

Draw It or Lose It

# **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  1.1  1.2 | 11/14/21  11/28/21  12/12/21 | Daniel Taylor  Daniel Taylor  Daniel Taylor | Adding Executive Summary, Design Constraints and Domain Model  Adding Evaluation Section  Adding 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 desire of The Gaming Room is to develop a web-based game that will run on multiple platforms. The name of the game is “Draw It or Lose It” and is only available on android currently. The game consists of multiple teams containing several people going for four rounds with one minute for each round. A picture is pulled from a library of images and one team guesses until time runs out. If not answered correctly each opposing team member gets to answer with a fifteen second time limit.

## [Design Constraints](#_2et92p0)

* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

These requirements must be followed while writing the code and developing the software. The Gaming Room would like this application to run on all devices, and while we already have it on android, we need to work it into Apple, Windows, and Linux devices. To accomplish this we could re-write the code or find a way to use the existing code to run on other devices through inheriting other languages.

## [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 Player, Game, and Team class. They all inherit information from Entity class. This is shown in UML with inheritance. Each class shares common references like “ID” and “name” which makes Entity a superclass. When we look at the relationship, Team and Player is a “has a” type. While Game has a Team and GameService has Games. When using UML, we call it aggregation, when a user “has a” it means 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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Has flexible terminal commands to configure the server, access, or make changes  It is popular in web hosting  It is upgradeable, and it has various options for different web hosting requirements  It is less preferred for web hosting services | Like Mac plus more cost-friendly  Secured, most preferred.  Security flaws are caught before they become an issue  It is very difficult to find applications to support the web hosting required needs | More software available than other OS  It is the dominant platform  High resource requirements, less loading time, high comfortability level  Susceptible to viruses and somewhat poor technical support | It's better if the server can be tracked in a single place  Extremely popular, and easily portable  Has a broad reach, better compatibility, and is cost-effective  It is very selective to various smart mobile devices  Poor security |
| **Client Side** | Moderate expertise and time are required  Cost similar to Windows  Consider what is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices | Maximum expertise and time are required  Minimum cost  Consider what is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices | Minimum expertise and time are required  Cost similar to cost of Mac  Consider what is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices | Provides flexibility to developers and clients to see updates at any place  More difficult to implement than other devices |
| **Development Tools** | When running languages on macs swift is the most popular option while mixing in nice tools like notepad++  Macs can run all languages  HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages which can be Java, Python, PHP, and Ruby | Linux can work with visual studio, eclipse, and with notepad++ along with many more languages and tools  HTML/CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby | Visual studio, eclipse to name a few of the many IDE’s and notepad++ is a simple tool  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 | You can create many apps using android and swift  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 |

## 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 recommend that The Gaming Room starts on windows devices as there is more software available as well as minimum cost and expertise to get projects going. There are also many options for IDE’s to work with on Windows OS.
2. **Operating Systems Architectures**: Windows provides services used by all Windows-based applications that enable applications to show a Graphical User Interface (GUI) while accessing system resources and much more. These applications also refer to multimedia and graphics, messaging, and web services. These services can be used via a user account or a server specifically.
3. **Storage Management**: Ability to choose save locations for apps making them easier to find. Much like other drives, the cloud can be used to save data. The built-in storage system allows for easy file creation and placement for large projects, so they won’t get lost.
4. **Memory Management**: This game will require us to create a database or library with a lot of pictures. The memory allocation allows for easy storage of pictures outside of the default picture folder. This allows us to keep the whole project together in a more secure area on the computer.
5. **Distributed Systems and Networks**: Because of all the different operating systems I investigated ways to publish the game to run on all devices. I found Develop 4 which enables cross-platform game creation. It is an IDE that can be run on any device, and when the game is created you can export the game file into the web, iOS, Android, and many more options that will allow cross-play. To prevent connectivity issues, the company will need to make sure their servers are strong enough to support large player volumes along with backup power in case of power outages.
6. **Security**: Windows comes equipped with built-in security software. However, it would be recommended to use another source to secure user data and information. Again, windows is pre-equipped with protection which will scan for malware, viruses, and security threats all in real-time, and because threats change the system updates automatically to keep the system and user information safe.