

Draw It or Lose It

**CS 230 Project Software Design Template**

Version 1.0

**Table of Contents**

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 1.0 | 07/15/2021 | Steven Stutts | To discuss a plan to meet the needs for the 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**

The client, The Gaming Room has hired our consultant firm to develop a web-based game that serves multiple platforms based on their current game, Draw It or Lose It, which is currently available in an Android app only. Our firm must get with The Gaming Room personnel and discuss what their needs are so we can sketch out a diagram with the key attributes they need for their web-based game. We will also identify the design constraints and show The Gaming Room personnel how we will go about solving these particular constraints. We will also discuss what kind of budget they are looking for everything they need to get done.

**Design Constraints**

*(Design constraints are conditions that need to happen for a project to be successful.)*

-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.

These constraints must be addressed so we can meet the requirements of the gaming room. Each of these constraints pose a problem if not implemented and The Gaming Room personnel will not be happy if not addressed. We will resolve these constraints by implementing a program that will solve these issues.

**System Architecture View**

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**

The ProgramDriver class and the SingletonTester class would be use case. The arrow with the uses is an additional relationship arrow. The plus sign means that their methods are public. The remaining 5 classes are showing the inheritance object oriented principle. The 3 classes on bottom (Game, Player, and Team) are the child classes inheriting from the parent (Entity) class. GameService can have 0 to many games, Game can have 0 to many teams, Team can have 0 to many players. The attributes are private members and the methods are public. These classes are arranged in a way that will allow the software to run efficiently and correctly. They explain how these classes will be coded into the program. The blueprint so to speak.



**Evaluation**

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** | Flexible terminal commands to configure the server, access, or make changes. | Flexible terminal commands to configure the server, access, or make changes and more cost friendly | More softwares available compared to other OS. | It's better if the server is immobile and can be tracked at a single place. |
| **Client Side** | Moderate expertise and time required. Cost similar to windows. | Maximum expertise and time required. Minimum cost. | Minimum expertise and time required. Cost similar to mac. | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | Common languages:HTML/  CSS/JavaScript and supporting libraries to support the frontend and general purpose programming languages like Python, Java, php, Ruby on Rails.  Tools: PyCharm, Eclipse, Visual studio, Github, Notepad++, databases (eg. mongoDB,SQL,Cassandra)  These languages/tools are availablefor all OS. | Common languages:HTML/CSS/JavaScript and supporting libraries to support the frontend and general purpose programming languages like Python, Java, php, Ruby on Rails.  Tools: PyCharm, Eclipse, Visual studio, Github, Notepad++, databases (eg. mongoDB,SQL,Cassandra)  These languages/tools are availablefor all OS. | Common languages:HTML/CSS/JavaScript and supporting libraries to support the frontend and general purpose programming languages like Python, Java, php, Ruby on Rails.  Tools: PyCharm, Eclipse, Visual studio, Github, Notepad++, databases (eg. mongoDB,SQL,Cassandra)  These languages/tools are availablefor all OS. | Common languages:HTML/  CSS/JavaScript and supporting libraries to support the frontend and general purpose programming languages like Python, Java, php, Ruby on Rails.  Tools: PyCharm, Eclipse, Visual studio, Github, Notepad++, databases (eg. mongoDB,SQL,Cassandra)  These languages/tools are availablefor all OS. |

**Recommendations**

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

**Operating Platform**: Microsoft Windows

* **Operating Systems Architectures**: The architecture of Windows is a line of operating systems produced and sold by Microsoft, is a layered design that consists of two main components, user mode and kernel mode. It is a preemptive, reentrant operating system, which has been designed to work with uniprocessor and symmetrical multi- processor SMP based computers.
* **Storage Management**: Microsoft Azure is a highly secure, massive scaled cloud storage for windows.
* **Memory Management**: Microsoft provides an resource describing Windows Operating System Memory Management on their Windows Development Desktop technologies site. Each process on 32-bit Microsoft Windows has its own virtual address space that enables addressing up to 4 gigabytes of memory.
* **Distributed Systems and Networks**: Distributed File System (DFS) is a set of client and server services that allow an organization using Microsoft Windows servers to organize many distributed SMB file shares into a distributed file system. DFS has two components to its service: Location transparency (via the namespace component) and Redundancy (via the file replication component). Together, these components improve data availability in the case of failure or heavy load by allowing shares in multiple different locations to be logically grouped under one folder, the "DFS root".
* **Security**: Windows Security is available to manage the tools that protect your device and your data: virus and threat protection monitor threats to your device, run scans, and get updates to help detect the latest threats. Account protection, access sign-in options and account settings, including Windows Hello and dynamic lock.