

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 | 3/20/2022 | Naomi Ramsaroop | The Gaming Room wants to develop a web-based version of their current game, Draw It or Lose it which is currently only available on the Android App. |

## [Executive Summary](#_sbfa50wo7nsh)

**The Gaming Room** is looking to develop a web-based version of their game which is currently only available in the Android app store. The company wants their game, Draw It or Lose It, available on multiple platforms including Mac iOS, Windows, and Linx operating systems.

**Requirements:**

* Available on multiple operating systems
* The game will have the ability to have one or more teams playing at the same time with each team having multiple players.
* Streamline Development
* Train The Gaming Room staff on software and how to keep the program maintained and up to date.
* Provide the client with the best hardware requirements to operate the game at its best.

## [Design Constraints](#_2et92p0)

Based on The Gaming Room business and technical requirements, constraints the company will face are:

* Written and developed in Java and must be clear for even beginners on the “The Gaming Room” team can maintain, debug and review.
* Only one instance of the game can exist in memory at any given time.
* Multiple teams and players should be able to play at one time.
* The game must be able to check whether a name or team name has been chosen.
* Rendering of the image must be completed in 30 seconds
* The game consists of 4 One-Minute rounds.
* If the team does not guess the puzzle by the end of 1 minute, the other teams have 15 seconds to guess.

## [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)

In “The Gaming Room” UML Diagram below, you can clearly identify the Super and Subclasses of the program. The Superclass, also known as a parent class, is the Entity class. The subclasses, also known as child classes, are the Game, Team and Player classes. These classes have an inheritance relationship to the superclass, Entity. These classes will inherit the attributes “id” and “name” from the entity class, but they also have their own unique attributes.

GameService, Game, Team, and Player classes are all associations of each class. Players are created by the id and name attributes from the entity class. Then teams are created by creating a list of players that were created. The team then gets its own id and name through the attributes in the entity class. The game then takes the teams that were created and creates a list of teams for the game. The game then gets its own unique id and game name by calling the attributes from the entity class.

GameService is an association of the game class, but it does not call the attributes from the entity parent class. GameService has its own unique attributes such as “games”, “nextGameId”, “nextPlayerId”, nextTeamID” and “service” which is the singleton pattern to ensure only one instance of the GameService class can exist in memory at any given time. Using an iterator pattern in “addGame” and “getGame” attributes allows us to make sure game and team names be unique and allows the users to check whether a name is available or not.

**"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.**

## 

## [Evaluation](#_2o15spng8stw)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as a mobile device, 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 must work together.

