

<The Gaming Room Web-based Game >

# **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 | <06/27/21> | <Christopher Ivey> | <This is a Software Design Document for our client *The Gaming Room* > |

## [Executive Summary](#_sbfa50wo7nsh)

<Write a summary to introduce the software design problem and present a solution. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.>

Our client *The Gaming Room* has reached out to us at Creative Technology Solutions (CTS) for streamline help in the development of a web-based game called Draw It or Lose It, that is supposed to be designed to be able to be played on multiple platforms. *The Gaming Room* isexpecting the game to be developed based on their current game that is available only on an Android app. *The Gaming Room* is expecting us to convert an already working Android App into a game application that is functional on the Mac, Linux, and Windows OS platforms. The game’s features need to include the ability to have one or more teams involved, and need to be able to have multiple players assigned to it. *The Gaming Room* wants the web-based game to be able to contain unique game and team names, so that users will be able to check whether a name is in use when choosing a personal and/or team name. Our client is also expecting the web-based game to be able to have only one instance of the game existing in memory at any given time.

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

While developing the web-based distributed environment, there are various implications that will arise. Currently the game app is not full featured meaning it has some limits to its portability and functionality. We are being asked to extend the app into a full featured application. To accomplish this, we will need to address the various interface designs that the game will be played on. Computer game systems platforms like Mac, Linux, and Windows, would be a unique interface to develop the game on, because of the various computer systems that understand different coding languages. We will need to make use of the Python, Java, and C++ programming languages in order to develop the gaming application so that it is fully functional on all platforms that you are requesting. In addition to that, when involving the application being suitable for Apple store as well, will require understanding of the architecture of Apple’s software, just as we did for the Mac game application. The resolution and screen size constraints for the game application will vary based upon the user’s interface. In addition to that, some interfaces specifically smartphones come in various memory sizes, which may operate primarily with only one programming language and/or may not even be able to support the application if the software is not advanced enough, whereas another platform maybe able to operate utilizing multiple programming languages at once, resulting in higher resolution and higher memory capabilities. Therefore we will need to have multiple people working on this project as the design process for the game will be very detailed and large in workload, because of the multiple languages and requirements to the game. Also, in order to guarantee the complete functionality of the game on all of the various platforms, we will need to run the game through many test. We will constrain our team to using flowcharts to create organized diagrams of the processes through developing the application to provide industry standard code to our client.

## [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)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

The UML class diagram below uses the OOP principle of Inheritance with the *Entity* class. The *Entity()* class is the private “ - ” base class that holds common attributes and behaviors, as it relates to all the other classes. The Parent*Entity()* class extends “ + ” to the *GameService, Game, Team, and Player* classes, by using the public attributes and methods for String names to analyze between id’s and name instances, in order to prevent duplicate name variances. The base class uses the public methods getId(), getName(), and toString(), in order to access each name in the other classes.

The GameService class extends to the Game class, holding the 0 to many symbol, stating there will be many games being played within the GameService. The GameService class also holds private “ - “ attributes and public “ + ” methods . The GameService class makes use of a public String when creating the nextGameId(),nextTeamId(), nextPlayerId() attributes within the class. The GameService class contains the addGame() , *get*Game(), getGameCount(), getNextPlayerId(), getNextTeamId() methods.

The Game class extends to the Team class and holds the private team attribute, connecting it to the team class. The methods box holds the Game: id and string name for the game. This will allow the game to check for duplicate instances of the game or name variations. This class also holds the + addTeam() method as well as the public toString() method which allows the game to add a team to the game by name.

The Team class holds a private attribute player. Which holds a list of names. It extends to the player class with the 0 to \* many symbolism. It also contains public methods addPlayer() which is a String name, that will be added to the toString() method.

The *Player* class is a public class that is accessed by the *Entity, Game, and Team* classes. The *Player* class holds public methods Player () and toString().

