

<Name-of-Software-Application>

# **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 | 09/19/21 | Jason Yaeger |  |

**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 game rooming would like to develop a web-based game that can run on multiple platforms. “Draw It or Lose It” is the title of it, and it is currently exclusively for android. The purpose of the game is multiple teams consisting of several people playing four one-minute rounds. When a picture is pulled from a library of images one team guesses until the time runs out. If left unanswered each opposing team member gets to guess until 15 seconds runs out.

## [Design Constraints](#_2et92p0)

* Requires one or more teams
* Each team has multiple players
* Game and Team names must be unique so that users can check whether the name is in use or open
* Only one instance of the game can exist at any time
* Must run on multiple platforms/architectures

## [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 creates a relationship between the Game, Team, and Player classes. The Game, Team, and Player classes all inherit or get information from Entity. This can be seen in UML with inheritance. Each class will share common references like “name” and “id”. Creating Entity a superclass. Looking at their relationship, we see Team and Player is a “has a” type. While Game has a Team and GameService has Games. When we use UML, we call it aggregation (HAS-A). When a user “has a” I mean it's an instance of one class and has a reference to an instance to another class. When we look at this diagram, we see GameService has a reference of Games, Games a reference of Team, and Team a reference of Player.

**"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** | |  | | --- | | Flexible terminal commands to configure the server, access, or make changes.  **Characteristics** popular for web hosting  **Advantages** It is upgradeable, it has various options for different web hosting requirements  **Disadvantages** It is less preferred for web hosting services | | The same goes for mac plus more cost-friendly  **Characteristics** Secured, open source  **Advantages** Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  **Disadvantages** It is more difficult to find applications to support the web hosting required needs. | More software available compared to other OS.  **Characteristics** It is dominant to the other platforms. proprietary  **Advantages** High resource requirements, less loading time, high comfortability, lts of support  **Disadvantages** virus susceptibility, poor tech support | It's better if the server is immobile and can be tracked in a single place. Specifications are better in other devices.  **Characteristics** More popular, high portability.  **Advantages** wide reach, good compatibility, cost-effective  **Disadvantages** selective to various smart phone devices poor security |
| **Client Side** | |  | | --- | | Average expertise and time required. Cost similar to windows. | | Maximum expertise and time required. Minimum cost. | Minimum expertise and time required. Cost similar to mac. | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | When running languages on macs we can use swift which is a popular option. While mixing in nice tools like notepad++. Though Macs can run all languages. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Linux can work with visual studio, eclipse, or notepad++. There are also many more languages and tools such as HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Easier to use than Linux but can run the same as it. So visual studio, eclipse to name a few of the many languages. And with multiple tools notepad++ is a simple to use tool. Languages consist of but not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. Like Linux these can be Java, Python, PHP, and Ruby. | You can create many apps using android and swift. Both languages and software can be run on all three machines. Languages consist of but are not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. |

## 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 think there are a lot of pros to starting on windows devices as it has more software available along with minimum expertise and cost to get projects started. There are also plenty of IDE’s to work with. There are many developers that are knowledgeable and experienced in Windows which always helps as well. We can run windows on the cloud be it with AWS or Azure or google.
2. **Operating Systems Architectures**: Windows provides services used by all Windows-based applications that enable applications to show a Graphical User Interface while accessing system resources and much more. These applications also refer to Graphics and Multimedia, messaging, and web services. These services can be used using a user account or a server specifically. The standard windows architecture uses a single processor system that is either 32 or 64 bit. The 32 bit versions are limited to 4gb of RAM while some of the 64 bit versions such as pro for workstation can support up to 6tb of RAM. If we implement a cloud approach, scaling the services for Draw it or Lose It will be simple and potentially quite cost effective.
3. **Storage Management**: Windows 10 has many great features, one of which is storage sense. This allows you to scrutinize and manage files on your hard drive. The deletion of temporary files is important and can be done with storage sense in order to prevent no longer needed files/data from piling up. Other features include being able to choose to save locations for apps making them easier to find. And just like other drives, you can also use the cloud to save/store data. The built-in storage system allows for easy file creation and placement for large projects, so they won’t get lost or carelessly deleted. The tree-structured directory is effective and every file will have a unique path name.
4. **Memory Management**: The process of creating this game will require creating a database or library with lots of pictures. The memory allocation allows for easy storage of pictures outside of the default picture folder, meaning you can keep your whole project together in a more secure area on the computer. This also includes when you’re using the IDE.
5. **Distributed Systems and Networks**: The software or game is hosted in a distributed environment. This means it is network intensive and the actual game runs on the application servers and the inputs are taken through client applications, processed at servers and output is rendered on the client screen. To prevent problems like outages or connectivity issues, the company will need to make sure their servers are strong enough to support large player bases along with backup power in case of outages.
6. **Security**: Windows features windows defender and comes with built-in security protection software. Security software can scan for malware, viruses, and other threats. This all happens in real-time or can be scheduled depending on the software, and because threats change the system often has required updates automatically to better protect the system. I think as always using two-factor authentication would be a great help for security and access control. Most modern apps utilize this and I think The Gaming Room would be wise to do the same.