

<Name-of-Software-Application>

**CS 230 Project Software Design Template**

Version 3.0

**Table of Contents**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0  2.0  3.0 | 3/23/25  4/6/25  4/20/25 | Ronald Green  Ronald Green  Ronald Green | Updating information  Adding more information  completing recommendations |

**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 cost, time, and expertise required to support multiple types of clients will vary depending on the specific platforms and technologies that are chosen. However, in general, it can be expected that supporting multiple platforms will require more resources than supporting a single platform.

One of the biggest challenges of supporting multiple platforms is ensuring that the application is compatible with all of the different web browsers and devices. This can be a time-consuming and complex process, especially if the application is complex.

Another challenge is ensuring that the application has a consistent user experience across all platforms. This can be difficult to achieve, as different platforms have different design conventions and user expectations.

**Requirements**

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

**Design Constraints**

To ensure that the application is compatible with all web browser platforms and mobile devices, it is important to use cross-platform development tools and frameworks. Cross-platform development tools allow developers to write a single codebase that can be compiled for multiple platforms. This can save a lot of time and effort, as developers do not have to write separate codebases for each platform.

In addition to using cross-platform development tools, it is also important to test the application on a variety of different platforms and devices. This will help to ensure that the application is compatible with all of the different web browsers and devices that it will be used 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**

A structured gaming system exists within the UML diagram model. All classes that inherit from ⁠ Entity ⁠ share standard properties (⁠id⁠,⁠ name⁠) between them while reducing duplicate data definition. ⁠ The GameService employs Singleton Pattern design to control game management through a solitary object instance. Regulation between the ⁠ Game  class and the ⁠ Team  class exists as one-to-many and ⁠ Team  connects to several ⁠ Player  objects. ⁠ The system executes under control of ProgramDriver and SingletonTester runs its tests. The structured hierarchy enables effective organization which simplifies game and team and player management by enabling secured access to data through encapsulation methods



**Evaluation**

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** | Flexible terminal commands to configure the server, access, or make changes.  Characteristics It is popular in web hosting  Advantages It is upgradeable, it has various options for different web hosting requirements  Disadvantages It is less preferred for web hosting services | The same goes for mac plus more cost-friendly  Characteristics Secured, most preferred.  Advantages Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  Disadvantages It is more difficult to find applications to support the web hosting required needs. | More software available compared to other OS.  Characteristics It is dominant to the other platforms. Close platform  Advantages High resource requirements, less loading time, high comfortability  Disadvantages easy virus susceptibility, poor tech support | It's better if the server is immobile and can be tracked in a single place. Specifications are better in other devices.  Characteristics More popular, high portability.  Advantages Have a wider reach, better compatibility, cost-effective  Disadvantages It is highly selective to various smart mobile devices Poor security |
| **Client Side** | Moderate expertise and time required. Cost similar to windows. | Maximum expertise and time required. Minimum cost. | Minimum expertise and time required. Cost similar to mac. | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices |
| **Development Tools** | When running languages on macs we can run swift the more popular option. While mixing in nice tools like notepad++. Though Macs can run all languages. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Linux can work with visual studio, eclipse, along with notepad++ for a nice and easy-to-use tool. Along with many more languages and tools. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Easier to use than Linux but can run the same as it. So visual studio, eclipse to name a few of the many languages. And with multiple tools notepad++ is a simple to use the tool. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | You can create countless apps using android and swift. Both languages and software can be run on all three machines. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. |
| **Licensing Costs** | The Mac App Store offers Xcode for no charge, which lowers the cost of licensing the development environment. Costs associated with Mac hardware, though, can have a big impact. | The majority of Linux development tools are open source and cost-free, which lowers licensing expenses. However, if necessary, there can be fees for support or enterprise-level tools. | There are several editions of Visual Studio, including the commercial Professional and Enterprise editions in addition to the free Community edition. Depending on the edition and team size, licensing prices can change. | Downloading Xcode and Android Studio is free. When developing mobile apps, using third-party tools or cloud services may result in licensing fees. |

**Distributed Systems and Networks:** Because each operating system being different I investigated ways to publish the game to run on all dives. I found Develop 4 which enables cross-platform game creation. An IDE that can be run on any device. Once the game is created you can simply export the game file into the web, iOS, Android, and many more options that will allow cross-play. This will help with dependencies. To prevent other problems like outages or connectivity, the company will need to make sure their servers are strong enough to support large player volumes along with backup power for power outages**.**

**Recommendations**

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

* **Operating Platform**: Adopt a web-based cross-platform architecture powered by technologies such as Node.js for backend development and React.js for the frontend, combined with Cloud Services (AWS, Google Cloud, or Microsoft Azure).
* **Operating Systems Architectures:** Windows OS is a hybrid operating system, taking in elements from the monolithic and microkernel designs. This helps it to make use of its resources more efficiently and at the same remain stable & secure.
* **Storage Management**: Use a cloud-based database system such as Amazon RDS (for relational data) and Amazon DynamoDB (for NoSQL requirements).
* **Memory Management**: Windows Memory Management: Windows memory management is a blend of paging, virtual memory and segmentation. This helps the system to run big applications by swapping their data between memory and disk whenever required.
* **Distributed Systems and Networks**: Distributed System Design: Can be done through client-server model using a RESTful API. Game logic is processed on the server, and clients (devices of players) interact with it via API calls.

Networking Decisions: With the use of a Content Delivery Network (CDN), latency will decrease as game data is cached in servers located closer to your players. All of this to bring owners faster loading times and smoother in-game performance.

Dependency Management: Use microservices to compartmentalize the game into smaller services that are easier to manage. It makes upgrades and scaling easier, you have better fault isolation.

**Security**: Deploy TLS(Transport Layer Security) for data on move and AES(Advanced Encryption Standard) for at rest. These are encryption methodologies that the industry has adopted to protect information from being accessed by those who are not authorized.