

Draw it or Lose it Application

# **CS 230 Project Software Design**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design** 1](#_Toc135590011)

[Table of Contents 2](#_Toc135590012)

[Document Revision History 2](#_Toc135590013)

[Executive Summary 3](#_Toc135590014)

[Requirements 3](#_Toc135590015)

[Design Constraints 3](#_Toc135590016)

[Domain Model 3](#_Toc135590017)

[Evaluation 4](#_Toc135590018)

[Recommendations 5](#_Toc135590019)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/21/2023  Updated on: 06/04/2023  **Updated on:**  **06/18/2023** | Ashley Johnson | Contains the summary for the requirements for The Gaming Room, the constraints for the design, and the domain model for the new game application that is to be develop for the company. |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room is deciding to develop a web-based game for their game Draw it or Lose it that can be ran across multiple platforms not just android. The purpose of the game is having multiple teams that consists of several players play four rounds with a minute time frame for each round. When a picture is pulled from the library of images one team will be able to guess until the time runs out. If the team that is guessing cannot answer, then each opposing team will get to answer until 15 seconds runs out.

## Requirements

* *Each team needs multiple players assigned*
* *Game and Team names must be unique*
* *A game will have the ability to have one or more team*
* *Only one instance of the game can exist in memory*

## [Design Constraints](#_2et92p0)

* There needs to be one or more team involved
* The teams need to have multiple players
* Game and Team names have to be unique to allow the players to check and see it that name is in use or not
* Only one instance of the game can exist at any time
* Must run on multiple platforms

## [Domain Model](#_8h2ehzxfam4o)

The UML diagram provides/displays several classes including Game, Entity, and Program Driver. Program Driver holds the main function which uses SingletonTester to guarantee unique entries. The classes Game, Player, and Team inherit variables and functions from the parent class Entity. The game class works with the class GameService to go over the game’s arraylist and initiate the SingletonTester class.

**"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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Simple but highest cost for server development with limited hardware options  Advantages: Good graphic interface  Easy accessibility and server configuration  Weakness: Limited software options  Expensive | Cost-effective for server development with constant upgrades but lower compatibility with software  Advantages: Open source, Secure, Free/Cost effective, Easy accessibility and server configuration  Weakness: Less Support | Less secure compared to the other options but has a wide variety of hardware options and a large support community  Advantages: Open Source, Secure, Support Avail, Vast software options  Weakness: Proprietary, Paid Service/Adds to cost | Hidden from users and must be integrated with cloud services  A challenge in compatibility across all platforms(iOS, Android)  Different specifications depend on the user |
| **Client Side** | * Limited product options * Generally safe due to OS security * High customer satisfaction * More Expensive for users * Moderate expertise and time required | * Lowest system requirements * Free virtual machine options * Runs on a wide variety of hardware * Maximum expertise and time needed * Not expensive for users | * Most used OS with plenty of supported browser options * Most familiar OS * Works on most current hardware * Minimum expertise and time needed * Cost is in line with Mac | * Varied experience through many different UI designs * Flexibility * Minimum expertise and time needed * Can be difficult to use on all mobile apps for different mobile platforms |
| **Development Tools** | May require Apple approval to release the app but can use IntelliJ or Eclipse as IDE  Development Tool/IDE- Xcode, Eclipse, A certain version of visual studio  Programming Language- Swift is mainly used but can use JavaScript, Python, CSS etc. | Open source and free licensing can use Eclipse for Java development  Development Tool/IDE- Eclipse, Github, Visual Studio, Repl.it etc.  Programming Language- C is the default but can use Java, Python, Ruby, HTML etc. | Simple to code with a wide variety of IDE options and language integration is available  Development Tool/IDE- Visual Studio, Eclipse, Repl.it etc.  Programming Language- C++, Java, Python, HTML | Different IDE and developer paths depend on the operating systems  Development Tool/IDE- For Android: Android Studio. For iOS: Xcode is mainly used but other IDE can be used  Programming Language- Swift, Objective C |

## 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:** Windows due to the operating system being used with a large variety of development options and having a large support community. Also, Windows can allow the client to reach a larger customer base out of the other operating systems. **I would still recommend the company to start with Window devices due to there being more software available paired with the minimum expertise and cost to get the projects going.**
2. **Operating Systems Architectures**: Windows NT is a hybrid platform that has two layers: physical and virtual architecture. The physical layer can be considered the now layer and only has access to the necessary resources. The virtual layer has access to the hardware and can pass the resources found on the physical layer based on what is running and what is needed. **I would still recommend Windows due to Windows providing the services that are used by all applications that are Windows-based which can enable applications to show GUI or Graphical User Interface alongside accessing resources in the system.**
3. **Storage Management**: OneDrive is free to use with a Microsoft account and offers a beginning capacity of 5GB of storage. If the user has a subscription to Office 365 then they get 1TB of storage. **I would still recommend OneDrive but if OneDrive does not work, Windows also offers storage sense which allows the user to manage files on the hard drive along with the amount of space being taken up. Storage sense also allows the user to choose different save locations for easy access, easy file creation, and placement for any large projects.**
4. **Memory Management**: Windows utilizes an internal feature for memory management called Windows Memory Management. Windows Memory Management takes a portion of storage space and addresses it for RAM. Windows will move processes that are not being accessed by Windows Memory Management to protect the physical RAM. **I would still recommend Windows memory management due to there being the need to create a database/library to hold a large number of pictures for the game.**
5. **Distributed Systems and Networks**: Developing Draw it or Lose it to be distributed to be released on the Xbox can be beneficial due to it allowing safe distributions and sales from a trusted platform that is built with Windows. **I would now recommend Develop 4 because Develop 4 offers cross-platform game creation and the IDE is able to run on any device. Once a game is created the user is able to export the file and import it into iOS, Android, or the web. The only issue is to prevent outages and connectivity problems, the company will have to make sure the servers are strong enough to withstand large volumes of players and have another power in case of outages.**
6. **Security**: Windows has a ton of security built into the OS. Windows Hello and Microsoft accounts protect the system from unauthorized users due to them having to be logged in. Windows Defender is an antivirus software that is a standard that comes preinstalled on Windows devices. **I would still recommend Windows for security due to Windows having built-in security protection that scans the system for any potential malware, viruses, or threats that could harm the system.**