

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 | 01/17/21 | Arys Pena | Changes made to GameService.java:  Implemented singleton pattern to adapt an ordinary class definition.  Implemented iterator to complete both addGame() and getGame() methods in order to insure there is only one instance of each class |
| 1.1 | 1/22/21 | Arys Pena | Added new base class Entity. Entity will be the super class to Team, Game, Player to hold common attributes (id, name) and common  Methods: getId(), getName(), toString() |
| 1.2 | 1/24/21 | Arys Pena | Refactored Game, Player, and Team classes to inherit from super class Entity.  Implemented iterators in Player and Team Class in order to ensure there is only one instance of each respective class. |
| 1.3 | 2/7/2021 | Arys Pena | Evaluated pros and cons of the development process on Windows, Linux, Mac, and mobile devices in Evaluation section |
| 1.4 | 2/16/21 | Arys Pena | Updated Evaluation section to narrow in on specific weakness and strength of each platform in regards to the application |
| 1.5 | 2/17/21 | Arys Pena | Updated Design Constraints to better account for software requirements |
| 2.0 | 2/21/21 | Arys Pena | Filled in Recommendations section based on updated pros and cons for each OS |

**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)

Development of a web-based game “Draw It or Lose It” will be created for The Gaming Room, the client, by technical consultants at Creative Technology Solutions (CTS). The game is currently an app only on Android and will be expanded beyond Android to serve multiple platforms. CTS will provide the technical capabilities, environment, and ensure that the scope of this project will facilitate development of a web-based version that meets five key software requirements. The game will support one or more teams involved, each team will be able to have multiple players associated with it, the game and team names will be unique, and only one instance of the game will exist at any given time. The development of this project will follow industry standards and best practices to ensure scalability and bug free code that compiles and is fully functional.

## [Design Constraints](#_2et92p0)

1. **Constraint 1:** **Storage**
2. **Constraint 2: Player/Game/Team Management**
3. **Constraint 3: Concurrent Requests**
4. **Constraint 4: Security**
5. **Constraint 5: Player Login**
6. **Constraint 6: Programming Language**

**Constraint 1:** **Storage** – Must account for enough storage on the server side to hold client’s collection of stock photos.

**Constraint 2: Player/Game/Team Management -** Must develop player management system in order to oversee players, games, and teams.

**Constraint 3: Concurrent Requests** Server must be able to handle many concurrent requests for new, multi-platform player base.

**Constraint 4: Player Login** Must have system to allow player log in to identify themselves to the game.

**Constraint 5: Security** Must have system to maintain security/login/identification in order to differentiate players with login authentication.

**Constraint 6: Programming Language:** Existing code for client’s android app is Java

## [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 following diagram depicts how the "Draw It or Lose It" game will be implemented in the technical environment. The diagram shows seven classes that help establish the environment requested by The Gaming Room.

The Entity base class is created to hold the common attributes(id and name) and behaviors(getters and toString) of its three subclasses: Game, Team, and Player.

Player extends Entity in order to hold information about each player in the game, and output that info via overridden toString() method. Player also has a relation to the Team class, and this is where the customers requirement that “each team will have multiple players assigned to it” can be upheld. Each Team holds a private list of Players and the public method to add another player(addPlayer).

Much like the Player-Team relation, there is a Game-Team relation, where the Game class holds a private list of Teams and the ability to add a team(addTeam). The addTeam method implements an iterator pattern in order to verify that each team name is unique and an overridden toString method to show the user if their desired name is taken. This fulfills the customers requirement “a game will have the ability to have one or more teams involved” and “team names must be unique to allow users to check whether a name is in use”.

The relational pattern continues with the GameService-Game relation, where GameService holds a private list of Games and has the ability to add games(addGame) and an iterator to check if the Game name is unique. The GameService is a singleton class that only allows for one instance of the game to exist in memory at any given time. In order to verify this, a SingletonTester class with the testSingleton() method was implemented and then utilized in the ProgramDriver class. With the tests succeeding, all the user software requirements have been met for the game application in an efficient and maintainable manner, but utilizing standard object oriented principals standards like Inheritance, overriding relations, and common design patterns such as Singleton and Iteration patterns.

