

Tanner Humphreys

# **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 | 11/11/20 | Tanner Humphreys | Update Of 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](#_sbfa50wo7nsh)

The Gaming Room is in need of a web based game that is able to run across multiple platforms. A title called “Draw It or Lose It” will be the name of the application and it is currently available on the Android Playstore. The application will be playable with teams and will be played on a round-to-round basis. Each round will last one minute. During each round, the application will render images from a library and team members will need to guess the image before time runs out. Failure to give a correct answer will result in the opposing team to being able to guess the image.

## [Design Constraints](#_2et92p0)

Use of these requirements will ensure success of the development of the application. The Gaming Room is wanting to have this application available on all devices including Android, Apple, Windows, and Mac. The application is already on Android devices and the code will need to be rewritten or redeveloped to function properly on all devices.

Design Constraints include:

* The use of one or more programming teams
* Well rounded team members per team
* Unique game and team names will be used in the application
* Game instance must be unique
* Only one game may exist at a time
* Application will run on multiple platforms

## [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 Entity class will create the relationships between Game, Team, and Player classes. Each of these classes will inherit information from Entity class. The included UML showcases common references between classes. Ex: name, id. The Entity class will be considered a super class that will handle the relationships between each class. Each team will have a player, while each game will have a team. Below it can be seen that GameService has a reference of Games, and Games has a reference of Team, and Team has reference of Player.

****

## [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** | Adaptable commands for use on Terminal to configure server, client access, and changes | A cost friendly version of Mac. Flexible commands to be used on Terminal | Easily the most availability for software when compared to other operating systems | Servers should be stationary. A server being ran from a mobile device is susceptible to crashes and overloads |
| **Client Side** | User friendly interface with minimal learning curve. Higher costs of clients that can be unaffordable from some users. | Smallest user base with very customizable interface. Large learning curve while being budget friendly. | Largest consumer base with well-known UI. Budget friendly with minimal learning curve. | Versatile, customizable, and offers flexibility. Development can be tough due to the number of devices and difference in hardware. |
| **Development Tools** | The use of Swift, or X-Code is commonplace for Mac. While other languages can be used, these are specific for Mac. Other tools such as notepad++ can be used across platforms | Support for almost every language and IDE on the market. Visual studio, eclipse and notepad++. Python and java script are popular languages. | Similar to Linux in support. Many IDEs are readily available with almost every language being supported. | Mobile app development has taken off over recently and many languages support mobile development. |

## 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**: To start, development on Windows based machines is recommended as it is the most used and most supported. The are numerous IDEs to use with resources available for additional help.
2. **Operating Systems Architectures**: Because Windows is the most used system for desktop and laptop computers, the services available will help with a unique and user-friendly interface. The development of this application will need access to a graphical interface, multimedia, and web services.
3. **Storage Management**: Most machines are equipped with proper storage. Users are given the ability to manage storage as they see fit. An installation wizard should provide a user to select the place of installation. After installation, users are able to move applications and files to places of their choosing.
4. **Memory Management**: The development of this application will need to utilize a database or library of images. The memory management should allow easy storage of pictures, and it is recommended that these images stay with the application. They should not be store in random locations on the machine.
5. **Distributed Systems and Networks**: Differing operating systems require different development strategies and can require the use of different IDEs and languages. 3rd party development tools can be used to ensure the application can be ran on different operating systems and platforms. The Game Room is recommended to ensure servers are able to handle the traffic that comes with server-based applications. Servers should be strong and secure enough to allow such traffic.
6. **Security**: The Windows platform has built-in security for virus and malware. The use of additional services and programs can be used at the discretion of the user. Proper security measures will ensure a safe and enjoyable environment. The use of automatic updates is recommended as new threats are found often.