

Draw It or Lose It

# **CS 230 Project Software Design Template**

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 | 11/29/20 | Steve Barnes | Introduction of Subclass Entity and inherited to other Classes. |

**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 application/game design comes with the ability to have multiple teams per game played. Each game can have multiple teams and each team may have a singular to several different players per team. Each player and team can choose names unique to said team and player with the ability to check if it is already in use. Other challenges to be met and overcome are end-user/client security, improvements to software after consumer purchase, bandwidth requirements and challenges to multi-platform systems in terms of hardware and software.

## [Design Constraints](#_2et92p0)

With any web-based games, some of the challenges come to mind are crossed-platform design constraints, security, minimal bandwidth requirements per end-user, client screen resolution (both software and hardware) and multithreading efficiency challenges because of the mass improvements in multi-processor systems.

Digital signal processing applications (DSPs) and Application Specific Integrated Circuits (ASICs) are well known because of their ability to “perform computationally intensive signal processing tasks.” When this method is employed, future improvements (software releases) may be pushed out to end-users via network. For this to happen, systems will have to be designed with Embedded Systems (“those that are used to develop concurrent software and hardware”) to meet any future challenges previously unseen. (Ronkainen 2003)

## [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)

In the UML class diagram, Entity has a relationship with Game, Team and Player. Team and Player both inherit attributes from Entity. Game, Team and Entity make up a superclass because they share id, and name. Game Team and Player inherit from Entity.

Team and Players have a relationship, as does Game and Team. GameService and Game also have a relationship. To satisfy the client’s requests, the diagram indicates that one instance of GameService may have multiples of Games, Games have multiple Teams and each Team have multiple players. Finally, ProgramDriver uses SingletonTester.

****

## [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** | Easy to servers. Uses Unix systems that has not been commonly challenged by hackers but is still vulnerable. Learning curve can be steep. Software and security updates. | The most commonly known open system. Great server systems but, too many flavors that poses a consistency challenge. Learning curve is steep. | Extensive Server Tools and software, Wider knowledge base and training availability. Security is needed because of know exploits by hackers.  Software and security updates. | Miniaturized application that can fit to smaller screens would have to be designed when it comes to servers. Also, the challenge is to make touch screens server friendly in at a smaller scale. Software and security updates., |
| **Client Side** | Apple is ready to use out-of-the-box. It presents a less maintenance for users and is known for its durability. It is expensive. It is a proprietary system that does not allow a lot of personalization.  Software and security updates. | Cheaper to purchase and maintain, with some knowledge. Uses less energy to maintain efficiency. Learning curve for end users can be tricky. Some come without a Graphical user interface (GUI), which makes it scary for new commers. | Most common personal computing on the market. Most used all over the world and training is highly available for this platform. Portability is good. Software and security updates. | Because Cell phones have a lot of manufacturers, the challenge would be to engineer software work efficiently to the client’s satisfaction. The smaller screens can be challenging for anyone for developers. Software and security updates. |
| **Development Tools** | * Icon Tools for developers * X code * Deploye Studio. * Eclipse * Visual Studio | - Emacs  - Eclips  - Visual Studio  - Git  - Netbeans | - Notepad ++  - Microsoft Visual Basics  - Visual Studio Code  - Atom | - Google Chrome Dev Tools  - Mobile phone emulator  - Browser Stack. |

## 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 am recommending Windows Systems because it is the most used Operating System on the market. Also, it is known for its server efficiency, programming language platforms availability and has an advantage when it comes to Security software and Firewall and software/security updates. Also, emulation tools make using other software operating systems available for use to include Linux as a dual boot on a separate partition.
2. **Operating Systems Architectures**: Windows Server, used and known for its server applicability. Has two modes, User mode and Kernel mode. This system has been around for more than twenty years and has made a comeback on the market, because of its reliability to manage large and small networks.
3. **Storage Management**: Storage management Provider, “enables management of storage systems” on Windows Server platforms. File sharing runs more efficiently than most platforms.
4. **Memory Management**: Windows Caches file data read from disks and written to disks, read from the system memory known as system file Cache and not from the actual physical disk. All of this is coordinated by the Cache manager which operates continuously, as long as the system stays on.
5. **Distributed Systems and Networks**: “Connection of coupled processors interconnected by a communication network. May include small microprocessors, workstations, minicomputers and larger general-purpose computer systems. Good for resource sharing, speed up, reliability and communications.”
6. **Security**: Windows Server has a hierarchical user platform that separates users between administrators and regular users. Certain elevated resources and hidden away non administrators and each users profile remains private from other users.

References

Ronkainen J., Abrahamsson P. (2003) Software Development under Stringent Hardware Constraints: Do Agile Methods Have a Chance?. In: Marchesi M., Succi G. (eds) Extreme Programming and Agile Processes in Software Engineering. XP 2003. Lecture Notes in Computer Science, vol 2675. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44870-5\_10

Phstee. (n.d.). Performance Tuning for Cache and Memory Manager Subsystems. Retrieved November 16, 2020, from <https://docs.microsoft.com/en-us/windows-server/administration/performance-tuning/subsystem/cache-memory-management/>

Operating System Concepts, 8th Edition. (n.d.). Retrieved November 16, 2020, from https://learning.oreilly.com/library/view/operating-system-concepts/9780470128725/silb\_9780470128725\_oeb\_c16\_r1.html