

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)4

[**Evaluation**](#_2o15spng8stw)5

[**Evaluation**](#_2o15spng8stw)**(cont.)** 6

[**Evaluation**](#_2o15spng8stw)**(cont.)** 7

[**Recommendations**](#_m8aleynsvzvc)8

## 

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03/18/2021 | Uriah Fadum | Created Software Design Documentation  Completed Executive Summary  Completed Design Constraints & Tech Requirements  Completed System Architecture View  Completed Explanation of Domain Model |
| 1.0 | 04/03/2021 | Uriah Fadum | Started Development Requirements Table |
| 1.0 | 04/04/2021 | Uriah Fadum | Completed Development Requirements Table |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## [Executive Summary](#_sbfa50wo7nsh)

Gaming Company “The Gaming Room” operates an android OS application interface, called “Draw It or Lose It”. In order to effectively target a broader base of users/devices, they have decided to develop a web-based version of the application. Several key issues will need to be addressed starting with reviewing existing android application server/services, application architecture and existing client-server design (including programming languages used). After reviewing current application operations, careful consideration will determine the appropriate integration of both the original Android App and the new Web-Based version of the game. Considering the likelihood of the original android application being written in either C, C++, JavaScript or Java, our plan to move forward includes using Java as the main language used to create the web-based version of “Draw It or Lose It”.

## [Design Constraints](#_2et92p0) & Software Requirements

Web-Based Interface (Multi-OS Architecture Compatible)

Game(s) have the ability to host multiple teams

Team(s) have the ability to host multiple players

Unique Game and Team names to verify gaming service/name not already in use

Dev-Op Personnel (Current & Proposed)

Web Server/Application Hosting Services (Current & Proposed)

Programming Language(s) (Current & Proposed)

Web/Application Security (Current & Proposed)

Server Security (Current & Proposed)

Client Security (Current & Proposed)

User Integration (Account)

Legal/Licensing

Marketing/Advertising

Testing/Beta-Testing

## [System Architecture View](#_ilbxbyevv6b6)

Some considerations would be to implement a n-Tier or multi-tier architecture. This would then address issues of scalability, bandwidth and system control that would be faced when designing an application and implementing a strategy for integration and security.

## [Domain Model](#_8h2ehzxfam4o)

The java package com.gamingroom contains the source code for creating and initiating unique games with unique teams that are made up of one or more players. Within the code is the ability to control the creation of game specific service with the singleton design pattern, thus the singleton tester class. Also within the application is the program driver class with the ability to initiate and test all other package classes. The other four classes are considered the working parts of the actual package. The Entity class is the base/super class of game, team and player classes and follows industry standards and best practices by implementing inheritance and polymorphism when creating game, team and player objects. The game service class is the game engine with the ability to create and initiate unique games.

****

## 

## 

| **Evaluation** | | | | |
| --- | --- | --- | --- | --- |
| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| **Server Side** | **Advantages**  -MacOS X Server is User-friendly when used w/ ONLY Apple products.  -Quality server support and products.  -Does NOT use Active Directory natively but can be connected to a Windows server/network using AD via LADPv3.  - Higher security due to the use of proprietary technology.  **Disadvantages**  -NOT easily integrated w/ other platforms without in-depth experience in MacOS X Server management.  -High initial hardware/software cost to initialize and host a website application.  -High cost to scale up, due to equipment costs. | **Advantages**  -Easily able to deploy, scale-up & manage a web application server/service.  -Most software is free and regularly updated.  -Low hardware & software cost to initialize and host a website-based application.  -Does NOT use Active Directory natively but can be connected to a Windows server/network using AD via LADPv3.  -Easily able to incorporate cloud-based features & services.  -Can operate on most computing equipment.  -Highly adaptable to computing environments and systems.  **Disadvantages**  -Large attack vector for security vulnerabilities.  -Expert knowledge required to create, operate and manage server operations. | **Advantages**  -Easily able to deploy, scale-up & manage a web application server.  -Quality server support and products.  -Easily able to incorporate cloud-based features & services.  -Uses Active Directory Protocols natively.  -Widely used server environment.  **Disadvantages**  -Large attack vector for security vulnerabilities.  -Expert knowledge required to create, operate and manage server operations. | **Advantages**  -Mobile…  **Disadvantages**  -expensive to program, deploy and maintain.  -Does NOT use Active Directory or LDAP natively and would require additional software to accommodate AD & LDAP use.  -Cap on scalability.  -Cap on Bandwidth.  -Cap on Application size. |
| **Client Side** | -Designing a web-based application that would utilize native safari browser features (Desktop & Mobile) would be top priority when assessing the MacOS X & iOS platforms.  -Would require expertise in Swift, Objective-C and other OS related languages. | -Designing a web-based application that would utilize Firefox browser features (Desktop & Mobile) would be top priority when assessing the client side of different Linux platforms.  -Would require expertise in programming in C, C++, Java and other mainstream programming languages (i.e., Python, Perl, etc.). | -Designing a web-based application that would utilize the native Microsoft Edge browser features (Desktop & Mobile) would be top priority when assessing the client side of the different WIN32, 64 & Windows 10 mobile platforms.  -Would require expertise in programming in C, C++, Java and other mainstream programming languages.  -Does have Licensing costs when using Microsoft features & services. | -Designing a web-based application that would utilize the native Android mobile browser features would be top priority when assessing the client side of the different mobile platforms.  This would have to be addressed & researched since some mobile device manufactures produce different variants of browser applications with their products (i.e., Huawei browser running on Harmony OS 2.0+, KaiOS,  etc.)  -Would require expertise in programming in C, C++, Java as well as a comprehensive background in Mobile OS architecture and developmental tools used. |
| **Development Tools** | Most modern web browser technology has been programmed at the core using C++ for developing purposes, utilizing other languages to best suit the need of whatever web function is being addressed and whatever core OS is being used.  -MacOS X & iOS use C, Objective-C and others like Swift and JavaScript + corresponding IDE’s. | -Linux systems utilize a majority of the mainstream programming languages and IDE’s (i.e., Eclipse, etc.)  -offers the ability to effectively program in every language.  -Linux has a large collection of corresponding program language libraries & tools.  -Excellent cross-platform programming capabilities. | -Win32 & 64 based platforms use C# and C++. Other languages are used such as Java, HTML and others and for the purposes of a webpage-based game on Microsoft Edge, these would be the main programming languages used, plus corresponding IDE’s. | -Most mobile applications are based on Android OS and are developed using Java, C, C++, HTML5 and the use of Android Studio. If not, then they are proprietary to their corresponding manufacturer and can use whatever language IDE the manufacture of the device uses (i.e., iOS=Xcode, Linux Mobile=Xaramin, etc.). |

## 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.>