

<Name-of-Software-Application>

# **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 |
| --- | --- | --- | --- |
| 1.0 | 8/25/24 | David Moon | <Brief description of changes in this revision> |

**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.

A screenshot of a computer program

Description automatically generated

[Executive Summary](#_sbfa50wo7nsh)

The Gaming Room is our new client that is working with us to devolop a game called Draw It or Lose It serving multiple platforms based on their based on the version they have currently on Android only.

We have been given a UML diagram and have been requested to start working on the project as well as prepare a software design document.

## Requirements

One or more teams each game.

Each team can have multiple players.

Game and team tames unique.

Only one instance of a game can exist in memory at once.

## [Design Constraints](#_2et92p0)

Performance: Needs to have minimal lag and responsive so the game is fair and enjoyable.

Scallability: Needs to be able to support multiple teams with a varying number of players.

Security: The players data should be protected at all times.

Compatibility: The game needs to be able to be run on standard browsers.

## [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)

The base class Entity holds the variables which are inherited by the Game, Team and Player classes. The GameServices class has a none to many relationship with the Game class. The Game class has a none to many relationship with the Team class. The Team class has a none to many relationship with the Player class.

## [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** | Offers Unix-based enviroment that is favorable to hosting web applications though is less often chosen over the alternatives due to the expense and limited hardware options. Great security and stability. | Popular due to great stability, security and large amount of suitable hardware for server use. More knowledge may be needed to set up. | Great intergration with Microsoft toools and technology with a large range of server options. Lots of support for .net applications. The cost is  expensive | Limited hardware use, security risks, limited scalibility as well as bandwidth and network limitations make moble device inferior to the other options despite the lower cost and ease of devolopment. |
| **Client Side** | Needs to use apple dev tools and knowledge of those specific tools may be needed. | Vary’s depending on enviroment with costs being generally lower. | Often uses Visual Studios with knowledge in .Net and C# often being needed. | Platform specific with expertise needed for the use case. Swift for IOS, Kotlin for android and crosssplatform tools if that is needed. |
| **Development Tools** | Xcode is primary IDE supporting Swift, Objective c and C++. | Supports many languages as well as most popular IDE’s. | Visual Studio is the most commonly used IDE and supports many languages. | Platform specific with expertise needed for the use case. Swift for IOS, Kotlin for android and crosssplatform tools if that is needed. |

## 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**: Linux due to it’s stability, wide range of hardware, performance and security.
2. **Operating Systems Architectures:** Multi tier seperating presentation data as well as the application.
3. **Storage Management**: SSD’s due to their speed being needed so that the images appear smooth and quickly.
4. **Memory Management**: Linux uses paging to manage memory so that unused data is swapped to disk keeping everything optimized.
5. **Distributed Systems and Networks**: Restful API due to it’s ease of implementation with Linux and it’s wide usage.
6. **Security**: Linux is already does a great job provide security with firewalls and access control. SSL/TLS encryption would be the go to for encrypting the communications making sure everything is secure.