

Kaitlyn Laflash

# **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 | 07/18/2012 | Kaitlyn Laflash | Updated project |

**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 Gaming Room is looking to develop a Web based game that is capable of running on multipl platforms. The game they want to develop will be named “Draw It or Lose It”, based on the 1980’s television show Win, Lose or Draw. The game is currently on a android platform and looking to expand to other platforms.

## [Design Constraints](#_2et92p0)

* One or more teams
* Multiple Players per team
* Unique Team Names
* Unique Game Names
* Only one game allowed at a time
* Expand to multiple platforms

## [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 domain model works ass follows. The entity Class creates a relationship between the game, team, and players. So this means that the Game, Player, and Team classes pull information from the Entity Class.

****

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Expensive. Excellent for ease of use for users. Mac has a pliable workstation for easy accessibility and server configuration. | Cost friendly. Difficult navigating the platform. Like Mac, Linux has a command shell for simple server configuration and accessibility. | Costly, like Mac. User friendly platform. Lots of Windows available software options. You have a Command prompt. | Inexpensive. Mobile device specifications vary from user to user; Android, iOS, WP. Can be difficult to create a game that is compatible with most mobile platforms. |
| **Client Side** | The time necessary for supporting Mac users is average. Adequate skill level is needed. Like Windows OS, the cost of Mac is more expensive than Linux. | Time and knowledge of Linux OS is at a maximum. Mastery of Linux is required. Minimal cost. | Minimal time and proficiency are required to support Windows OS users. Cost is relative to the cost of Mac OS. | Lots of time and ability needed to support diverse mobile device. mobile OS platforms are challenging to perform on other devices. |
| **Development Tools** | Tools: notepad  Programing languages: HTML, CSS, Java. Tools include libraries to support different languages, Eclipse, Visual Studio. | Tools: Ruby on Rails, Java, Python, CSS, JavaScript, HTML. | languages utilized is a mix of MAC OS and Linux: Ruby on Rails, Java, Python, C++, C#, JavaScript, HTML, etc. Programs used can be Visual Studio, Eclipse, Repl.it, command prompt. | Can use variety of languages: Python, Java, CSS, JavaScript, HTML, php, Ruby on Rails, C++, etc. Programs used to develop: Repl.it, nodejs, Github, Visual Studio, and command prompt. |

## 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**: My recommended operating platform is Windows. It has plenty of software availability and is cost effective. Windows is one of the most utilized operating platforms and most people have the knowledge to use and navigate Windows.
2. **Operating Systems Architectures**: Windows works with many different programming languages. By using Windows, the applications can use the GUI and have access to memory and other aspects of the Windows operating platform.
3. **Storage Management**: Windows had Storage and cloud services they are cost efficient. Windows had a disk clean up and storage sense that help manage storage space which is a benefit of using a windows operating platform.
4. **Memory Management**: windows had different levels of memory depending on the type of file, making it more compatible. The game needs this space for images and files that can be easily accessed.
5. **Distributed Systems and Networks**: Using a client server distribution system will benefit the program. This will allow each client to us a single server application and they can develop based on their systems strengths. The network needs to be a strong server, due to the high demand of clients.
6. **Security**: Windows has a security built into the system. Windows defender is the system that windows uses. All the data would need to be encrypted due to the amount of information exchange.