****

## [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** | **Pros:**   * Simple server based deployment * Proven security * Extensive documentation for macOS software * Excellent technical support plans and products available   **Cons:**   * Limited hardware options * Requires Mac system using macOS * Updates solely provided through Apple * New software versions require payment * Highest cost | **Pros:**   * Open-source software available * Lowest hardware requirement * Most cost-effective option * Proven Stability * Community constantly updating and guarding against potential security vulnerabilities. * Open Source is not centrally managed.   **Cons:**   * Not as much software available when compared to windows | **Pros:**   * Simple server based deployment * Large range of software support * Wide range of hardware options * Quicker to get updates due to large userbase   **Cons:**   * Less secure than both Linux and Mac * Closed Sourced * Updates solely provided through Microsoft * Limited customization | **Pros:**   * Can manage calls to the server side to optimize the queries on the database * Stores persistent data * Code cannot be seen by the user   **Cons:**   * Server side would need to integrate with cloud services or physical servers |

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Client Side** | **Pros:**   * Wide range of well supported web browsers featuring many tools for devs * Easy cross-browser testing software * Medium development time and deployment   **Cons:**   * Requires Apple product with MacOS. | **Pros:**   * Wide range of well supported web browsers featuring many tools for devs * Tree file structure accommodations * Works with every web browser due to plethora of open-source software * Quick development and deployment | **Pros:**   * Wide range of well supported web browsers featuring many tools for devs * Easy cross platform testing besides MacOS * Quick development and deployment   **Cons:**   * More difficult to test for MacOS browser. | **Pros:**   * Expertise with android app development   **Cons:**   * Difficulty testing other environments and browsers * Longer development time |
| **Development Tools** | * documentation for deployment on MacOS * Can easily run Windows and Linux through local Virtual Machines * Application must be reviewed and approved by Apple * 99$/year for Apple Dev program * Mac OS X * Eclipse for Java | * Deployment at any time * Can easily run Windows and MacOS through local Virtual Machines * Shell prompt and terminal * Open source community * No license cost * Eclipse for Java | * Deployment at any time * Extensive documentation for deployment on Windows * Can run Linux through VM * No license cost * Visual Studio Code for Javascript/HTML * Eclipse for Java | * Xcode 12 for deployment to iOS * 99$/year Apple dev program for iOS * SwiftUI |

## 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 operating platforms considered, Windows is what we recommend in order to expand Draw It or Lose It to other computing environments. This platform is a strong contender due to the following:

* Integrates with current Android build. “Install Android Studio, or choose a cross-platform solution like Xamarin, React, or Cordova, and get your development environment setup on Windows.”
* Windows allows developers to design, develop, and deploy apps and solutions for Windows PCs easily using a suite of Microsoft products
* Plethora of emulators for cmd/powershell/unbuntu capabilities available on Window, allowing for testing on every platform
* Largest userbase and popular operating platform for game development

1. **Operating Systems Architectures**:

Windows separates its operating system into User mode and Kernel mode. User mode processes are user facing and affect much of what the user interacts with. Kernel mode is more under the hood and low level, dealing with inputs and outputs, memory management, networking, hardware management, and routines. Windows utilizes a directory structure to hold data. Windows also supports multiprocessing and modularity of hardware, allowing for customization of the system.

1. **Storage Management**:

For storage, we would heavily recommend going with Microsoft Azure for their competitive prices, great customer support, and continuous updates/support. A few more features offered by Azure are:

* + Docker containers can be deployed in the Azure Cloud computing environment to leverage cloud storage instances.
  + Cloud based storage allows for scaling up or down easily depending on user base. For initial launch, when user influx is expected to be high, the storage could be increased
  + Storage options in Azure include: Azure File system, Azure Storage Containers, and Azure Blob Storage.
  + “Draw It or Lose it” has 200 8MB Base Game Images to be delivered to the application would be stored in Azure File Share, which is 1.6GB of storage per user.
    1. <https://azure.microsoft.com/en-us/services/storage/files/#features>

- Cloud storage is quickly becoming a widely used storage solution due to its extreme cost efficiency and scalability

1. **Memory Management**:

Windows 10 is the newest version of the Windows operating system and has once again improved its memory management to allow for faster and more efficient loading from memory. This is primarily done through disc paging and demand paging to act as an extension of the computer’s physical memory or RAM. Disc paging does this by reserving part of the hard disk as extra RAM. Demand paging works by separating processes into smaller tasks, only to be loaded into memory whenever it is required for immediate processing. Each process in Windows 10 can also utilize the entirety of virtual memory address space, which is more than enough for the application.

1. **Distributed Systems and Networks**: Another reason we recommended going with Azure as your cloud service provider is its ease of access when dealing with distributed systems and Networks.
   * Azure offers maximum uptime with cloud-based email alerts, Azure App insights Logging, and monitoring solutions offered. This will be critical when scaling up to support 1000 concurrent games with 4 players in each game.
   * By offsetting the network load onto Azure you allow yourself more focus on the application and its features.
2. **Security**: Azure allows for simplicity in the logistics of security of user information and personal data. The devices will connect to the “Azure App Service” running on an App Service Plan, which will use Azure Active Directory for logging in. Some extra features offered with Azure are:
   * IP configurations for whitelist specific access to resources (like player or personal information) or the entire app.
   * The option for storage in a VPN within the cloud for enhanced security
   * Database could be IP Whitelist access only, have a password, and require SSL connectivity to protect the user data.
   * Options to obfuscated user data to protect personal data and information in case of a breach.

Note: An active internet connection is required for communications with the cloud.