

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

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

[Table of Contents](#bookmark1)**2**

[Document Revision History](#bookmark2)**2**

[Executive Summary](#bookmark3)**3**

[Requirements](#bookmark4)**3**

[Design Constraints](#bookmark5)**3**

[System Architecture View](#bookmark6)**3**

[Domain Model](#bookmark7)**3**

[Evaluation](#bookmark8)**4**

[Recommendations](#bookmark9)**5**

## [Document Revision History](#bookmark10)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 01/28/2024 | Tori Adams | A game that supports one or more teams.  Each team has X amount of players. Players guess the image that is being drawn before time is up in order to win round |

**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](#bookmark11)

<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.>

## Requirements

*Web Browser, mobile friendly?.*

## [Design Constraints](#bookmark12)

#### *What is the Budget for web based application? Anything served in-house or divided to external party/resource? How does the company plan to make money? Is the game free? Paid ads?*

## [System Architecture View](#bookmark13)

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](#bookmark14)

In the diagram, the game and game service are associated with each other because the gameService provides the various amount of games that be played. The amount could scale larger or decrease over time. The gameService class will hold attributes related to that current game. There can only be one player on one team at a time, and one team can play one dame at a time. The ID of the player, team, and game are used to validate that there is only one entity of this attributes involved in the instance of a new game.

**"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](#bookmark15)

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** | -Server OS is not highly used in comparison to linux ow windows.  -All hardware is through apple | -Open source so can manipulate it however one wants  -Free to  -Known to be most secure and stable  - | -Used and known by most PC users-monthly cost -VPs -dedicated hosting | -Android uses linux kernal and Apple uses Its |
| **Client Side** | -Most expensive, but quality hardware  -All hardware is through apple so maintenance is one consistent -Supports Various Browsers  -Does not Support Edge | -Free to use  -Learning curve  -Supports Chrome -Supports Edge | -Free and paid Tiers  -Well known and used -Does not support Safari | -Consider page and feature responsiveness that are available on mobile vs desktop |
| **Development Tools** | -Can only develop Mac OS products so programming languages specific to Mac like swift  -Preinstalled package managers  -Swift developer | -Preinstalled package managers  -C -Python  -Java | -Cannot build Mac products using Windows -Java | -Build software natively that supports web browsers -Java 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**: Linux because it is open source. This allows more control to tailor the game in more specific ways. It is also free.
2. **Operating Systems Architectures**: Linux support cross platform frameworks in order to support different OS other than Linux. It also supports containerization and virtualization which is helpful with controlling and managing the resources from the game that will be used.
3. **Storage Management**: LVM(volume management) can be used. It allows volumes to be reduced and extended easily. They can be used if there if not partition space. LVM also supports data backup and failing disks. There is a variety of filesystems available for linux which allows different ways to arrange the files on the drive which is important to have control over with the gaming software.
4. **Memory Management**: Because virtual memory exists with Linux, we can support multiple applications simultaneously. The kernel is what is in charge of this. Two processes can have the same virtual address, but different physical RAM. Another note is caching which the page cache can which caches data from disk memory which speeds up access times to the data the game needs to retrieve. This is an important piece that helps with I/O performance.
5. **Distributed Systems and Networks**: Benefits of a distributed system gives users to the resources maintained by the system which affects speed, availability, and reliability. The server, clients and resources could be in different locations and they communicate on a network. If one are fails, it does not affect the entirety of the system. You can also have redundancy in place in order to make sure nothing completely fails. Ways users can gain access is through remote options if one resources is down. If a site is no longer available, the entire site can be migrated to a different location and still be accessed.
6. **Security**: Linux is built with security in mind with developers would wide consistently patching any issues. It is also known to be able to handle heavy workloads without crashing. Linux divides kernel space and address space. Important process can only access kernel space. The segregations helps so that the hackers can’t pollute the machine and crash the system with the kernel processes.