

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 | 05/27/22 | Jaden Schmidt |  |

**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 wants to develop a web-based game that can run on multiple platforms. The game will be called “Draw It or Lose It” and is currently only available on android. The purpose of this game is multiple teams consisting of several people going four rounds at a minute each. When a picture is pulled from a library of images one team guesses till time runs out. If not answered each opposing team member gets to answer till 15 seconds runs out.

## [Design Constraints](#_2et92p0)

* Needs one or more teams involved
* Each team has multiple people
* Game and Team names must be unique to allow users to check whether the name is in use or free
* Only one instance of the game can exist at any time.
* Must run on multiple platforms

These are the requirements needed to follow while writing the code and software. While this is only the game aspect, we still need to look at application development. The Gaming Room would like this to run on all devices. This means we already have it on android but need to work it into another mobile device. Along with machines like Windows, Linux, and Apple. To do this we will need to find a way to either re-write the code in swift for (Apple devices) or come up with a way to use existing code to be run on other devices by inheriting other languages. Like when we use multiple computer languages together to make stronger 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)

Entity creates a relationship between Game, Team, and Player class. This means they all inherit or get information from Entity. With UML we can show this with inheritance. So, each class will share common references like “name” and “id”. Making Entity a superclass. When we look 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

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | A deployment method for macOS is Apache which is compatible with Linux and Windows as well. There’s also Nginx which is free as well  Apache is an open-source tool, along with side programming languages and scaling tools, thus it comes free of cost. | The most common web servers for Linux is Ubuntu and Apache. There’s also Nginx which is free as well  These are both open-source software, along with scaling tools and have no charge. | Microsoft has there own built-in server, IIS or Internet Information Services which is free but is bundled and only compatible with Windows. There’s also Nginx and Apache which is free as well | There are plenty of web server apps for Apple and iOs products that will host a web server, usually free of charge. |
| **Client Side** | Most web apps can be ran in most browsers, which in MacOs’s case is safari generally. Using the web server software described above, other than IIS, is cross-platform and open source. Which makes them free of charge. The time and expertise to program the frontend and backend is dependent on team experience and expertise using various frameworks | Most web apps can be ran in most browsers, which in linux case is mozilla firefox generally. Using the web server software described above, other than IIS, is cross-platform and open source. Which makes them free of charge. The time and expertise to program the frontend and backend is dependent on team experience and expertise using various frameworks | Most web apps can be ran in most browsers, which in Windows case is Microsoft edge or chrome generally. Using the web server software described above, other than IIS, is cross-platform and open source. Which makes them free of charge. The time and expertise to program the frontend and backend is dependent on team experience and expertise using various frameworks | Most web apps can be ran in most browsers, which in mobile devices: IOs uses safari like Mac, and androids usually use Chrome like windows. Using the web server software described above, other than IIS, is cross-platform and open source. Which makes them free of charge. The time and expertise to program the frontend and backend is dependent on team experience and expertise using various frameworks  With mobile devices creating a separate mobile app for the two dominant operating systems will increase traffic since most mobile users use their phone’s apps instead of running an app through a browser app lie safari or chrome. |
| **Development Tools** | Xcode is a common IDE for developing apps on mac. Swift and objective C is the common language. These tools are free but require expertise from team. | Linux has a variety of open source ides and libraries and frameworks to help develop web based apps. There is Android Studio, Anjuta and Aptana Studios which is based of eclipse.  These support a variety of programming languages from Ruby python javascript etc. | There is open source Ides and tools and there is paid for software to develop using windows. This includes Eclipse which is a free IDE and bootstrap which is a web based framework for developing with CSS HTML and Javascript. These tools can support a wide variety of programming languages. |  |

## Recommendations

1. **Operating Platform**: Since they already have a mobile app with android the team will be more familiar with Microsoft and windows would be a great place to start building the web-based application of Draw It or Lose It. There is a variety of IDE’s and softwares such as Eclipse, visual studio IDE and Bootstrap framework, React framework, and cross-platform .Net framework to help build and develop apps and most are free of charge.
2. **Operating Systems Architectures**: Windows operating system architectures is a layered design that consists of two main components: user mode and kernel mode. Kernal mode has direct access to the systems hardware or reference memory while user mode doesn’t. This is good built-in protection.
3. **Storage Management**: Cloud storage is up and coming since you just pay for the executions your application is using, compared to hard drives and power banks and fans, and a 24/7 server team to be ready for maintenance, depending on the applications success. Dependent on how many teams and pictures and whatever is needed to be in storage cloud services might be the way to go
4. **Memory Management**:  Having all the photos and game code would be stored in the cloud or physical hdd or sdd drives. The memory is where all the code the user needs for the game that running is stored.
5. **Distributed Systems and Networks**: I recommend using Apache since it has cross-platform for all browser compatibility wanted from the client
6. **Security**: Windows comes with built-in security protection software. Though to secure user data and information it would be recommended to use another source. Though if we are talking about what is on the machine windows comes pre-equipped with protection. This system scan for malware (malicious software), viruses, and security threats. This all happens in real-time, and because threats change the system updates automatically to keep the system and user information safe