

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 | 01/20/2022 | Eduardo Orozco | N/A |

**Executive Summary**

Creative Technology Solutions is a company that would like to create a web-based game "Draw it or Lose it" which has resemblance to the 1980's television game *Win, Lose, or Draw*. The game requires that the game has the ability to allow one or more teams involved, to have multiple players assigned to each team, the game and team names must be unique, and only one instance of the game can exist in the memory at any given time.

**Design Constraints**

The main design constraint would have to UI design and interface. Since, this will be a web-based game, the Ui has to be designed to incorporate a variety of platforms for palyers to play on. The other constraint would have to be inclusion of a network database that will allow the players to interact and connect the data between the various platforms.

**Domain Model**



**Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| **Server Side** | * Using Mac clients can come at a disadvantage for ease of us when it comes to Mac OS development. * The disadvantage is you are limited on the hardware usage. | * Linux is one of the better options for web-based software application since it is Unix-based. * Unix is known for their stability and reliabilty. | * Window is another great option, the advantage is that it locates user files on a web server that uses windows OS. * The only disadvantage is that its compition is Linux which seems to be more reliable and stable. | * For mobile development, there is the hurdle of finding a provider to host the mobile app. |
| **Client Side** | * Mac can be the most costly in terms. * Mainly, if a client wants to us Mac OS they have to buy a product specifically from Apple. | * Linux is probably the best, and most cost effective OS there is for software developers. * The only requirement is finding a compatible Linux capable computer, which is extremely common. | * Windows is a great OS for develpoment. * Not only can you program native windows app, but there is the option to bootcamp to Mac OS and be able to develop there as well. | * Deveoloping for mobile devices can be dificult. * It has to be taken into account the smaller screen size than a computer, and UI design has to to fall under that smaller screen size, and just generally less features. |
| **Development Tools** | * The most common programming language in the Mac Os is Objective-C. | * Linux is generally really good with any programming laguages. * Includes: C, C++, Java, Javascript, Python, and even HTML. | * The most commonly used Language for Windows is C++. * Microsoft even has their own IDE known as Visual Studios. | * Not many options for programming, but for IOS there is Swiftic, and for Android Java is the official programming language. |

**Recommendations**

* **Operating Platform**: Windows would be the better recommended operating platform due to its access to multiple programming lanuages and IDE's. Also its web-server is on par with the higher end Linux. We will also not have to be subject to design contraints from mobile based platforms.
* **Operating Systems Architectures**: With windows and many other OS x86 is a very powerful architecture for development of games.
* **Storage Management**: HDD are very common, especially when it comes to price. However, SDD are much more recommended for their speed which is a drastic change from HDD.
* **Memory Management**: With the recommend OS windows, the 64-bit proccess can contain a virtual space of eight terabytes, which is double compared to the windows 32-bit. Also, each process on windows has access to the virtual address space, but threads cannot access other proccess, which keeps thos proccess safe from being corrupted.
* **Distributed Systems and Networks**: It is crucial for web based games to be able to connect and interact on various platforms. A data base must be created that will be shared amoung players and be able to interact with one another.
* **Security**: Security is obviously very important when it comes to clients sensitive information. The most recommended way of security is understanding where the sensitive data is stored and who has access to it. A need-to-know basis policy can help minimize risk by limiting access to entries, so that if someone has access to that data it does not go undetected.