

<>

# **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 | 3/20/2023 | Ryan Alvarado | Version 1 |

**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)

## Requirements

*Game Room has requested a deployment of a web-game that runs on all available platforms. “Draws It or Lose It” being it’s name. It’s restricted to only the Android platform. The runs of the game has multiple teams of several people going rounds for minutes at a time. A picture is pulled from the image library. With a minute per round each opponent gets to answer for 15 seconds at a time.*

## [Design Constraints](#_2et92p0)

Needs a team or more to be involved

Team names must be unique and searchable for in use status.

Only one game at a time can be played at a time

Multiple people should be in each team

Multiple platforms need to be capable of running the game

## [System Architecture View](#_ilbxbyevv6b6)

These requirements

## [Domain Model](#_8h2ehzxfam4o)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

**"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** | Adaptable commands to manage and change the server.  Characteristics, popular with cloud services  Advantages, web services have different possible options.  Disadvantages, web services and less likely | Adaptable commands to manage and change the server  Characteristics, security and heavily preferred.  Advantages, flaws are caught early. Web hosting services are preferred.  Disadvantages, niche platform and limited support | Highly used platform compared to others.  Characteristics, dominant platform  Advantages, resource management is high, loading time lessen, comfortability is high for users  Disadvantages, easy for viruses to interact. | Mobile servers are less preferred.  Requirements are better on other platforms  Characteristics, popular in the mainstream, very mobile  Advantages, reach more players and cost-friendly  Disadvantages, weakened security, and only certain smart devices would be capable |
| **Client Side** | Skill level and time required are med realm.  Costs are like competitor IE Windows.  Requirements of the development process are compatible with all web launchers and mobile platforms? | Due to the niche platform, the time and skill needed are at maximum.  Low cost  What requirements of the development process are compatible with all web launchers and mobile platforms? | Limited skill and time are required.  Costs are like competitors IE Mac  What requirements of the development process are compatible with all web launchers and mobile platforms? | Adaptable for clients and developers to use for updates. Due to the mobility of the platform, implementation is challenging |
| **Development Tools** | When running languages on mac, swift is the popular option. Mac can run all languages. Languages like JavaScript, HTML, and CSS being supported by their libraries.  Java, Python, Ruby, and PHP. | Eclipse is usable on Linux along with notepad++. Many languages are usage. Languages like JavaScript, HTML, and CSS being supported by their libraries.  Java, Python, Ruby, and PHP | Easier than Linux to use but has the same functionalities,  Eclipse and programs like it.  Languages like JavaScript, HTML, and CSS being supported by their libraries.  Java, Python, Ruby, and PHP | Unlimited apps can be made using android and swift. These languages can run the three machines.  Languages like JavaScript, HTML, and CSS being supported by their libraries.  Java, Python, Ruby, and PHP |

## 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**: Gaming Room should start on Windows platforms, being that is mass adopted and has more software options. Since the skill needed is limited to minimum the cost along with it. With the added benefits of plentiful development environments.
2. **Operating Systems Architectures**: Windows services provide graphical user interface usage. This functionality allows access to system resources. Applications like Graphical User Interface, media programming, and web services. Account services can be used.
3. **Storage Management**: Windows 10 has storage sense as an option. This option allows for the management of files. Different locations can be used as save locations to simplify app building. Multiple app projects can be developed at the same time with limited saving confusion.
4. **Memory Management**: Memory storage in Windows let users have easier access to databases or libraries for their utilization. App construction can be all in one location for easy access and archiving. Plus Windows come with a cloud server for saving. Letting users have access to their files away from their main systems if need be.
5. **Distributed Systems and Networks**: Being that each operating system is different a relatable software needs to be used. A relatable IDE is Visual Studio. Being that VS has been known to work on mobile apps web and game development there is a high chance to produce a strong and working product over multiple OSs. Visual Studio has Python support making for a decent library to work with.
6. **Security**: Windows being one of the biggest OSs they have built-in security protection software. Defender can scan for threats, malware, or possible system change. This on board protection cuts down on virus protection on the project’s budget and has the backing of Microsoft behind it for insurance.