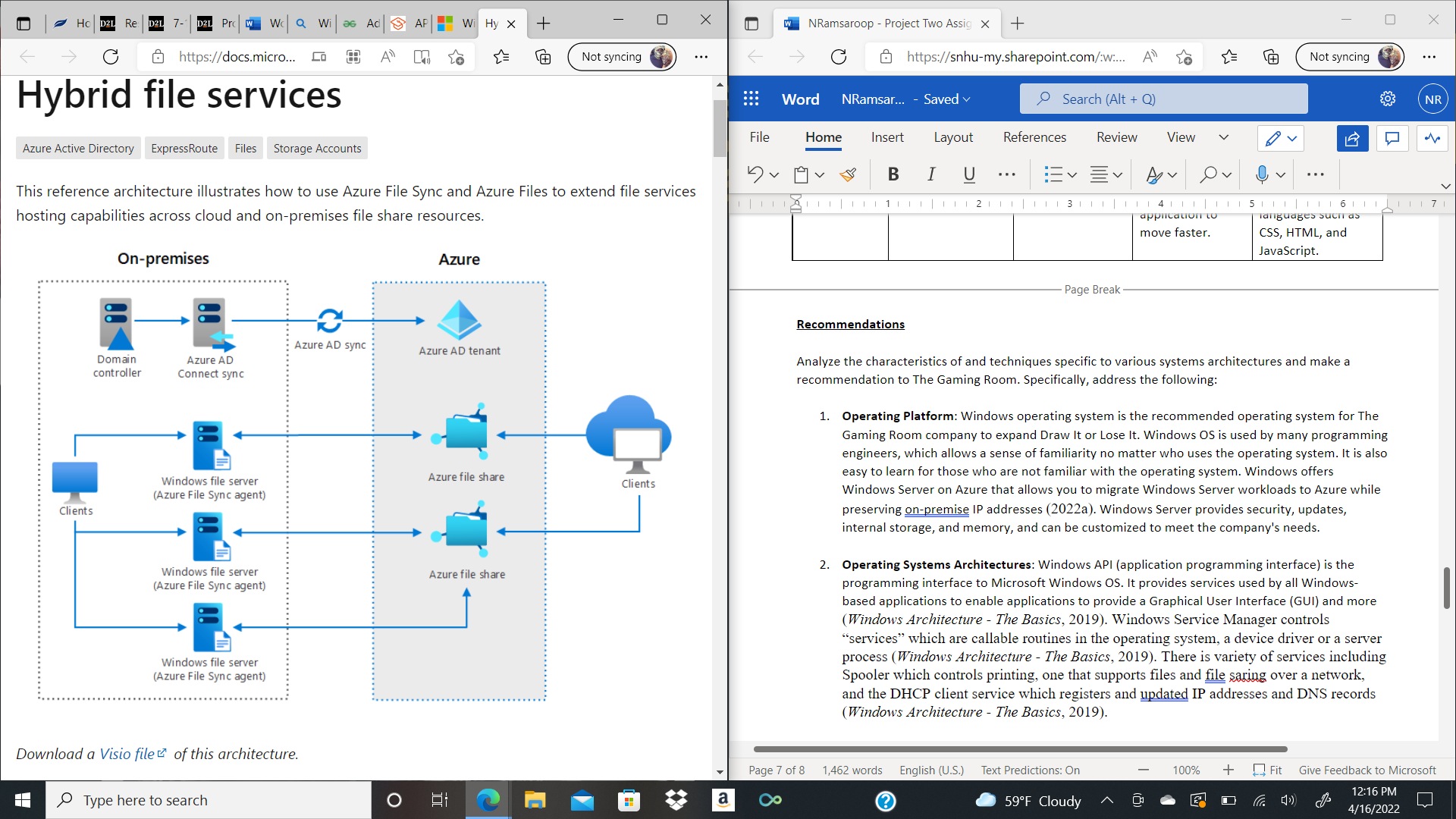
In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Simple and user-friendly GUI and their workflow is like their iOS for mobile applications. Seamless software-hardware integration.  MacOS server provides server functionality and system administration tools to manage both macOs – based computers and iOS-based devices. | Linux is an open sources operating system meaning most of the tools you will use on Linux are free.  Security on a Linux is a bit more customizable and easier to adjust compared to other operating systems.  There is a bit of a learning curve if you are not used to Linux servers but not too difficult. | Windows is consistent and flexible and gives the user the capability of hosting multiple websites on one tab.  Weaknesses include compatibility issues and security is average. | Several things but be considered when choosing a web-server framework for a mobile device, including a REST API and internet connection. |
| **Client Side** | Great for client-side development.  Simple to install, set up and manage for just $19.99  MacOS includes built in web service Apache for server web content. | Linux is typically cheaper because Linux is free compared to paying for other OS like macOS or Windows. | The cost of hosting is more expensive compared to other OS such as Linux because it is not a free-licensed software. | Creating, developing, deploying a mobile application, it’s important to ensure it can run on multiple phone operating systems such as iOS and Android.  Cost will depend on a wide variety of considerations, including if you will outsource development from a third-party company or develop and deploy in house. |
| **Development Tools** | MacOS Server provides many tools including the profile manager that simplifies deployment, configuration and managing mac computers and iOS devices. | Linux does support multiple languages including PHP, Perl, Ruby, and Python. | Windows uses .Net Framework, which is the perfect choice when creating web applications using .net, ASP and Visual Basic Language. It allows the website and application to move faster. | Web software applications for mobile devices are available in an online browser which is accessible with an internet connection. These applications are developed by using programming languages such as CSS, HTML, and JavaScript. |

## 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**: Windows operating system is the recommended operating system for The Gaming Room company to expand Draw It or Lose It. Windows OS is used by many programming engineers, which allows a sense of familiarity no matter who uses the operating system. It is also easy to learn for those who are not familiar with the operating system. Windows offers Windows Server on Azure that allows you to migrate Windows Server workloads to Azure while preserving on-premises IP addresses (2022a). Windows Server provides security, updates, internal storage, and memory, and can be customized to meet the company's needs.
2. **Operating Systems Architectures**: Windows API (application programming interface) is the programming interface to Microsoft Windows OS. It provides services used by all Windows-based applications to enable applications to provide a Graphical User Interface (GUI) and more (*Windows Architecture - The Basics*, 2019). Windows Service Manager controls “services” which are callable routines in the operating system, a device driver, or a server process (*Windows Architecture - The Basics*, 2019). There is a variety of services including Spooler which controls printing, one that supports files and file sharing over a network, and the DHCP client service which registers and updated IP addresses and DNS records (*Windows Architecture - The Basics*, 2019).  
     
