

<Name-of-Software-Application>

# **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 | <05/24/23> | <Brian Blackman> | <Creating the classes and the functions for the game app> |

**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)

We will be creating a web-based version of Draw It or Lose It. The game consists of at least two teams guessing what is being drawn. There are four rounds lasting for one minute. The image will take thirty seconds to be fully complete. The team that’s guessing will have to guess what is being drawn within the time limit if not the other team gets fifteen seconds to guess one time. This app will have to be able to perform on all devices such as PC, mobile or table. We will need to take into consideration the application memory and be able to create teams and unique name players. We will be solving this problem with java programming to provide an object-oriented program.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

**Some design constraints are:**

Ability to play across different platforms.

Must have enough Storage space.

Must fit different Screen size.

Bandwidths need to be Taken into consideration to allow fast speeds between player communication.

And copyright issue

## [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)

As shown below in the UML diagram, we will have the Entity class which will create a entity which will be a id number and a name. Below the Entity class there is Game, Team, and Player Class. These classes Inherent from the entity class to create a player, team, or a game. The “+/-” shows the encapsulation of the variables within the classes. The “+” being public which allows other classes to call on them and the “-” which protects the variables from the other classes. The GameService class which also have an association with the game, team, and player classes, which allow the game to be set-up we one instance at a time. The 0…\* (which means zero to mean) shows that the game, team, or player will be unique. The programDriver runs the program and the SingletonTester which act as the users and runs a demo of the app.

**"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 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:**   * Intuitive GUI * Unlimited user license   **CONS:**   * Only runs on Apple hardware * Most costly | **PROS:**   * High security * Integrated open-source software * High-level language compilers * Ability to control the system using a GUI   **CONS:**   * Expensive training courses * Low third-party app support | **PROS:**   * Intuitive GUI * Symmetric multi-processor systems * Third-party application support   **CONS:**   * user-based licensing * more virus security threats | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.>  **PROS:**   * mobile cloud   **CONS:** |
| **Client Side** | Very costly | Cheapest | Must beef up Security | More expensive to maintain |
| **Development Tools** | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mac.>  Python  C++  Java  XCode  NetBeans  Eclipse  Unreal Engine  Google Cloud Platform  Unity  Adobe AIR  Scratch  Godot 2.0  Gdevelop  Torque2D  Torque3D | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Linux.>  Python  C  C++  Lazarus  Unity | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Windows.>  C++  C#  C  PHP  JavaScript  Visual Studio  Eclipse | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mobile Devices.>  Python  C  C++  Android Studio |

## 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**: Linux OS
2. **Operating Systems Architectures**:
   * **Kernel:** The kernel is the most important part of the OS it carries out the main task in Linux OS which would be operating the hardware.
   * **System Library:** These are the special functions or programs using which application programs or system utilities access Kernel's features. The libraries implement the functions of the OS without kernel module’s code access right.
   * **System Utility:** These are the unique programs that do specialized tasks.
   * **Hardware:** This is the CPU, RAM, Keyboard, Mouse, etc.
3. **Storage Management**: Logical Volume Manager
4. **Memory Management**: The memory management for Linux OS manages the memory inside the system and can use techniques like swapping or mapping memory.
5. **Distributed Systems and Networks**: Linux can run on any system which will allow the game to be run on any device.
6. **Security**: Linux OS offers password protection, controlled access to files and encryption of data

References:

*Operating system - linux*. Online Courses and eBooks Library. (n.d.). https://www.tutorialspoint.com/operating\_system/os\_linux.htm#