

Gamimg Room

# **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)** 6

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/22/2022 | Chao Guan | Background, constraints and domain model discussed |
| 2.0 | 06/06/2022 | Chao Guan | Evaluation of various OS for hosting the game |
| 3.0 | 06/14/2022 | Chao Guan | Recommendations for the system architectures |

**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 would like to develop a web-based game named Draw It or Lose It that can serve multiple platforms. This game is only availabe in an Andriod app. The client will need the game to support multiple teams with multiple players in each team to guess randomly selected images with the certain rules of time limits.

## [Design Constraints](#_2et92p0)

1. The application will need to be run on the web-based distributed environment.
2. The application will need to allow teams and players assigned to the game.
3. The applicaiton will need to check no duplicating team names created.
4. The applicaiton will need to allow only one instance of the game to exist at any given time.

Since the app is currently running on the Andriod platform, the software development team will need to decide either to create new codes or modify original codes so that the app can be run in other platforms such as MacOS, Windows, etc. In addition, the development team will need to choose right data structure to avoid duplicate team names. The team also needs to implement singleton pattern for the game design.

## [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)

Firstly, the UML class diagram indicates that class Game, Team, and Player all inherit from the Entity class. Thus, Entity is the super class and the other three are the child classess.

The Game class, the Team class, and the Player class inherit from the Entity class. The Game Service class has the association of multiplicity with the Game class. The “0...\*” notation means that one GameService could have 0 or multiple Game. It goes the same with the Game class to the Team class and so on. The ProgrammDriver class has a directed association with the SingletonTester class.

The principle of inheritance is demonstrated in this UML diagram. Entity is the superclass, and Game, Team, and Player are the subclasses. By inheritance, the subclass is able to inherit all features from the superclass. For example, all the subclassess will have the variables such as id, name and the methods such as getId(), getName(), etc. Inheritance allows one object to acquire features from the object so that the programmer does not need to add those features time to that object.

**"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** | Apply company just announced that it [discountinue macOS server](https://support.apple.com/en-us/HT208312) on April 21, 2022. Even though existing macOS Server customers can continue to use, the service will not be updated. | Linux is suitable if the website does not require scripting support. Linux web hosting is open source and compatible with many web-hosting softwares but not with Windows applicaitons. | Hosting the web-based app is easier because Windows provide many suports, but at a cost. Window servers will need require robooting for proper running. And the hosting can be troubled with a high volume of visitors. | Not all the mobile devices can support the web-based games, but they become more popular for its portability. |
| **Client Side** | MacOS provides a high-level of security for its users. However, the server is discountinued for new users. | Linux is free and thus appeals to customers with limited budget.  High-level of expertise is required for operation, such as command line knolwdge | Windows is very user friendly and the operation does not require much expertise. The resource loading is fast thus does not require a lot of time. But it is quite susceptiable to virus. | The cost can be high as it requires development costs, maintenance costs, and some other hidden costs.  Mobile Devices provides highest level of user comfortability. It is portable and the user interface makes it convenient for operation. |
| **Development Tools** | Object C and Swift are the mainly used language in developing software for macOS. The recommended IDE is Xcode | Object C and C are the main programming languages used to develop applications in Linux. The recommended IDE is Visual Studio | C++ and C# are the main programming languages used to develop applications in Windows. The recommended IDE is Visual Studio | Java is the default language to develop Android apps.  The recommended IDE is Eclipse. |

## 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**:The recommended operating platform to start with is Windows. Window OS has the the largest population base among all current operating systems. Besides, many gaming softwares are operationable in the Window OS, which provides great user experience and requires little expertise on for the operation.
2. **Operating Systems Architectures**: The most updated Windows OS has two architectures: 32-bit and 64-bit. The 32-bit architecture is more compatible with resources of older versions, both software and hardware, but the 64-bit provides newest features of Windows OS. One of the biggest differences is that 32-bit can only utilize 3.2-GB RAM whereas 64- bit can utilize much more RAM. To maximize the computing performance, we recommend for 64-bit Window OS. Regardless of the two Windows architectures, Windows OS provies convenient Win GUI (Window Graphic User Interface) for game players to interact with the game applications.
3. **Storage Management**: An solid-state drive, or SSD, is recommended as the storage system because SSD utilizes the newest technology to store data which can be accessed instantly on the memory chips. It runs faster but quiter compared to traditional Hard Disk Drive. Besides, the price difference is insignificant. It is highly recommended to choose SSD.
4. **Memory Management**: Since the game requires a library of 200 pictures, it will requires a good RAM to create the game while maintaining other operations such as media, antivirus processes, etc. In this case, 64-bit Windows with over 4 GB RAM is more appropriate.
5. **Distributed Systems and Networks**: Each game player will have play the game with different operating platforms. Thus, to scale the game to a distributed environment, all players from different OS will need to connect to the Server where the gaming information is stored and upated synchronously. To implement the this, the developers will create a shared database that allows access from other platforms such as Windows, MacOS, Android, etc. In the game application, there should also be a user interface which can provide the current network status while the playing is on the game.
6. **Security**: The Windows OS has default anti-virus protection. The game application should allow the running of the anti-virus software while connecting to the Server during gaming. Windows OS users can use the built-in options to encrypt sensitive data files so as to minimize the outcome of the data compromise. For users from other platforms, it is recommended to turn on the anti-virus software to scan for possible malicious codes or apps.