

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 | 07/17/2022 | Joan Estepan | Initial design |

## [Executive Summary](#_sbfa50wo7nsh)

My client, the Gaming room, needs to develop web-based gaming app for their game “Draw it or lose it”. The game will consist of teams competing to guess the puzzle by deciphering clues, which the application will render from a large library of stock drawings. Rendering the drawing will take 30 seconds and the team will have a timer to answer. If the respective team fails to answer, the rest of the teams will have 15 seconds to answer.

## [Design Constraints](#_2et92p0)

* The application shall run on a web-based environment
* It shall have the ability have one or more teams in the game.
* The game and team names must be unique.
* Only one instance of the game shall exist in the memory at any given time.
* Teams shall have multiple players assigned to it.

## [System Architecture View](#_ilbxbyevv6b6)

-not required in this project.

## [Domain Model](#_8h2ehzxfam4o)

The class “entity” has a parent-child relationship with the Game, Team, and Player. All the children inherit information from the parent(“entity”). We can also see that gameServices has a zero to many relationship to the Game class, then the Game class has a zero to many relationship to the Team class and lastly the Team class has a zero to many relationship to the Player class.

The OOP principles utilized in this diagram are encapsulation, inheritance, and polymorphism.

**"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** | *Characteristics:*  -High level of security  -Extremely user friendly, great for inexperienced users  Weakness:  -Low customization ability  -Can be hard to find a host | *Characteristics:*  -Highest level of security  -Not very user friendly  -Open source code  -powerful and versatile  -Frequent updates  -Highly customizable  Weakness:  -Some drivers are hard to find | *Characteristics:*  -Most popular OS in the world.  -Compatible with other windows tools  -User friendly  -Highly customizable  Weakness:  -Low security  -Low stability | *Characteristics:*  -Very portable  -Not user friendly  -Can run any other hosted site in the browser  -Moderate customization  Weakness:  -Low security  -Low stability |
| **Client Side** | -High cost  -requires a medium level of expertise  -Fast development time | -Low cost  -Requires a high level of expertise  -average development time | -Moderate cost  -Requires a low level of expertise  -Fast development time | -High cost  -Requires a medium Level of expertise  -slow development time |
| **Development Tools** | -Visual Studio  -Atom  -Eclipse  -XCode  -PHP  -HTML, CSS  -JavaScript  -JAVA  -Python | -Visual Studio  -Atom  -Eclipse  -IntelliJ IDEA  -PHP  -HTML, CSS  -JavaScript  -JAVA  -Python | -Visual Studio  -Atom  -Eclipse  IntelliJ IDEA  -PHP  -HTML, CSS  -JavaScript  -JAVA  -Python | -Mobl  -jQuery mobile  -Jdrop  -DroidScript  -AWD  -PHP  -HTML, CSS  -JavaScript |

## 

## Recommendations

1. **Operating Platform**: The operating platform I recommend for this system is linux. Linux is the best option because is the most secure, reliable, low cost and versatile platform. Making it the best choice for a web based app.
2. **Operating Systems Architectures**: The architecture of the Linux OS consists of several layers which are:

* The hardware layer: which consists of all peripheral devices.
* The kernel: It is the core component of Linux, it interacts with the hardware directly, and provides low level services to upper layer components.
* The Shell: Is an interface to the kernel, hiding complex functions of kernel from the user. It accepts commands from the user and executes the kernel functions.
* Utilities: Provides most of the user functionalities of an operating system.

1. **Storage Management**: For out storage management, a cloud based storage would be the best option since cloud storage is not only more convenient but also getting more popular with time. Amazon Web Services (AWS) would be a great choice since its reliable, scalable, and inexpensive. For our database, MySQL would be recommended.
2. **Memory Management**: Linux memory management contains the implementation of demand pagingand virtual memory. Demand paging is a type of swapping done in virtual memory systems. the data only is copied from the disk to the RAM when is needed by a program, and if the data is already in the memory, it is not copied again. This memory management system would be very efficient for my client’s, software “Draw it or Lose it”.
3. **Security**: Linux is one of the most secure operating systems. Thanks to its open source code, anyone can detect any vulnerability in the code and fix it. Also, any malicious hidden code would be found by the community and fixed as well. Linux is more secure than both Windows and MacOS.