

<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 | <mm/dd/yy> | <Your-Name> | <Brief description of changes in this revision> |

**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 purpose of this project is to set up the environment for The Gaming Room to facilitate the development of the web-based version of the game. The game is a image puzzle recognition game.

## Requirements

* A game must 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 using unique identifiers for each instance of the game, team, or player.

## [Design Constraints](#_2et92p0)

The game must be able to run in a various web-based distributed environment so that it is cross compatible with different browsers. The game must be able to display images and set timers for the players and the time to render images, this is integral to how to game is played. The game will be built in Java, this is important because it defines what type of developer expertise is needed to complete the game.

## [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)

Program Driver contains the main but uses the singleton tester class to test the program.

Entity is the parent class to Game, Team, and player.

Game can contain one or more teams and teams can contain one or more players.

Game Service contains game and

**"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** | While Mac OS is capable of running all types of browsers it could possibly have some proprietary software that would make it more expensive. Also, mac hardware tends to be less configurable and be more expensive than the competition. | Linux would most likely be the cheapest option due to it being open source. There is also a large amount of documentation in the open source community due to its popularity which can help ease the project along. | Windows is also very popular and Microsoft does have a fairly popular variety of server management applications. | A mobile device would not be ideal to host a web server because it would lack the physical hardware to handle a constant connection and memory for it. |
| **Client Side** | If the software is made as an application for mac, it will require its own separate development process. If it is used withing a browser then a mac can easily download most web browsers. | Using Linux in the form of an application would be possible but would require its own development process. If its in a browser then Linux Is a bit more limited in default browsers and can be difficult to get the right version. | Using a windows machine would also require their own development team however windows does have good access to most web browsers. | Using a phone would require a whole different type of input while also developing the application specifically for android or IOS. While phones are currently pretty good for games it is a different experience for the user entirely. |
| **Development Tools** | Mac has access to most development tools. For example, it is possible to install windows software such as visual studio. This however doesn’t work in the opposite situation where you can’t install Xcode on a windows pc. | Linux has access to a large variety of development tools however I do think it has a more limited variety in comparison to mac and windows. | I believe windows has the largest variety of development tools with the exception of apple development software. | Developing on a mobile device would be difficult due to there not being very many simple solutions to developing a testing code on a mobile device. |

## 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**: Chrome Web Browser.
2. **Operating Systems Architectures**: I believe the best option for operating platform would be to build it to run as a web app within the Google Chrome Browser. The reasoning is that the default browser for windows is Microsoft Edge, which is actually based off of a chrome browser therefor it can run the same web apps. This would also allow users to use any platform to access the web application due to the free availability of Chrome and Edge which can also be accessed on mobile devices.
3. **Storage Management**: I believe the best option would be to build a Linux server.
4. **Memory Management**: The reason I chose to make it a Linux server is primarily for how much documentation and how popular this option already is as a server. Also due to Linux being free it allows for the best value.
5. **Distributed Systems and Networks**: I think building it as a web application would make it the most easily distributed software that could have cross platform compatibility. This cross-platform compatibility does however come at the cost of the application being run by multiple users at the same time on the same server therefor needing to have multiple servers will have to be an option. Another issue that would arise would be the difference in how input in done between mobile devices and other operating platforms.
6. **Security**: By building it entirely in the server and letting it run as a web application we can ideally keep the least amount of information regarding the users because the application is not installed on the platform, it is just accessed by the browser. We can also have login credentials to be necessary to play the game and that would increase our security.