

<Draw-It-Or-Lose-It>

# **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)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 5/29/2021 | <Tierney-Shaughnessy> | <Version 1> |
| 2.0 | 6/4/2021 | Tierney Shaughnessy | Made corrections and added on new requests from client |
| 3.0 | 6/19/2021 | Tierney Shaughnessy | Added on new requests from client |

**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)

Draw It or Lose it by The Gaming Room is an Android app game that is loosely similar to the television game Win, Lose, or Draw that ran in the 1980’s. The game’s purpose is to allow multiple teams consisting of multiple people going four rounds at a minute each. But rather than drawing the images themselves, the app will pull up a drawing from a library of images and slowly show more of the drawing until it is complete at the 30 second mark. If the team fails to answer correctly, each opposing team gets to answer until 15 seconds run out.

## [Design Constraints](#_2et92p0)

* Requires multiple teams for the game to work
* Game/Team names have to be unique to allow users to verify whether the name is taken or available.
* Each team has more than one person.
* Needs to run on multiple platforms.
* Only one game instance can exist at any time.

## [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. They all will inherit information from the entity class. In UML, this is shown with Inheritance. Therefore, each inherited class will have items and variables in common with the Entity class. This makes Entity a superclass. Breaking it down further, we can see that Team and Player are instances of one class and have a reference to an instance to another class while Game has a Team and GameService has Game – their own references. This is called aggregation in UML.

****

## [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** | Mac has flexible terminal commands for configuring the server, access, or to make changes. Offers Server-Based Deployment.  Advantage:  it is upgradeable and has multiple options for various hosting requirements.  Disadvantage: It’s less preferred for web hosting services. Higher cost for software and hardware. | Also has flexible terminal commands but is more cost effective. Offers Server-Based Deployment.  Advantages:  Security flaws are caught quickly and don’t become as much of an issue and it’s the most preferred choice for web hosting. Server costs are much more affordable if not almost free depending on what one needs.  Disadvantage:  More difficult to find applications that can support the needs of web hosting. | There are more software programs available than compared to the other Operating Systems and it is dominant to the other OS. Offers Server-Based Deployment  Advantages: There is less loading time, higher comfortability, high resource requirements.  Disadvantages:  Very susceptible to viruses and a lower quality of tech support. Servers, hardware, and software can still can set one back a pretty penny depending on what you need. | More popular and very portable. Server-based deployment is available.  Advantages:  Has higher compatibility, a wider reach, and very cost-effective. There are companies that can offer server hosting if you do not have the machines for it yourself.  Disadvantages:  There is typically poor security and it is selective to various mobile devices. Potential display resolution issues as well. |
| **Client Side** | Has a moderate level of expertise and time required. The cost is similar to windows, for hardware and software as well as servers.  Frequent testing would likely be prudent and it will be needed to keep in mind that security can be tricky across platforms. | Requires the maximum level of expertise and time but has the minimum level of cost. Server costs could be nearly free depending on one’s needs.  Frequent testing still will be prudent and it will be needed to keep in mind that security can be tricky across platforms. | Least amount of expertise and time is required, and the cost is similar to Mac.  Frequent testing will be prudent and it will be needed to keep in mind that security can be tricky across platforms. | Clients and Developers have flexibility to see updates at any place, but it’s a little more difficult to implement than others. Would need added security and will likely be a weak link in terms of security issues. |
| **Development Tools** | For running languages on Macs, the more popular option is swift. Notepad++ is another useful tool and languages consist of but are far from limited to: Javascript, HTML, CSS. There are supporting libraries for the front end and general purpose languages such as Java, PHP, Python, and Ruby.  Multiple development teams would likely be the most efficient especially as it would allow for people to work solely on testing across platforms. | Linux works with Visual Studio, Notepad++, and Eclipse. Supports many more languages and tools which consist of but are not limited to: Javascript, HTML, CSS. There are supporting libraries for the front end and general purpose languages such as Java, PHP, Python, and Ruby.  Multiple development teams would likely be the most efficient especially as it would allow for people to work solely on testing across platforms. | Windows is easier to use than Linux and can run the same as Linux more or less. Therefore, Visual Studio, Eclipse, and Notepad++ will also work with Windows. Supports many more languages and tools which consist of but are not limited to: Javascript, HTML, CSS. There are supporting libraries for the front end and general purpose languages such as Java, PHP, Python, and Ruby.  Multiple development teams would likely be the most efficient especially as it would allow for people to work solely on testing across platforms. | Infinite apps can be created using android and swift. Both languages and software are able to be run on all three other machines as the languages and supporting libraries are all the same.  Multiple development teams would likely be the most efficient especially as it would allow for people to work solely on testing across platforms. |

## 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**: I would suggest to The Gaming Room to start on windows devices. It has more software available and the minimum expertise and cost to get projects going suits the client well. Moreover, one can expand to other computing environments and most cloud servers and storage options are compatible with Windows.
2. **Operating Systems Architectures**: The services provided by Windows are used by all Windows-based applications to enable apps with a GUI (Graphical User Interface) while also accessing system resources and more.
3. **Storage Management**: Windows provides the freedom to choose save locations for apps, to scrutinize and manage files on your hard drive, and use the cloud to save data. The built-in storage also has easy file creation and placement for large products as well as organization so you can easily find your files when needed. Most cloud storage services are compatible with Windows as well.
4. **Memory Management**: For the game, a library with lots of pictures will be needed. Windows memory allocation provides easy storage of pictures away from the default folder and includes when you’re working with your IDE and opening files to create the game.
5. **Distributed Systems and Networks**: For this project, a client-server system would make the most sense. Perhaps one that works with cloud servers depending on the budget.
6. **Security**: Windows does come with built-in security software but it’s highly recommended to use another source for protection. Firewalls, two-factor authentication on both the design and user end, and regular updates and testing are recommended.