

<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 | <09/19/2021 > | <Alex Espinoza> | <Added executive summary and design constraints > |

**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 company is hiring us to develop a web-based game that can run on multiple platforms. The game is called “Draw it or Lose it” and as of now is developed for android so it will need to be ported. The game consists of multiple teams guessing what the image is in a certain time frame. The picture fills in and becomes clearer making it easier to guess as the time goes on. Points are awarded to the team that guesses it right first.

## [Design Constraints](#_2et92p0)

-Each team is going to have multiple teammates

-There needs to be more than one team before the match can start

-Game and team names cannot be the same meaning they are going to have to be unique for each iteration. This will be so the player can check whether the name is in use or free to take.

-Only one instance of the game can exist at one time.

-Must run on many platforms.

These requirements are for writing the code and software, but we still need to remember the application development. The requirements for that are it needs to run on multiple platforms. Currently it is developed for android. However, with a web-based development we could run it on multiple platforms by either having it accessible through a webpage. Which every operating system has mobile or desktop. So that could be one route, or we could develop for each operating system. However, since they want a web-based game and it’d be easier to implement it’d be preferred to have it developed for web and then accessible for other platforms through web.

## [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 is a parent class meaning that other classes like Team, Game and Player inherit from it. This allows for code reusability. Then there is ProgramDriver and SingletonTester which are separate. ProgramDriver is a code driver meaning it initiates the code and starts the process. Then the singletonTester is a test code that is called to test the code and see if it is working properly. Another thing we notice in the UML is that there can be 0 to many instances of the GameService class which is a child class of game and ultimately entity. This is here so it can be adjustable for the number of games being played simultaneously. You can see this for the other classes like game, team, and player. This also allows for code reusability and uses the process of inheritance and instances. Game “is an” Entity, Game “has a” player and it “has a” team. Meaning that game is a type of entity, and a game needs a team and player to start. This here describes a little bit of the relationship between the different classes.

**"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)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements, and look at the situation holistically, as it all must work together.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | It’s used by the us military for web hosting because of its security  Advantages It is upgradeable, it has various options for different web hosting requirements, Is by far the most secure  Disadvantages it is costly, and not used by most web servers | Advantages cost, security updates, it is the most popular and preferred for servers.  Disadvantages it is not as secure as mac. | Widely available and usually easy to setup since most understand how to use windows.  Advantages it can run a lot of programs like WordPress and other things. As well as proprietary things like MSQQL Databases or .NET or Microsoft Access.  Disadvantages not as secure | Can be used to make a decentralized server  Advantages Everyone has a mobile phone. So, there can be a lot of “servers” it can also be used for decentralized servers.  Disadvantages Not everyone will want to host, it will drain phone battery and be inefficient unless used for decentralized server. |
| **Client Side** | Medium knowledge and time required. The cost is like windows. | Most knowledge and time required. But it is the cheapest. | Least knowledge and time required. Price comparable to macs. | Very hard to implement and not very well used or heard of. Also the processing of a phone is pretty low to a server/computer. |
| **Development Tools** | Mac can be used to develop every other programming language. However, it has one language that is unique to mac and is popular and nice to develop with. Despite being expensive Swift and XCode are two great development tools that are unique to apple. | Linux is a great operating system to develop many things with. As far as I know there is not a language or IDE that is only for Linux. However, Linux can access Eclipse, Visual Studios and develop for almost every major language like PHP, Java, HTML etc. | The most popular software for developing anything. It has multiple uses and has basically every development tool besides Apples, Swift and XCode. Microsoft is dominant in this realm and has many of its own IDEs like visual studio and has contributed a lot to languages like C++. It also has many development tools that are only on windows. | There are plenty of things and languages you can develop for mobile with. For example, java, ruby, python, Android, and swift with iOS. Mobile is a great way to get your program into a simple and easy to use format that can be accessible by many people. It does have limitations but as time goes on mobile devices are getting more and more powerful and the apps that they can run are too. |

## 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 recommendation is Windows. It is most likely on most of the development teams computers already and is very versatile and moderately easy to develop on. It also can develop for everything besides Mac and iOS.
2. **Operating Systems Architectures**: Most operating systems have good operating system architectures, but windows do have the most prominent ones and are very versatile.
3. **Storage Management**: Windows has many storage features that you can use when developing you can even partition sections of your hard drive to be encrypted in case your program is top secret and gets lost or stolen.
4. **Memory Management**: Windows has easy storage management for things like the draw it or lose it image library. You can just define a directory and create a folder that access the images. It’s very simple and straightforward.
5. **Distributed Systems and Networks**: There are a few IDEs that enable cross platform play. One of them is unreal engine. We could use this to get our game on many devices at once with minimal cost and work.
6. **Security**: In terms of security, I recommend using a very secure and encrypted file system for the player and the application. We also want to take minimal data from the player so if things are stolen or hacked nothing important to the user will be taken. In terms of cheating or server hosting making sure we get sufficient security software and take the required protocols when setting up the server should be enough. Perhaps implementing a bot catcher and anti DDOS might help.