

Draw it or Lose It

# **CS 230 Project Software Design Template**

Version 3.0

## 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 | 03/22/2022 | Justin Dougherty | Explaining the changes from android only to multi platform |
| 2.0 | 04/03/2022 | Justin Dougherty | Complete Evaluation |
| 3.0 | 04/22/2022 | Justin Dougherty | Updated Recommendations |

## [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)

* Requirements would need one or more teams involved
* Each team consists of multiple people
* Game and Team names would need to be unique to allow users to check whether the name is in use or free
* Only one instance of the game can exist at a time
* Must be cross platform compatible.

These are the requirements needed to follow while writing the code and software. While this is only the game aspect, we still must consider the application development, which may be the biggest constraint. 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 multiple coding languages i.e Swift for iOS users or find a way to inherit other languages into existing code. Much like when we utilize the strengths of multiple computer languages together to make better, efficient 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 additional class will share common references (name and id). Because of this, Entity would be considered 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 this is considered 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. Also, within the diagram you see GameService has a reference of Games, Games a reference of Teams, and Team a reference of players.

**"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** | Versatile terminal commands to configure the server, access, or make changes.  Characteristics It is popular in web hosting  **Advantages**:  It is upgradeable, it has various options for different web hosting requirements  More Secure than Windows as it has less virus susceptibility  **Disadvantages**:  It is less preferred for web hosting services  Limited Hardware options with obscure productivity software unavailable. | The same serve side requirements as that of Mac.  plus more cost-friendly  Characteristics Secured, most preferred.  **Advantages:** Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  Open Source  Cost Effective as it will most likely be free  **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. Close platform  **Advantages** High resource requirements, less loading time, high comfortability  Elaborate antivirus software (at an additional cost)  Great Customer support (also considering costs)  **Disadvantages** easy virus susceptibility, Windows also has inefficient technical support when compared to Mac.  It is the most expensive option when considering security and additional costs | Strengths increase when the server is immobile and can be maintained and tracked in a single place.  Specifications are better in other devices.  Characteristics Popular as most people have mobile devices. Portability.  **Advantages** Wide Reach, universal compatibility, cost-effective  **Disadvantages** It is highly particular to various smart mobile devices and security issues are amplified. |
| **Client Side** | Intermediate level expertise and time required. Cost similar to windows.  Development Tool would most likely be: XCode to ensure iOS smart phones are compatible however, eclipse IDE and visual studio also work on Mac.  Language: Swift or Objective C for added compatibility | Maximum expertise and time required. Minimum cost. Software available is mostly open-source  Since you don't have any financial upfront costs, the only cost is training the development team: is gaining an understanding of it.  Would also require finding a Linux compatible system which is relatively common  Development Environment to be utilized: Eclipse IDE.  Programming Language: C | Desktop Application with Visual Basic has been proven difficult and would likely require an additional team with intermediate level expertise and time required. Cost similar to mac.  Development Enviornment:  VisualStudio  Language: C++ | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices  Android:  Development Environment:  Android Studio  Language : Java  iOS:  Development Environment:  Swiftic  Language Swift |
| **Development Tools** | While using languages on mac iOS swift is the more 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. Although the most common language is Objective C | Linux will work with visual studio, eclipse, along with notepad++ for a nice and easy-to-use tool. Along with many more languages and tools. 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, Perl, Vala, C, C++, CSS, Java, JavaScript, HTML, | Microsoft is easier to use than Linux but can operate similarly. So visual studio, eclipse to name a few of the many languages. And with multiple tools notepad++ is a simple to use the 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. C++ is the wo | There exists the possibility countless apps using android and swiftic. Both languages and software can be run on all three machines. 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.  Most likely scenario:  iOS: Swiftic would be thebest framework available  Android: would be coded in Java |

**Web Application vs Desktop Application and Mobile Devices (Evaluation Continued):**

The game can be hosted in a distributed environment meaning if the game remains network intensive and the game runs on the application servers, the inputs are taken through the client applications. Traditional Desktop platforms and modern mobile platforms will have to be included with the understanding that every platform has a separate set of development tools, APIS and programming languages. Desktop applications through mac or windows typically provide the richest user experience and can play to Draw it Or Lose It’s strengths when the response time is demanding, the downside being that the application would need to be downloaded on each device limiting the portability to each device it is installed on. This would make playing the game on the road or at work or a library difficult as some shared devices require admin rights to download software to their desktop. Whereas a web application would be limited to the strengths of the browser being used. This would require additional programing most likely in HTML CSS or JavaScript and likely be easy to develop be very difficult to predict especially because the app will rely on data from the serve and other users. There is no way to control the speed of delivery of that data through web browser. Mobile Applications are great because they run independently on devices allowing the best of both worlds.

## 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**:  My personal recommendation would be that the Gaming Room starts on Windows. Optimizing the software availability and the lower level of complexity required would be most cost effective. As the strengths of software availability, customer support and server security outweigh the weaknesses (cost efficiency).
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 Graphics and Multimedia, messaging, and web services. These services can be used using a user account or a server specifically.
3. **Storage Management**: Storage-Sense is a new Windows 10 feature that would be perfect for Storage Management. The convenient product allows you to dissect and manage the files on your hard drive and helps analyze the space of each file. It is convenient for locating the apps and setting save and retrieval locations. Storage Sense is also compatible for cloud migration, which has its own security issues, however, is perfect for larger and nonproprietary files making it difficult to mistakenly delete them.
4. **Memory Management**: Pictures, mostly for reference on Draw it or Lose it, are going to require a majority of space. However, creating a database or library of said pictures would allow for easy storage of the pictures. Keeping a safe and secure file storage system is essential. Windows could offer built in memory management by way of the system process that will allow Creating an internal data base will ensure all intellectual property rights are respected as well, in addition to ensuring processing speeds are improved by ensuring the search times are limited.
5. **Distributed Systems and Networks**: As each platform has different development requirements, I had done some extensive research on cross platform systems. Luckily, all roads led to Develop 4 an IDE that enables cross-platform game creation and can be run on any device. After the game is created Develop 4 makes it possible export the game onto the web, iOS, Android, and even to gaming consoles. As cross-platform expansion occurs, severs will need to be strengthened and improved to ensure that connectivity is maintained throughout each platform.
6. **Security**: Windows provides its own security protection software. When it comes to user data it is almost standard practice to use another source i.e. another agency to take liability for any data breach. Windows will scan itself for malware, viruses and other threats to security in real time so long as the software on the machine stays updated consistently.