   Below is a reference architecture from Microsoft.com which illustrates how to use Azure File Sync and Azure Files to extend file service hosting capacities across the cloud.



The Windows Server architecture includes storage accounts that is used to host file shares. Azure files is a serverless cloud file share that provides the cloud endpoint of a sync relationship

(2022a). Sync groups are logical groupings of Azure files shares and servers. An Azure File Sync agent is installed on the Windows server machine to enable and configure the sync with cloud endpoints.

Files are stored in the cloud in Azure file shares and can be used in two different ways including directly mounting serverless Azure file shares (SMB) or by caching Azure files shares on-premises using Azure File Sync (2022a).

1. **Storage Management**: Window computers come with their own internal storage, x64 architecture. Windows Server 2012 has its own storage management which is comprehensive and fully scriptable, and administrators can manage it remotely (2016). Management applications can use a single Windows API to manage different storage types (2016). Cloud Storage would be recommended and is included with Azure. Azure files are scalable which allows the company to only use what they need.
2. **Memory Management**: The memory management implements virtual memory and each 64-bit Microsoft window as its own virtual address space that enables addressing up to 8 gigs of memory (2021). Applications such as the Windows Task Manager, the Reliability and Performance Monitor, and the Process Explorer tool can be used to display and track the performance of the memory information of the system (2021).
3. **Distributed Systems and Networks**: Distribution systems has many benefits including scaling, low latency, and cost-effectiveness. Distributed systems allow systems to handle multiple requests and upgrades without downtime. These systems are more fault-tolerant than a single machine (Fawcett, 2020). Scaling allows the company to add more servers by either horizontal scaling or vertical scaling. Vertical scaling will add more power (CPU, RAM Storage, etc.) to your existing server (Fawcett, 2020). There is a difference between distributed systems and cloud computing. I recommend cloud computing because it uses network-hosted servers for storage, process, and data management plus, it doesn’t take up additional room in-house which will be cost-effective for the company. Priorities like load-balancing, replication, auto-scaling, and automated back-ups can be made easy with cloud computing and Azure makes it possible to create such systems (Fawcett, 2020).
4. **Security**: Azure File Sync topology increases security, along with security from Microsoft operating system and additional security features added on by the company. Azure File Sync can tunnel sync and file upload/download traffic over the Azure ExpressRoute or Azure virtual private network (VPN) (2022a). It would be required that networking features have service endpoints and private endpoints. Configure AFS to support proxy in the company environment and throttle network activity from Azure File Sync (2022a).  
     
   Azure File Sync also includes three different layers of encryption such as encryption at rest for data stored in Windows Server, encryption in transit between Azure File Sync agent and Azure, and encryption at rest for data stored in Aure file share. AFS can be backed up.

**References:**

*Windows Architecture - The Basics*. (2019, March 16). TECHCOMMUNITY.MICROSOFT.COM. Retrieved April 16, 2022, from <https://techcommunity.microsoft.com/t5/ask-the-performance-team/windows-architecture-the-basics/ba-p/372345>

E. (2022a). *Hybrid file services - Azure Architecture Center*. Microsoft Docs. Retrieved April 16, 2022, from <https://docs.microsoft.com/en-us/azure/architecture/hybrid/hybrid-file-services>

Fawcett, A. (2020, December 4). *What are distributed systems? A quick introduction*. Educative: Interactive Courses for Software Developers. Retrieved April 6, 2022, from <https://www.educative.io/blog/distributed-systems-considerations-tradeoffs>

A. (2016, September 6). *Storage Management Overview*. Microsoft Docs. Retrieved April 16, 2022, from <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh831751(v=ws.11)>

A. (2021, January 7). *Memory Management (Memory Management) - Win32 apps*. Microsoft Docs. Retrieved April 16, 2022, from https://docs.microsoft.com/en-us/windows/win32/memory/memory-management