

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/16/23 | Anthony DeGennaro | Completion of Exec. Summary, Requirements, 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)

Creative Technology Solutions is being asked to develop a web-based game based on their Android-based game Draw It or Lost It. The game should be able to support multiple players along with multiple teams with each instance of the game being able to only occur in memory once.

Best way to accomplish the task at hand is to create a singleton pattern that will create an object to prevent multiple game instances and an iterator pattern to prevent conflicting team names and team members.

## Requirements

* The game will have the ability to support one or more teams
* Each team should be able to have multiple players assigned to each
* Both the game and team names must be unique, with a way to check whether or not a team name has been chosen already.
* Only one instance of the game can exist in memory at any given time.
* The task above is accomplished by creating unique identifiers for each game instance, team or player.

## [Design Constraints](#_2et92p0)

The Gaming Room has an existing Android-based deployment of the game Draw It or Lost It. Creative Technology Solutions is being asked to extend this application to the web; the tech stack that will be used will need to be compatible with web-deployment. Java is the language of choice to accomplish this program. Java is the native language to Android and should ease development as well as deployment.

Any existing APIs that are using the Android platform will need to be reviewed or extended for usage on mobile devices.

## [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)

The UML that will be used for the creating of this program is below.

**"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.**

The application contains a main driver class that has the primary purpose of initiating the creation of the games, teams and players. The actual creation of the games are through the GameService class and this follows a singleton pattern to show that only a single GameService class can exist in memory at any time.

GameService blocks additional creation of instances by being set to a private class. The only way to create an instance through GameService is using the getInstance() method. getInstance() checks whether GameService has been started and will only start if it is not present in memory.

After the game is created, a player can be added to the game with the addPlayer() method. addTeam() uses the iterator pattern as well to prevent similarity between team objects being added to the game. This new team object will then be added to the *teams* list.

After a team is created, a player may be added to the team using the addPlayer() method. addPlayer() uses the iterator pattern the same as addTeam() to prevent similarities between names of players being added to a team. This new player object is then added to the *players* list.

Game, Team, and Player classes are all subclasses of Entity. Entity contains 2 protected attributes. 1: id and 2: name. The default constructor is also protected so that null objects are blocked by the program during creation and only overloaded constructors may be used.

This UML diagram shows multiple Object-oriented program techniques. Inheritance and polymorphism are present and are used in the extension of the Entity class and the overloading of necessary constructors. Encapsulation and Abstraction are also here and are used to add teams. A team object may not be created directly since the only way to do so is the user used the addTeam() method.

## [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 has to 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** | Mac has the most amount of security for hosting a web-based application out of every other option. However, there is a high price point to using Mac for hosting services. On the other side of the coin with the higher price point we get a command shell for easier server configs, and an easy-to-use GUI. | Much more difficult to navigate than the others. Is cheaper to use while also being a lot more flexible in terms of overall operation. Linux also contains a command shell making it easier to configure a server while also being able to provide easier accessibility. | Comes with a lot of different expansion options. Gives ability to use Windows server. Contains a very user-friendly GUI. Provides command prompt with a larger library of software when compared to Mac and Linux. | Specifications will vary based on different cases. When looking at specifications it is more favorable when looking at other devices that will be used for the application. |
| **Client Side** | Mac has its high price point with a requirement of having some knowledge of its platform. GUI will need some adjustment period for the end-user as functionality is different compared to Linux and especially Windows. | High level of knowledge and skill is required to use this system. There are distributions of Linux that have a GUI while also having a low cost to get into as Linux is free. But, with all of this there requires a knowledge gate to run and use. | The middle ground in price when compared to Linux and Mac. Costs more than Linux but lower than Mac. System is easy for the average user to use and understand. Takes a much lower knowledge requirement when compared to Linux. | Increased ability to access information. Flexibility, however, comes at the cost of requiring extra time to learn the platform with a high amount of knowledge required. Is more difficult to integrate with other devices. |
| **Development Tools** | Languages provided are HTML, CSS, and JavaScript with IDEs such as JavaScript, Python and Ruby. Other tools that are provided with these are PyCharm and Eclipse. | Linux provides more of the same as Mac, with additions being different libraries to back those languages. Also provides IDEs such as Python, JavaScript, Ruby, and PHP. Linux also carries different tools to help with development. | Windows provides the same languages as Linux and Mac as well. Also provides backing libraries for these languages with the added ability of having IDEs like Eclipse, PyCharm, Visual Studio and a command prompt. | Languages like HTML, CSS, and JavaScript could be used for development here. Libraries are provided to support frontend and languages. IDEs for these languages include HTML, PHP, C++ and Python. |

## 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**: Out of all of the different operating platforms that were considered for this project. Windows is considered to be the best platform to choose as its capability to expand to other computing environments.
   1. Integrates with Android.
   2. Windows allows developers to design, develop, and deploy apps using Microsoft suite products such as visual studio.
   3. Windows contains a bunch of emulator platforms: cmd/powershell/ubuntu to test functionality on different platforms.
   4. Largest user base and the most popular environment for game development and gamers.
2. **Operating Systems Architectures**: Windows has two separate operating system forms. One being user mode and one being kernel mode. User mode processes what an average end user will interact with. Kernel mode what lies underneath the GUI, dealing with inputs and outputs, networking, memory management, and routines. Windows has a directory structure that is used to hold data to later be processed.
3. **Storage Management**: Microsoft Azure is going to be highly recommended as they have the most competitive price point, great customer support, and constant updates/support. Other features:
   1. Containers called Dockers can be deployed in the Azure cloud environment to leverage multiple cloud storage instances.
   2. Cloud based storage allows for the free will of up-scaling or down-scaling easily depending on the number of overall users.
   3. Cloud storage is and will be one of the most widely used storage solutions. Due to its modularity, cost efficiency, and scalability.
4. **Memory Management**: Windows 11 is the newest version of the Windows operating system. Albeit it’s still sort of new and hasn’t been out as long as Windows 10. Overall operating system structure may not be fully developed and may have some overarching bugs in Windows 11. Regardless, Windows 11 as it stands does a great deal of efficiency for loading from memory and disc paging. Disc paging acts as an extension of the computer’s physical memory or RAM. Disc paging reserves part of the hard disk as extra RAM for processing other tasks. Once something is required to be immediately processed, the extra RAM from disc paging will process the memory through.
5. **Distributed Systems and Networks**: Microsoft Azure is the best choice as the cloud service provider because of its ease of access when dealing with distributed systems and networks.
   1. Azure offers maximum uptime with cloud-based email alerts, insight app logging for information tracking which is critical for future scaling upwards or downwards depending on amount of overall users using the platform.
   2. By offsetting the network load to Azure, it leaves extra time for additional focus on features and innovations to the game.
6. **Security**: Azure allows for simplicity in the logistics of security for user information and personal data. Devices will connect to the “Azure App Service” which uses Azure active directory for logging in so everyone has their own distinct roles and privileges. Additional features:
   1. IP configurations for whitelist and blacklist access to resources.
   2. Additional option for storage in a VPN while on the cloud for additional security.
   3. Options for support in case of leak of personal data and ways in which to protect personal data.
7. NOTE: Active internet connection is need for use with any cloud service for storage!!!!