

Draw It or Lose it

# **CS 230 Project Software Design Template**

Version 1.0

## 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 | 11/13/2020 | Andrew Vue | Initial project set up |

**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 design problems of the project are that the application needs to have access to a database of images to display for users. The initial issue is that the library of images needs to be large enough so that the same image will not be presented to the users within a given time frame and it also needs to be random. To do this we need to write code to pull the images from a database, have a randomizer set up and another layer to parse the image ID so that we don’t get a match for an image already used within the given time frame.

## [Design Constraints](#_2et92p0)

The initial design constraints I can see right off the bat are:

* The availability of an image library and the rights to use these images for the game/app
* Setting up an API to pull the images
* Creating ID for each individual image
* Connecting all of the different devices using this game

## [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 diagram below shows that different aspects of the game broken into chunks that will work together. The processes of creating each class and object to function as opposed to having one large file will make the game run smoother and faster. This allows for the developers to update and fix specific parts of the game as well. The Entity is what holds the player information and the Game, if there is a Team then players are a subclass of the Team. Game Services and the Game require Players or Teams to run.

****

## [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.*

There has not been enough development at this point in the development to be able to tell the performance difference between each of these operating platforms. From the looks of how things are so far, I predict that there to will not be any major differences in performance between the different operating platforms.

The only thing to look out for is the development of the client side interface. Moving forward designing the user experience and interface will rely heavily on the types of devices using this application. We must design with the wide array of device sizes and specific user hardware peripherals in mind.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | <Evaluate Mac for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Linux for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Windows for its characteristics, advantages, and weaknesses for hosting a web-based software application.> | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.> |
| **Client Side** | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mac.> | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Linux.> | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Windows.> | <Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients as they pertain to Mobile Devices.> |
| **Development Tools** | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mac.> | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Linux.> | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Windows.> | <Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on Mobile Devices.> |

## 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****: <Recommend an appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments.>*
2. ***Operating Systems Architectures****: <Describe the details of the chosen operating platform architectures.>*
3. ***Storage Management****: <Identify an appropriate storage management system to be used with the recommended operating platform.>*
4. ***Memory Management****: <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.>*
5. ***Distributed Systems and Networks****: <Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).>*
6. ***Security****: <Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.>*