mobile devices work on the different operation systems. Being able to use a server such has Linux Servers that is cost-effective and provides good scalability will ensure that the application can connect to more clients. As more clients expand it is a good Idea to have multiple servers to ensure the availability of the website. Using a load balancer is a great idea so it can optimize the response time and avoid overloading. The use of RAID is great to provide redundancy for memory and storage. I recommend using RAID 5 for parity and striping. If there is a single drive failure RAID 5 can withstand it. Or they can use RAID 10. It is a nested RAID configuration. It is good too because it provides security by mirroring all the data on a second drive and uses striping to speed up the data transfers.
6. **Security**: When programming the front-end portion of the applications it is very important to keep security in mind. It is important for programmers to ensure that the check user input to avoid cyber attacks like SQL injection, or buffer overflow. C++ is prone to buffer overflow attacks, so it is a great idea to keep that in mind. SQL injections is where a user with malicious intentions can exploit information of database. They do this by using the places where the application inputs data. My recommendation is ensuring the both the application and server side uses the best coding practices to avoid exploitation. Before the site deploys, they need to obtain a root certificate from a trusted Certificate Authority. This is a public key certificate that verifies that this is the right website. This ensures that the website can be trusted. Another thing to keep in mind is proper encryption. This ensures confidentiality. It is important to use TLS for port 443. The Client-Server applications should use TLS protocol to ensure confidentiality of the data being transferred. Do not use SSL because SSL is prone to the man-in-the middle exploit called the POODLE attack. Also use encryption algorithms for data like AES 128/ 192 /256. The use of hashing algorithms is a must to ensure DATA integrity. I recommend using SHA 256 or SHA 3. The use of MD5 SHA 0 and SHA 1 should not be used. They are prone to co to collision attacks, such as birthday attacks. To prevent Zero Day attacks, we must implement efficient design, coding a testing practice. Using a threat model is good practice because it provides a framework on which to implement on security efforts. The Gaming Room needs to use a SIEM (Security Information Event Management) System to manage security alerts. It is a good idea to implement Firewalls, IPS/IDS. They can also investigate getting Security as a Service. I suggest using vulnerability scanners to ensure the security. Providing Honeypots and Honey Nets is also good to essentially trick a malicious user. This can provide forensic information on how a malicious user was able to exploit the system.

**Server side:** After a review of the options of servers that The Gaming Room could implement, I recommend using Linux servers. Each operation platform offers a server-based deployment method, and the website can be hosted from it. Using the “REST API”, or Representational state transfer, a software architecture style uses interactions with web services. This enables users to make request with a RESTful API. This information uses formats via HTTP: JSON, HTML,, Python, PHP, or even lain text. Using a Linux server is great option for this because it is open source. This means that Linux is cost-friendly. It is very stable. You can modify a system or change configuration with out having to reboot the server. Linux has a variety of security mechanisms to secure files from malicious attacks and abuses. You can use SELinux or Security enhanced Linux. This limits the resources a service can access. Some downside of Linux is that system support like hardware drivers is not easily available. In addition, Linux is more for advanced users. Users must have knowledge in the command-line interface to unlock the full potential that Linux has to offer. Using Windows and MAC servers require licensing -fees and is more expensive.

**Client side**: The cost of developing different client-side applications is more. This is because different platforms must be programmed specifically to that device’s OS. Different platforms require different programming languages. The client-side applications need to have uniformity amongst each other and successfully access the sever. When programming these applications there must be a plan developed to account for the goal, the time allotted for each phase, the security requirements, testing and validation of the applications, and time to fix any issues before the scheduled date to deploy the application. We need programmers with a diverse knowledge in different programming languages to account for the different OS. This will ensure that the app will be compatible with all web browsers and mobile devices.

**Developmental tools:** The languages that developers must be familiar with is Java for android client-side applications, objective-c and swift for iOS client-side applications, and HTML, JSON XML, or even SQL. SQL can be used to manage a database. XML is markup language that defines the rules for encoding documents. JSON is an open standard file format and data interchange format. In REST API request are made to a resource and use HTML, XML, JSON, or other formats for responses. Developers need to have knowledge in this language to successfully create a REST API application. For website applications a developer should be familiar in HTML, CSS, and JavaScript. JavaScript is the client-side scripting where has PHP is the server-side scripting language. So, PHP is also needed. There will be teams needed for the Server-side applications to take requests from the client side and authenticate users, and teams for the client-side applications: for the web browsers application, the applications for android OS devices, and the applications for the iOS devices.

**IDE for Android App Development:**

Source:

<https://www.intelvue.com/10-best-ide-for-android-app-development/>

**1. Android Studio:** is a great programming tool that google recommends! It has high-level features to facilitate work for developers. It even has emulators.

