

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 2.0 | 10/14/2025 | Mason Doyle | 1. Alterations made to bracketed areas to define the questions/fields parameters 2. Expanding on defining the client side, server side, and developmental tools for mac, windows, Linux, and mobile devices 3. Provided recommendations regarding technological resources and practices to increase the efficiency of the game, its reliability, and its security and user safety |

**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

*Our client, Chat Away, currently operates a successful online platform, but struggles with outreach to customers who use their phones applications more than actual websites. Given this need, Chat Away as expressed interest in the creation of a mobile application, usable on both iPhone and android, which can be used to bridge this gap. Identified as an area of opportunity, they would like to have an application developed which both meets their needs and stays within the confines of their projected budget for this project.*

## Requirements

*The requirements of the customer, Chat Away, are as follows:*

* *Develop an application which operates and functions as similar to current website as possible*
* *Application is available for download and use to both iphone and android users*
* *Project is as cost efficient as possible and stays within suggested budgetary restrictions*

## Design Constraints

* *Theme, function, and database/resource availability and access to mimic website*
* *Programs and drivers must be built to be operable on both IOS and Android Systems*
* *Focus on crucial and required functions – limit code to as much iterations, loops, and program functions as much as possible to limit time requirements for lengthy code*

## [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)

To begin, a super class named ‘Entity’ was created to offer parentage to its child classes; Respectively named ‘Game’ , ‘Team’, and ‘Player’. This now establishes a relationship between classes where the child classes will inherit the attributes of its parent class, while having their own respective methods, formulas, and attributes. The class name ‘Game Service’ uses its own and inherited attributed to verify that user requirements are met, unique names for team, game and platyer as well as only providing a single instance of a game at a given time. The class named ‘ Program driver’ holds the Main statement for the program and used the class named ‘Singleton Tester’ to test for singleton creations. The class named ‘Game’ hold a ‘team’ list while the class named ‘Team’ holds a ‘Player’ list. The class named ‘Player’ contains no lists as itsa main operation is the verification of unique Ids being issued to players which in turn can be assigned to a team.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [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** | One advantage is the ability to run applications designed for mac, windows, and Linux all simultaneously. Another is the consistency of the programs and product, which allows for easier and more consistent implementations and updates of programs. However, a disadvantage is the price, which generally runs much more expensive than other options. | Unlike mac, the main advantage here is the affordability, with multiple free, or nearly free, options with a wide range of distribution available, it offers the potential for a lot of power with a low cost. A disadvantage though however is pre built machine options and issues stemming for file formatting compatibility. | An advantage for windows is the included options which make corporate usage highly desirable with active servers and authentication without additional charges. A disadvantage is centered around its protocols for security, and how this can expose you to malware, spyware, and other malicious viruses. | An advantage is a wide range of web application servers available for android deployment and works great for smaller in size or scope programs and applications. A disadvantage is that with being mobile, cloud based servers are required for usage which creates a potential for viruses and malware attacks. |
| **Client Side** | An advantage that comes with using mac is the usability found within, once familiarized with the console and OS, the functionality and general usage are much more user friendly opposed to other systems. A disadvantage is the limited scope, which limits the accessibility to Mac devices, limiting the range and impact to users who may not use these devices | As an open-source OS that is highly affordable, Linux offers a combination of control and manipulation of your environments for a reasonable price. This, however, comes at the cost of secutrity and resources for security, since being open sourced, the exposure and risk of data leakage is a problem which needs tro be addressed and maintained. | An advantage of Windows is the affordability and customization. There are multiple tiers of accessories and options with varying costs, which allows for a system which is more a la carte, and you pay for what yuopu need. A con can be the cost as well as the accessibility and options can begine to add up. | An advantage to mobile devices is the accessibility, where an application or software can be accessed anywhere there is a wireless connection. A disadvantage is that the capacity and features available are often limited compared to PC operated versions, with the highest variety of OS amongst other systems. |
| **Development Tools** | Swift, a programming language used for mac, mainly developed on Xcode. Xcode had been designed specifically for Mac to offer a more straight forward and easier way to develop programs. | Though there is a large variety of tools available for development, Docker is one which offers a consistent development environment which streamlines the process. | VS or visual studio is one of the most commonly used IDE’s available for windows. It offers a free and paid version with tools available to best fit your needs | Mobile applications can user and host a variety of syntax with variable uses, though java remains the most common. Similar to windows, visual studios seem to offer one of the better development experiences. |

## 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**: The suggested Operating platform for the development of this game is Windows, which will best integrate with the currently used android application. The resources available, a large userbase, and many tools all make it a reliable and beneficial choice.
2. **Operating Systems Architectures**: Utilizing the UWP offered by Microsoft can allow for the singular application to be available and run on all devices. This again, adds weight to the power offered by a icrosoft based operating platform.
3. **Storage Management**: For best storage management, a server-based storage will be the best option as it offers automatic back up with a faster and optimized performance capacity. This, combined with a limited storage requirement placed on the players device which offers the performance capacity, offers smooth, reliable, and efficient experience for the game players.
4. **Memory Management**: There are both physical and virtual memory spaces available with other cloud-based services to supplement storage of memory. Determining what informations needs readily available and what data can remain dormant until called, will help identify the proper storage locations to optimize readability and accessibility in the program itself, inturn increasing the user experience.
5. **Distributed Systems and Networks**: Cross platform development encironemtns will help reduce reliance of multiple forms of expertise and offer a more efficient development environment. For the sake of connectivity ad outages, identifying and using a server that has or exceeds the requirements of all potential users across environments.
6. **Security**: With an evolving world comes evolving threats, investing in a verifiable and trustworthy security service, such as Aura, can help mitigate these threats. Using an additional security service outside of what is additionally provided offers a much needed extra layer of protection. Offering supplemented security measures not only protects possible corruption and infection of the game, but helps protect those who may interact with the game from potentially malicious software.