

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.2

## 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 | 07/16/22 | Eric B. Hunter | Design Inspection and recommended requirements |
| 1.1 | 07/31/22 | Eric B. Hunter | Evaluate |
| 1.2 | 08/12/22 | Eric B. Hunter | Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wants to develop a web-based game that serves multiple platforms based on their current game, Draw It or Lose It, which is currently available in an Android app only. The application will render images from a large library of stock drawings as clues. A game consists of four rounds of play lasting one minute each. Drawings are rendered at a steady rate and are fully complete at the 30-second mark. If the team does not guess the puzzle before time expires, the remaining teams have an opportunity to offer one guess each to solve the puzzle with a 15-second time limit.

## [Design Constraints](#_2et92p0)

* Must run on multiple platforms
* Only one instance of the game can exist at a time
* Game and Team names must be unique and allow users to check if names are available
* Needs at minimum one team

## [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”.

**"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** | Easy to configure, most secure servers (used by the US military because of their security levels). Hardware is top of the line. Base level hardware starts at around $1000, but hosting companies are reasonable based on the size and quantity of your website. | Similar to Mac because it will also runs on Apache servers. more cost efficient. You can dust off pretty much any modern hardware you have sitting around and build your Linux server. Best for Web Hosting. Difficult to find required applications at times. | More software run on windows than other OS. More comfortable to most users. Susceptible to virus and poor tech support.  Base level hardware starts around $300. Your only choice if you need to use Access or SQL Server since they are Microsoft properties. | Scalability, Affordability, Hardware-Free, Improved Speed, Disaster Recovery are all pros of Cloud server.  Cons include  Security, Data Management, Bandwidth Limits |
| **Client Side** | The end user today is familiar with apple devices and should have no struggles, especially if using a mobile device. These devices can be more costly with a entry level mobile device starting around $700 | Because it is free the cost is absolutely the best of any option. Because it is free the expertise in using it needs to be pretty high, you can’t just hand it off to someone and walk away. | Minimum expertise needed. Cost less than Apple software and devices with mobile devices starting at around $450. Because it is Windows on a mobile device it is familiar to lots of users. | More flexible fore the clients and/or developers. Slightly more difficult to deploy, especially with Apple. |
| **Development Tools** | Flexihub, Homebrew, Xcode, iTerm2, Sublime Text, NetBeans, Pretty Much any programming language can be used so long as the compiler selected supports it. | Gedit, Vim, Netbeans, Bluefish, Geany, Aptana, Quanta Plus. Pretty Much any programming language can be used so long as the compiler selected supports it. | Atom, GitHub, Buddy, IntellliJ, Azure, Vim, Visual Studio, Jira. Pretty Much any programming language can be used so long as the compiler selected supports it. | Back4App, Firebase, Heroku, Expo React Native, Unity, Xamarin. . Pretty Much any programming language can be used so long as the compiler selected supports it. |

## 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** Amazon Web Services (AWS)
2. **Operating Systems Architectures**: The key components to AWS Architecture are:
   1. Load Balancing-this helps to enhance the application and server’s efficiency
   2. Elastic Load Balancing-can easily shrink and increase the capacity of load balancing
   3. Amazon Cloud Front- mostly used for delivery of content
   4. Elastic Load Balancer-mainly used to deliver the required traffic to the web servers
   5. Security Management-works like the inbound network firewall
   6. Elastic Cache-web service where the memory cache can be managed in the cloud with ease
   7. Amazon RDS-helps to deliver the same access as MySql or Microsoft SQL Server database engine
3. **Storage Management**: For storage I think the AWS Simple Storage Service (S3) will provide what we need.
4. **Memory Management**: The Elastic Cache offered by AWS will allow us to keep the files stored and ready making things run faster.
5. **Distributed Systems and Networks**: Using AWS will allow us to use cross platform communication. It will not care what operating system is installed on the device so long as the correct version of the game is installed, and we have added the needed provisions in the software.
6. **Security**: AWS is vigilant about privacy and security. It offers the most secure global infrastructure, you always own your data and the ability to encrypt it, move it and manage the retention of it. Encryption is automatically applied at the physical layer to help ensure your data is safe.