

<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)4

[**Recommendations**](#_m8aleynsvzvc)7

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | <03/20/22> | <Tabitha Sinde> | <Executive summary, design constraints ,system architecture view, domain model > |

**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 implementing a web-based version of its already existing app Draw It or Lose It, which will render images from a large library of stock drawings as clues, rather than a player drawing images on an easel. The design constraints implemented involve both the technical and financial constraints in its development. My goal is to prepare a software design document and develop the gaming application

## [Design Constraints](#_2et92p0)

Financial constraints;

...As a developer, all the programming phases of development will have to follow a schedule in favor of client needs. I will have to strictly maintain the budget expectations to avoid unexpected expenditure.

...I should also be fully aware of all the web-based app development principles provided by my client to ensure quality, optimal user experience and heightened domain knowledge.

...I should also ensure I am aware of any known license requirements/ limits, legal compliance requirements and any purchase restrictions/ procedures in purchasing vendor projects.

Technical constraints;

…The main important technical consideration is to choose what language I will be using to build the app, in this case, Java programming language. This is because the client has its existing app running on an Android platform ,which is mainly supported by Java.

...The app design development will use Object Oriented design principles to help expose a lightweight API and ensure the app runs across multiple systems.

…This web-based app internal requirements should be cost-effective to outsource its hosting to the third parties. I should ensure to consider scalability ,manageability, feasibility and availability while building the app.

…The app should also be compatible with other leading web browsers that are similar in functionality and support especially Chrome.

## [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)

This domain model represents the business rules that are to be implemented through related classes and app entities. The four classes associated dare connected and depict a logical relationship . The GameService manages the games represented and presently played. The Singleton restricts creation of service to a single instance preventing occurrence of multiple instances. The Entity Class is a base class that interconnects to sub classes which share a name and a unique identifier. The Program Driver and Singleton tester share ana association that shows direct relationship with the arrow pointing from diver to tester. The driver creates object instances to facilitate implementation and then uses the tester to validate the service.

**"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 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** | This is an expensive platform whose software packages have comparable open-source Linux counterparts. Implementation technologies are more diverse hence fewer technical constraints, and its architecture enables typical functionability to be scaled with less performance impact.  It is however expensive since it has licensing costs and carries unnecessary features to host its application. | This is an open source with numerous libraries and pre-packaged utilities.  Its open-source application availability allows it to yield low technical constraints.  However, its file structure is more susceptible to outside sources and its modular architecture only exposes core functionality through the kernel | This is an expensive platform that carries both the OS and product licensing costs . It is more secure since it has the most restricted user access and has the controlled lock screen options that control outside users. However, its architecture is subject to performance issues since its layered architecture is heavyweight, and has a costly scalability | This is a framework that supports scalability and performance for hosting webs. It has many deployment tools that support major OS and still be independent of the OS. It however has additional relay steps hence requires connectivity to an internet enabled computer to support DNS conversion. |
| **Client Side** | It should be optimized, support modern browsers and ensure authentication, support a minimal number of file structures to limit media types and ensure cost effectiveness. | It should limit media type options that will be delivered to the client without additional overhead and support a minimal number of file structures. | Should be optimized to run with web browsers that impose technical constraints, | It should be strictly used to support lightweight API with no executable media types. It also requires expertise in developing UX across varying factors. |
| **Development Tools** | It has multi-threaded one-to-one architecture .IDEs and languages leverage vast libraries and support OS to promote scalability and concurrency ,which will still be independent of the OS, and eliminate licensing cost to provide robust solutions. | It has multi-threaded one-to-one model , its compiled languages help offset the overhead while the interrupted languages give additional runtime overhead. | It has one-to-one thread model, with its compiled languages offsetting the overhead and optimizes widows for its propriety | While android majorly supports Java , and expertise in particular frameworks are needed to support cross platform development |

## 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**:

The Gaming Room should have a serverless architecture and reduce complexibility and inefficiencies through single-purpose autonomous services and frameworks. It should also promote scalable and efficient solutions requiring less maintenance and encourage efficient design leveraging few dependencies hence decreasing dependency time.

1. **Operating Systems Architectures**:

Serverless Architectures build on the three tier solutions that provides logic and passes information between the user view and data store. The Linux OS contains core components delivering a lightweight solution, that occupies less disk space. The OS solution modules should also bypass more latent communication methods that are used with layered OS solutions.

1. **Storage Management**:

**The Serverless solutions integrates storage space into a resource pool that promotes greater virtualization space**. This kind of storage offers the fastest load times, and the indexed allocation eliminates wasted storage space caused by fragmentation under other allocation schemes.

1. **Memory Management**:

The CPU is majorly involved in memory management and needs to execute the app existing only partially in memory. The processes are resided on the disk ,broken into smaller segments that are pulled into physical memory only when needed for execution. The swap algorithm also influences the performance by ensuring low thrashing to avoid too much effort swapping between physical and virtual space.

1. **Distributed Systems and Networks**:

Serverless architecture runs across multiple servers to handle a large clientele demand. Through this load balancing the client needs are spread evenly across multiple servers improving responsiveness and availability. Database instances must also be synced continuously to appear as a single instance , which makes adding data to the database less frequent. The app should also support an API server to promote multi-platform.

6.Security:

Authentication and authorization are important in developing an app that is secure .This is done through role-based access control that promotes least privilege . Lightweight communication between clients and the server promotes stateless interaction that requires cached client-side identifiers, that are meant to be unique to the session. Linux will promote its security through independence, preventing a process running in one application to affect another.

PROJECT 2 SOFTWARE DESIGN TEMPLATE

Server Side:

The server side is majorly the systems that run the server, so they do not necessarily affect the client. The server side should majorly ensure advanced networking performance, high security for data stored, advanced backup capability and ability to update its hardware and software without requiring the system to reboot. Considering these general characteristics, the Linux server offers an open source and is free to access hence cost effective but has less support and being an open source makes it less secure for usage and storage of data. Windows server on the other hand is proprietary and expensive to use but has more software support/availability and is more secure. Mac also has flexible terminal commands to configure the server ,access or make changes to it.

Client side:

The client side is user focused that majorly creates the part of the website that the user can interact with, designs user interface, adds form validation and visual elements. Mac has moderate expertise and is time consuming and more expensive. Windows has minimum expertise and time required for operation, while Linux has the maximum expertise and time required but the minimum cost.

Development tools:

The most used development tools in programming include PyCharm, Eclipse, Visual Studio, GitHub, Notepad ++ and databases.

Different operating systems have different development tools and languages;

1. Mac -uses XCode as the development tool and has Swift as its programming language.
2. Windows- uses Visual Studio as its development tool and has Visual Basic as its programming language.
3. Linux – Eclipse is the main development tool and C is its default programming language.
4. Android – uses Android Studio as the development tool and mainly Java for programming.
5. iOS – has XCode as its development tool and Swift as its programming language.