

**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 | 01/23/21 | Michael Bauer | Added executive summary, design constraints, and domain model |
| 1.1 | 02/22/21 | Michael Bauer | Added recommendation section |

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wants to bring their Android app, “Draw It or Lose It,” to a web-based game in order to serve multiple platforms. The web-based game will have to be rewritten in another language in order to make it web-based and serve multiple platforms. This web-based game will need to be hosted on a web server. This web-based game will have the same features as the Android app.

## [Design Constraints](#_2et92p0)

* The web-based app needs to be written in a language for the web
* The web-based app needs to be hosted on a web server
* The web-based game will have the same features as the Android app

**Rationale**

The Gaming Room wants to bring their Android game to the web and this provides a technical constraint because the web-based game requires a different language. The web-based game will need to be hosted on a web server and this provides a business constraint because this will cost to host. The web-based game will have the same features of the Android app, this provides a technical constraint since these features need to be provided with a completely different language.

## [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)

Entity is the parent class for Game, Team, and Player classes. The Player class extends the Entity class and adds two more methods. The Team class extends the Entity class and adds one attribute as well as three methods. The Team class can have zero to any number of players of the Player class. The Game class extends the Entity class while adding one attribute and three methods. The Game class can have zero to any number of teams of the Team class. The GameService class can only have one instance running. The GameService can have zero to any number of games of the Game class. This is a singleton pattern to only allow one instance of the GameService class to be running.

****

## 

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | **Pro** – More secure from malicious users, reliable (does not require constant updates)  **Con** – More expensive than others, does not have access to Microsoft proprietary software | **Pro** – More secure from malicious users than Windows, reliable (does not require constant updates), cheaper than all  **Con** – Hard to find pre-built machines, more difficult to use for some people, does not have access to Microsoft proprietary software | **Pro** – Cheaper than Mac, Access to certain proprietary applications  **Con** – Less secure from malicious users, has more updates causing server downtime | N/A |
| **Client Side** | In order to support multiple browsers, it would be best to use things that are compatible on multiple browsers. Expertise within HTML is important, having the knowledge of cross-browser compatible libraries and frameworks is important to know. Having expertise in CSS is also important in order to understand how to reduce browser inconsistencies across multiple browsers. As an example, using the Foundation framework would help to ensure cross-browser compatibility. | | | |
| **Development Tools** | **Programing Languages** – HTML, CSS, and Javascript  **IDE** – Some examples include: Visual Studio, RJ TextEd, and Light Table  **Other** –  **Javascript** **libraries**: examples include jQuery, Dojo, and React  **CSS** **preprocessor:** like Sass  **Frameworks:** like Foundation and Bootstrap | | | |

## Recommendations

1. **Operating Platform**: A Linux based platform will be cheap, reliable, and secure
2. **Operating Systems Architectures**:
   1. **Hardware layer** – Consists of RAM, hard drive, CPU, peripherals, etc.
   2. **Kernel** – Responsible for main operations of the operating system, communicates with hardware layer
   3. **Shell** – users interact with the shell which then uses the kernel to execute those commands
   4. **Utilities** – programs that provide functionality of the operating system to the user
   5. **System Library** – A function that programs and utilities can use to access kernel features (operating system functionalities)
3. **Storage Management**: RAID 5 will offer redundancy and still perform well with keeping in mind of budget. This requires less hard drives than RAID 1.
4. **Memory Management**:
   1. CPU cache and RAM are used in memory management. The process is loaded into RAM and then goes to the CPU. From the CPU it is loaded into the cache of the CPU.
   2. The page cache stores recent data calls. Anytime there is something that needs to be done to data on the disk, it first is read from the disk into RAM. The part of RAM that stores this data is called page cache.
   3. The dirty cache is used when data is modified in the page cache, the page for that data is marked dirty. The data modified needs to be written to the disk, a dirty cache stores the dirty pages that have not been written to the disk.
   4. Virtual memory is used as a memory reservation. Pointers are used to make the reservations and once there is a reservation that memory cannot be reserved by anything else.
5. **Distributed Systems and Networks**:
   1. Using distributed systems, especially a three-tier system will allow access from various platforms. The three-tier system is a distributed system that has three layers the presentation (client) layer, the application layer, and the database layer. The presentation layer is what the user interacts with. This will be accessed through the user’s browser that connects to a web server. The user interacts with the presentation layer which in turn passes information to the application layer. The application layer is usually on a different server and processes the information that is passed to it from the presentation layer. Then the application layer communicates with the database layer in order to accomplish the user’s requests or inputs. The database layer then executes the actions needed and sends information back to the application layer. The application layer processes this information and sends it back to the presentation layer. This will allow different browsers and ultimately different platforms to access the webserver and interact with the application.
   2. Each layer is usually on a different server, this allows for optimum performance for each layer. Each layer will have to connect to the next layer to communicate to it which can be an issue if a connection fails or can be a bottle neck that slows things down.
   3. One option is to use the cloud to host the layers which can help with performance and reliability. Using the cloud also offers easy scalability but also less control over everything.
6. **Security**:
   1. To start, controlling who can access the application using a username and password is a must. Allowing multi-factor authentication can add to the security of the username and passwords. Being able to control what the users can access will also limit possible security issues. Encrypting communication between the layers of the distributed system will limit what can be sniffed out or seen by users. Encrypting part of the database that holds user account and any other information that is deemed sensitive is important to secure that information.