****

## [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** | * Con: Limited availability of Applications * Con: Very expensive Systems * Pro: Flexible terminal commands to configure the server, access, or make changes. | * Pro: Has a command line for standard operations. * Cons: Complex operations; Porting doesn’t exist for all programs * Pro: Large arsenal of applications included * Pro: Regular users have no access to basic system settings; known security gaps or breaches are handled quickly * Pros: It is considered the best option for hosting web servers because of its efficiency and reliability | * Con: Very prone to user errors; integrated interface seen as a potential point of attack * Pro: More software applications available compared to other OS * Pro: Has a Graphic User interface for its standard operation * Pro: System and System Applications are very well documented; Less so the case with API components and file formats | * Con: Android OS require a stable and continuous ethernet connectivity * Pro: Whether it be Android OS or Mac OS the server processes * Pro: Apple iPhone have an easy to use interface as well as iCloud capabilities for referencing personal information between the sever and the client on all Mac and iPhone mobile devices. |
| **Client Side** | * Con: Web browser is hidden * Pro: Many applications are available to the client at no cost * Con: Expensive machines , not cost friendly | * Con: Learning the system can be frustrating to the user , time has to be set aside to use the Linux OS * Pro: Complete source code of the system, API, libraries, and applications available. * Pro: Can be obtained without paying for a license * Pro: Cost friendly * Cons: A lot of the popular games are not supported on the Linux OS | * Con: Web applications may no longer be compatible with the new mandatory updates that are enforced by Windows OS * Pro: The browser for windows has easy to install buttons that makes it easy for * Pro: The time it takes to learn windows for the client side is simple because of the Start menu that is provided | * Con: Slightly more difficult to implement than other devices. * Pro: Provides flexibility to clients or even developers to see updates at any place. * Cons: A lot of money has to be spent on costly GPRS packets in order to run the server smoothly on the client side * Cons : Android has an opens source OS that invites a lot of attacks. Android-based applications are notoriously infamous for containing malicious viruses. Some of them to mention are Ground Force, Counter-Strike, etc. |
| **Development Tools** | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mac.>   * Con: You must have an Apple developer account and the Xcode IDE on a Mac computer to build this type of software for deploying on mac .   Con: Eclipse IDE needed | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Linux.>   * Con : Eclipse IDE needed | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Windows.>  Con : Eclipse IDE needed | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mobile Devices.>   * Pro: Android Mobile apps require the object-oriented language Java for development * Pro: Kotlin is an alternative language that is supported by Google and interoperable with Java |

## 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**: <Recommend an appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments.>

I would recommend using the Windows Operating Platform to use as a server to host our Java web application. Windows has the capability of supporting all of the tools ( IDE Eclipse for Java) that we need to complete The Gaming Room’s Draw It or Lose It application while computing it to other environments.

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

I would recommend using the event driven architectural pattern, which is for highly adaptable distributed systems. Using the event-driven pattern will also allow us to distribute the systems to other computing environments as well. The architecture of windows can be described in detail as a layered design that consist of two main components user mode and kernel mode. The windows structure is a modular structure composed of several different modules which include the kernel/microkernel , executive services, the environment subsystem and the integral subsystem. The application can be run in two modes within the Windows operating platform and those are Protected mode/ kernel mode. We will use the three tiered architecture for the client server.

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

For the storage management system it will be wise to implement the distributed file system method (DFS). For storage we will need 1.6GB of storage. Using the windows operating platform will provide a minimum of 16GB of storage. Windows

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

For the Draw It or Lose It software, preparing for the worse case our memory needs to be able to support 1000 games, 4 clients per game, 1 image per round. Windows operating system has a cache memory manager which is part of the I/O architecture that is used to improve the performance of the I/O systems. Instead of reading and writing directly to a disk, frequently used files are temporarily stored in a cache memory , and read and write operations are performed to these files in the memory management. The windows operating platform also has the kernel mode , where it can execute any CPU instruction and reference any memory address. Windows implements virtual memory using demand paging with clustering. Clustering handles page faults. When the amount of free memory falls below the threshold, the virtual memory manager uses a tactic known as automatic working-set trimming to restore the value above the threshold.

1. **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).>

The windows operating system provides support for multiple operating environments, or subsystems, where application programs communicate via a message passing mechanism called the local procedure-call facility. Windows uses a port object to establish and maintain a connection between two processes.

Windows XP uses two types of message-passing techniques over a port that the client specifies when it establishes the channel. This is how the client and the server will communicate so that the Draw It or Lose It game is distributed between various platforms. Also with the use of Restful API we will be able to distribute the system to various platforms.

1. **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.>

The windows operating platform has a security feature called the security reference monitor that is responsible for enforcing encryption through validating access and auditing-generation policy which is defined by the local security subsystem. We can implement the use of passwords and two factor authentication in order to provide user information protection.