**Languages ​​Used: Java, C, C ++, Kotlin, XML**

The advantages include:

* Built-in support for Google Cloud
* Emulator for devices
* Interface for all devices(tablets, tv, etc)
* GIThub integration
* UML diagram support
* Free
* Provides live change support

The disadvantages include:

* Offers restricted language support
* RAM usage is high
* Target system is only Android OS

**2. Eclipse:** is a free IDE used for application development and a variety of languages.

**Language used:** **Java, C, C ++, C #, JavaScript, Python, Perl, PHP.**

The advantages include:

* Free and open source
* Compiler runs fast
* Wide selection add-ons
* Quick testing
* Multiple language support

The disadvantages include:

* Not always reliable

**3. InetlliJ IDEA:** is an integrated development environment. It was developed by JetBrains. It is used to develop software with both Apache 2.0 license and commercial version. It has support for many languages.

**Language used:** **Java, Scala, Groovy, Kotlin, JavaScript, TypeScript, SQL**

The advantages include:

* Powerful debugger
* Clear and detailed documentation
* Support for many languages
* File corruption is barely experienced
* Offers support for Android and JavaEE

The disadvantages include:

* Shortcut keys are difficult
* Lack of plugins
* Errors are not properly dealt
* Some parts are closed source.

License fee is $499/year

**IDE for iOS App Development:**

**1. XCODE:** is a great programming tool that Apple developed! It has a user-friendly interface and application testing. It includes Swift 5.3 and SDKs for iOS 14, iPadOS 14, tvOS 14, watchOS 7, and macOS Big Sur.

**Languages ​​Used: C, C++, Objective-C, Java, Apple Script, Python, SWIFT**

The advantages include:

* Application testing
* Assistant Editor
* Friendly User Interface
* Schemes feature
* Application Testing
* Updates automatically
* Test multiple architectures

The disadvantages include:

* License fees for the Apple Developer Program
* More limited to Apple development
* It only runs on Mac OS

**2. AppCODE:** is a great programming tool that wasd created by JetBrians! It is bult on the JetBrians’ IntelliJ IDEA platform..

**Languages ​​Used: C, C++, Objective-C, Swift**

The advantages include:

* Coding assistance
* Code analysis
* Unit testing
* Supports modern C++ standars, libc++, and boost
* Supports JavaScript ,XML,HTML,CSS, and XPath
* Rename refactoring
* AppCode Integrations from CocoaPods, XCODE

The disadvantages include:

* License fees

**IDE for Web Development:**

**1. Visual Studio Code:** is a great programming tool that is a source-code editor that was made by Microsoft. It is for Windows, Linux, and Mac OS. It is great for building and debugging web and cloud application.

**Languages ​​Used: C, C++, Java, JavaScript, Node.js, Python**

The advantages include:

* Can be extended with extensions
* Operates as a language-agnostic code editor for any language
* Code can be synced between the editor and the server
* Syntax highlight, Bracket matching code folding and configurable snippets
* IntelliSense for JavaScript, TypeScript,JSON,CSS and HTML
* GITHub integration

The disadvantages include:

* RAM usage

**2. PHPStorm:** is a cross-platform ID that created by JetBrians! It is compatible with Mac, windows, and linux.

**Languages ​​Used: PHP,HTML,JavaScript,CSS**

The advantages include:

* Visual debugger
* Error highlight
* Databases/SQL
* Command line to9ols
* REST Client
* Unit testing

The disadvantages include:

* License fees

There are many web frameworks that exist that provide tools and libraries. This is great because it can simplify web development operation. Web frameworks can generate server-side code to work with requests and responses. Web frameworks can make scalability easier. Another think to consider is caching in the cod or in the server. Web frameworks have support for this. Caching is great for optimizing the storage of web response. Web security is also very important, and some frameworks provide support for handling web attacks. There are many web-frameworks that could be used. A popular one in Django. This is written in Python. It encourages raid development in a clean design. It is fast, secure, and scalable. For example, Django has templates that can protect against XSS or Cross Site Scripting. This attack allows an attacker to inject client-side scripts through the website. Django protects against these types of attacks because it uses automatic HTML escaping. This means that Django replaces special html characters into different text, so it is harmless. For example, the < is converted to &lt; This is just an example of what Django does to be secure. There are more ways that Django is secure. For example, Django also provides protection against CSRF or Cross site request forgery or where an attacker executes actions using credentials of another user. This is important in the Application for the Gaming Room because they may want to include in-app purchases using credentials of a user. More information of Django’s security can be found on their website, <https://docs.djangoproject.com/en/3.1/topics/security/>. Some popular websites and apps that use Django today are Instagram and Pintrest.

**System Architecture:**

**Operating Platform:** Recommend an appropriate operating (server) platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments.

