

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 3/19/23 | Michael Barbuzano |  |

**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)

<Write a summary to introduce the software design problem and present a solution. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.>

The Gaming Room development team wants to develop a web based game that serves multiple platforms, that is based on their current game which currently is only available for Android. The game is called “Draw it or lose it” and it involves two or more teams with one player drawing an image within thirty seconds, then their team must figure out what the image is within a time limit, if they can not guess correctly, the remaining teams will have an opportunity to guess within a fifteen second time limit.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

Requirements

* 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 with a unique identifier assigned to each instance of a game, team or player.

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

A web-based application is a program that is accessed over a network using an HTTP connection. Although creating a web-based application can be very cost effective, it does come with some design constraints. One example is reduced speed of the application, this limitation of speed is common in web-based applications as they do not operate on a server locally. Web based applications can also increase risk of a breach of data.

## [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)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## 

In the UML diagram, it is shown that the Game, Team, and Player classes inherit data from the Entity class, this demonstrates the object-oriented programming principle of inheritance. It is also shown that GameService has a reference of Games, Games has a reference to Team, and Team has a reference to Player. This is important because each GamerService has a Game, each Game has Teams, and each Team has Players, so each class must have a reference to the required class so that it can access an instance within that class. The use of classes to store data and methods demonstrates the object-oriented programming principle of encapsulation.

## [Evaluation](#_2o15spng8stw)

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** | Pros: Advanced software security, various subscription plans  based on web hosting requirements.  Cons: Price | Pros: Lowest cost.  Cons: Less compatibility with some applications. | Pros: Can run applications that use ASP, .Net, Microsoft Access, or MSSQL databases and these are exclusive to Windows.  Software compatibility.  Cons: Less secure. | Pros: Popular, portable, compatibility, cheap.  Cons: immobile devices have better hardware capabilities. |
| **Client Side** | Significant expertise, time, and cost required. | Significant expertise and time required, minimal cost. | Minimum expertise and time required, moderate cost. | Moderate expertise, significant time, and low cost. |
| **Development Tools** | Popular IDEs for deploying software on Mac are AppCode and Xcode, which support languages like Swift, C, C++, Objective-C and Java.  Appcode does cost $199 a year for businesses and organizations. | Popular IDEs for deploying software on Linux are Visual Studio and Eclipse, which support Java, Python, PHP, Ruby, HTML, CSS, Javascript, and many more.  Eclipse and Visual studio are free, | Similar to Linux, popular IDEs for deploying software on Windows are Visual Studio and Eclipse, which support Java, Python, PHP, Ruby, HTML, CSS, Javascript, and many more.  Eclipse and Visual studio are free, | Popular IDEs for deploying on mobile devices are Visual Studio, Eclipse, and Xcode. Common languages used for mobile are Java, Javascript, and Swift. |

## 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**:

For the development of the web-based application Draw it or lose it, I would recommend Windows as the operating system. The main reason I would recommend windows as the operating platform for the development of this web application is because the company The Gaming Room has mentioned that they intend to release the game on multiple platforms. One of the many benefits of Windows is its compatibility. With more development tools available, releasing an application on multiple platforms will be smoother than without those tools.

1. **Operating Systems Architectures**:

The windows operating system architecture is a hybrid kernel, “which means it attempts to combine features and benefits of microkernel and monolithic kernel architectures”. A monolithic kernel architecture is “a traditional model of software program, which is build as a unified unit that is self-contained and independent from other applications” and kernel microkernel architecture is a “software or code which contains the required minimum amount of functions, data, and features to implement an operating system”. By having a hybrid kernel architecture, the windows operating system can benefit significantly resulting in an increase in performance.

1. **Storage Management**:

Effective storage management for a web-based application would benefit greatly from cloud-based storage management. With the use of cloud-based storage management, data storage can be allocated based on the growing application’s needs. Also, data can be easily accessed from multiple different platforms, with the use of cloud-based storage.

1. **Memory Management**:

Each process on 64-bit Windows has a virtual address space of 8 terabytes. The virtual address space is the range of memory and addresses that can be used. Virtual address space is used by Windows to quickly access memory that is larger than the available physical memory.

When developing software in one of the many integrated development environments that are compatible with Windows, it is important to use good coding practices, such as making sure that the program is only creating objects as they are needed and then disposing of them when they are no longer necessary, to assure that memory is not being wasted.

1. **Distributed Systems and Networks**:

Considering that the client would like the application to interact and be compatible with multiple platforms, developing it as a web-based application would be ideal. When done correctly, web-based applications should be able to operate identically on a mobile device like IOS or Android, as well as on Windows or Mac OS. With the use of web-based application development in an integrated development environment, such as Eclipse or Microsoft Azure, all functions and methods can be run on the server side, while the client side just deals with the user interface to send requests to the server. Because all methods/functions are being run on the server and data is being stored in the server, the experience will be identical regardless of what device the client is using. Software such as Microsoft Azure can be a very useful tool for building and hosting stable and secure web applications.

1. **Security**:

If the project proceeds with the implementation of cloud-based storage, then there will also be many security benefits concerning user data. Of the many cloud-based storage services to choose from, the top services provide extensive security measures, to ensure the protection of user data. In addition, data encryption can also be implemented when storing user data, to further enforce a secure environment for user data.

Works Cited

*Monolithic structure of operating system - javatpoint*. www.javatpoint.com. (n.d.). Retrieved April 10, 2023, from https://www.javatpoint.com/monolithic-structure-of-operating-system

Atlassian. (n.d.). *Microservices vs. monolithic architecture*. Atlassian. Retrieved April 10, 2023, from https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith#:~:text=A%20monolithic%20architecture%20is%20a,monolith%20architecture%20for%20software%20design.

GeeksforGeeks. (2022, February 1). *Windows memory management*. GeeksforGeeks. Retrieved April 10, 2023, from https://www.geeksforgeeks.org/windows-memory-managment/