

< Draw It or Lose It >

# **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 | <01/28/24> | Jeremy Raines | Changes were made to the cover page, document  revision history, executive summary, design  constraints, system architecture view, domain model  and recommendation. |
| 2.0 | <02/11/24> | Jeremy Raines | Further Elaborated on Development Requirements |
| 3.0 | <02/22/2024> | Jeremy Raines | Given Recommendation for Operating Platform. |

**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 is to develop a web-based game that serves multiple platforms based on the

current game Draw it or Lose It, which is only available on android devices. The purpose of the game is to have multiple teams consisting of several people going four rounds at a minute each. When a picture is pulled

from a library of images one team guesses till time runs out. If not answered each opposing team

member gets to answer till 15 seconds runs out.

## Requirements

* *A game will have the ability to have one or more teams involved.*
* *Each team will have multiple players assigned to it.*
* *Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.*
* *Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.*

## [Design Constraints](#_2et92p0)

* Must run on multiple platforms.
* Each team should have multiple players.
* Only one instance of the game can exist at any time.
* Game and team names must be unique to allow users to check whether a name is in use when
* choosing a team name.

## [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 Entity Class creates a relationship between the Game, Team, and Player Classes, meaning they’ll inherit or get information from the Entity Class. With UML we can show this with inheritance via making Entity a superclass. When we look at their relationship, we see Team and Player is a “has a” type, while Game has a Team and GameService has Games. When we use UML, we call it aggregation (HAS-A). When a user “has a” it means It's an instance of one class and has a reference to an instance to another class. When we look at this

diagram, we see GameService has a reference of Games, Games a reference of Teams, and Team a

reference of Players.

**"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** | easy accessibility and server configurability.  Easy to use  graphical user  interface.  Flexible terminal  Commands  OS X Server is available to develop on Mac, but, unless the client wants to purchase their own hardware for development, finding hosts will be not only hard but pricey as well. 499 is the price for 10 clients, 999 for unlimited clients | Cost friendly.  Difficult to  navigate the  platform.  Command shell  for simple server  configuration and  accessibility  Being the most popular hosting OS for its open source nature, prices should be the cheapest to host and develop for, and prices depending generally on which Linux variation, further mitigated by cloud based services such as those provided by Google and Amazon | Server side is  expensive. User  friendly GUI.  Has a command  Prompt  Due to being GUI based alot of applications linked to Windows Office will also work with the server as well, providing a great commodity  However, server licenses range from 6,200 ( for up to 16 core licenses) to 500 (for up to 50 clients) per installation per year. Which is much compared to other OS’s, and furthermore actual hosting platforms can be more limited than others as well. | Mobile device  specifications vary  from user to user  While mobile devices can be used as web and file servers, they’re not built for such and underperform, due to the limited hardware and the lack of scalability in servers. Any hosting tools would likely need to be built from the ground up and thus the cost is unknown. |
| **Client Side**  **Note\*: to ensure among each OS that compatibility exists between web browsers, cross browser testing is in order** | Expensive for  users.  Moderate time and  expertise required.  Accurate skills and  needed to navigate  OS as the SDK uses a language known as SWIFT, which is lesser known. Furthermore, Mac is only roughly 16 percent of market used OS’s vs Window’s 75% | There is a lot of  expertise and  time required but development itself should be straightforward as C/C++, Java, and Python – all languages of potential use -- are all commonly used in general. GNU/Linux does boast multi user support. | More expensive  than Linux.  Easy to learn and  understand.  Minimum  expertise needed. Ever since the release of Windows XP, Windows as a whole has been a native multi user platform, and as mentioned previously, makes up a majority of the market | Provide flexibility to both  clients and  developers to see  Updates, though slightly more  difficult to  implement than  other devices and development of applications themselves are straightforward.  While Android’s SDK is java based and thus can take more directly from coding and development on Windows and Linux, IOS is SWIFT based and requires all the base needs, hardware and software wise, as Mac |
| **Development Tools** | XCode is the usual IDE used for Mac and other apple product development, and the language used is SWIFT, which costs 99 per year per developer | HTML,  CSS and  JavaScript.  Libraries to  support frontend/languages.  Other systems include  JavaScript, Ruby,  PHP and Python  Many IDEs exist between languages, varying between free and paid, some examples being PyCharm for python or notepad++, and then the eclipse ide, which is capable of all supported languages | HTML,  CSS and  JavaScript.  Libraries to  support frontend  and languages.  Developer tools  include Eclipse and PyCharm, with the most popular being Microsoft visual studio, which supports many plugins and costs around 45 to 250 per user per year | HTML, CSS  and JavaScript. IDE’s for programming  languages consist of HTML, php, C++ and  Python.  Android’s primary IDE is Android studio, which is developed by google and free to download. However, IOS uses Xcode as Mac does, and thus inherits the same costs, 99 per year, per developer |

## 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**: From the research done, my recommendation would have to be Windows, primarily due to the amount of IDE’s that can be used and the ease of learning for developers compared to other Operating Platforms. While it may be more pricey in comparison, the ability to more easily learn the programming required should allow for ease of development, not only for desktop but mobile versions of the app as well, since Microsoft Visual Studio does allow for android app development.
2. **Operating Systems Architectures**: Windows separates its operating system into User and Kernel, with User mode affecting much of what the user interacts with while Kernel mode is lower level, dealing with things such as inputs and outputs, memory management, networking, hardware management, and routines.
3. **Storage Management**: The primary recommendation I have would be to go with Microsoft Azure for the constant updates and support provided, not to mention the general efficiency and cost effectiveness of cloud-based storage as a whole. It is also to note that Windows’ newer releases such as 10 and 11 come with a feature called storage sense, allowing for the management of files on one’s hard drive and the space they take up
4. **Memory Management**: The Windows storage sense feature would allow for storage and management of both photos and game players, as well as allowing them to be together in one secure space in memory. primarily done through disc paging and demand paging to act as an extension of the RAM. The former by reserving part of the hard disk as RAM while the latter works by separating processes into smaller tasks to be loaded when required.
5. **Distributed Systems and Networks**: Network based multi-user interaction systems such as network games (Like Draw It or Lose It) typically include a database shared among players, physically distributed and able to interact with one another over the network. Currently, developers must implement these from scratch.
6. **Security**: Windows comes with built-in security protection software. Though it would be highly recommended to search for other sources of security software and other general protective measures.