An appropriate operating platform that will allow the Gaming Room to expand their App is Linux Servers. According to CompTIA, “Linux scalability is inherent to Linux and comes with every GNU Linux system by default.” This means the software and package manner that Linux has makes it easier to scale or expand to other computing environments. In addition, in the kernel 2.6 of Linux there are scalability features that are built in. Linux also has support for hyperthreading. This is the capability of using two virtual processors on one physical processor. This enables a Linux server to expand to more computing environments. Linux is also open-source, so it is open to the public and cost- friendly. A web-hosting company only must pay for the technical support to install Linux. Linux is also stable because if you make any changes to the configurations, you do not have to reboot the server. This is good because if clients are constantly connecting to the server, then The Gaming Room would want to limit any interruptions to the server. Linux servers also provide security. You can use firewalls, TCP wrappers, and even SELinux that all helps to secure files and services. Linux is also flexible because you can construct programs that let you create new functions and tools that lets you interact with the kernel. This enables you to have control over the entire Linux Server.

**Operating Systems Architectures:** Describe the details of the chosen operating platform architectures.

Storage Management: Identify an appropriate storage management system to be used with the recommended operating platform.

We can implement the use of TPM and disk encryption. This will protect the storage devices. In addition to Storage Management, we can either use storage devices that are Hard Disk Drive HDD, or Solid-State Drive SSD. HDD is a data storage device that uses magnetic storage and one or more rigid rotating platters to store and retrieve data. SSD is a storage device that uses integrated circuit assemblies. SSD are faster in terms of when it boots a system. It takes seconds to boot off an SSD and HDD requires more time. SSD is more expensive than HDD. A good way to implement Storage Management is using RAID. This is helpful fir reliability, availability, performance, and capacity. RAID 10 creates a striped set and a series pf mirrored device. This is good because is sustains multiple drive losses. RAID 0 uses stripping. It is good for applications that require a high performance, such as a gaming application. If we store data on the server side, we need to have multiple units and implement RAID to maximize storage management.

**Memory Management:** Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.

Software applications use memory caching to temporarily store data on RAM. The gaming application uses Memory management for short term data like the gaming instances itself, and the pictures and visuals displayed through out the game temporarily. A game can use memory allocation. It can either use Static or Dynamic. A game like Draw It or Lose It could use Garbage Collection to automatic memory management. It is also safer to use Stack allocation rather than Heap Memory Allocation. In Heap Memory Allocation the memory is visible in all the threads. Stack Allocation is also faster than Heap Memory Allocation. The data is automatically deallocated and flushes out in Stack Allocation. Using Stack Allocation is great because the Application will be able to render and display pictures at a rapid rate to meet clients need.

**Distributed Systems and Networks:** Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).

Using the RESTful API uses a good architectural style that allows interaction with RESTful web services. It uses the Client-Server Architecture made up of clients, servers, and resources. There is Stateless client-server communication. No client information would be stored in the get requests. There is also a layered system that organizes the servers. If we want a successful system, we need to be able to provide Availability of the services. Using RAID for Storage management is great to ensure that the data is always available. In addition, using load-balancing and multiple Linux servers will optimize the client requests. Having multiple Linux servers can also provide failover clusters or a way to switch to a redundant system. Using Other servers like AAA servers for authentication, authorization, and accounting is useful to implement in the network. This is a great because it separates different areas of task to make the entire system successful. Having one server do it all can create a single point of failure.

**Security:** Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.

There are various ways to protect user data and information of the system. Implementing secure coding techniques and creating a plan for identifying possible threats is a great way to keep data secured. Identifying a threat model will help programmers check what they implemented. Programmers can implement input validation on client side and server side. This will prevent any injection attacks or overflow attacks. Using AAA servers or proper authentication and authorization will ensure confidentiality to the data. Using secure practices in networking communications like using TLS or port 443. This is the cryptologic protocol that ensures that the communication between the client and server is encrypted. Using other encryption methods is great when securing data. The current encryption standard is AES. Before the application is deployed it should be tested and validated to prevent zero-day attacks. When updates and patches are available, we can use hashes and checksums to ensure the integrity of a file. SHA 2 is more secure than MD5 because MD5 has shown to have vulnerabilities. There should also be firewall in the network system. The use of o DMZ in the networking system can also provide security. There can be a perimeter firewall, DMZ, and an Internal firewall. You can also use NIDS/ NIPS or Network Intrusion Detection/Prevention Systems. HIDS/ HIPS can also be used, or Host-based intrusion detection/prevention systems. NIDS and NIPS can be put in the DMZ or at network borders. They capture or can prevent network traffic. Using NIDs/NIPS and HIDs/HIPS provide another layer of security.