

The Gaming Room

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_heading=h.gjdgxs)

[**Table of Contents 2**](#_heading=h.30j0zll)

[**Document Revision History 2**](#_heading=h.3znysh7)

[**Executive Summary 3**](#_heading=h.2et92p0)

[**Requirements 3**](#_heading=h.tyjcwt)

[**Design Constraints 3**](#_heading=h.1t3h5sf)

[**System Architecture View 3**](#_heading=h.4d34og8)

[**Domain Model 3**](#_heading=h.2s8eyo1)

[**Evaluation 4**](#_heading=h.17dp8vu)

[**Recommendations 5**](#_heading=h.26in1rg)

## [Document Revision History](#_heading=h.lnxbz9)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 3.0 | 08/9/2023 | Tuyet | Modified Recomendations |

## [Executive Summary](#_heading=h.35nkun2)

*The company wants to create a program that can run on multiple platforms, the game should be similar to the “Draw it or Lose it” television game in the 80s. The game consists of 4 rounds and each round lasts 1 minute the team has 30 seconds to guess a randomized picture from a database, if the timer runs out the opposing team has 15 seconds to guess the picture.*

## Requirements

*The game has to run on multiple platforms*

*They have to allow one or more teams to participate.*

*Each team will have more than one player.*

*Game and Team names must be unique, the system notifies the user if the name is available or used.*

*Only one instance of the game can exist at any time.*

## [Design Constraints](#_heading=h.1ksv4uv)

*The first requirement required the program to be run on all the operation systems, the system might need to be written in more than one language and software to accommodate multiple platforms as each platform can run certain types of coding language(s). The program could translate the existing program to different types of language to make it possible to run without remaking the whole process, meanwhile, that will keep the program consistent and standardized.*

## [System Architecture View](#_heading=h.44sinio)

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](#_heading=h.2jxsxqh)

*The Team, Game, and Player classes inherit from Entity superclass, they all have a relationship to one another. The Game, Team, and Player classes are sharing the same “id” and “name”. The Game class inherits from the Entity superclass and has a relationship with the Team class and GameService class. Team class has a relationship with the Player and Game class and the Player class has one relationship with The Team class.*

**"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](#_heading=h.z337ya)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Characteristic: flexibility in commanding the server and making changes. Mac OS is a series of graphical user interfaces and simple interfaces. The platform does offer server-based operations and it can be hosted at Apache, Caddy, Eclipse, Nginx, MySQL, and more. Many parts of MacOS is an open source but the GUI is closed source, and since some part of the system has copyrights, licensing fee might required to be made to develop a server-based.  Advantages: MacOS has many hosting capabilities, is quick and responsive, and can be easily upgraded.  Disadvantages: Not as popular as the others to host web servers. | Characteristic:  is a multitasking operating system and is highly memory protected. Linux is an open-source system that allows people to change, analyze, and redistribute the source code. The potential licensing costs are minimal to none. The platform does offer server-based operation and it can be hosted at Apache, Nginx, Caddy, Eclipse, and more.  Advantages: highly secure system, and one of the popular systems to host the web.  Disadvantages: the limited application that would host a web need. | Characteristic: The developer has full control of the application and can restrict it to other users, due to its simplicity to use the system became popular with many professionals as well as amateurs. The system has multitasking as well as a series of graphical user interfaces. The server can be also hosted at Apache, Microsoft IIS, Ngnix, Caddy, and more. Similar to MacOS the system also required fees for their licensing.  Advantages: fast loading time, familiarity for most users, and lots of resources.  Disadvantages: Prone to hacking and viruses infiltration and have required high system requirements. | Characteristic: Most popular among users, could get access almost anywhere, it allows smartphones, tablets, and other portable devices to run applications and programs. The system includes iPhone's OS, Apple iOS, and Google's OS, Google Android. Cross-platform applications are often used to accommodate multiple platforms of mobile OS. The cost is minimal to none and it would be difficult to host  Advantages: cheaper than other OS and has more users.  Disadvantages: It is highly selective to various smart mobile devices and has bad security |
| **Client Side** | Cost: Same cost as Windows OS.  Time: Decent time is required.  Expertise: Decent experience is required.  The application development process required for the system to run smoothly would be that the website should be able to run on various types of screen sizes on Mac Book, Mac Mini, iMac, and other versions of MacOS devices and MacOS web browsers like Safari, Chromium-based and other downloadable browsers. | Cost: Affordable and budget-friendly.  Time: very time-consuming.  Expertise: requires high knowledge and skills.  The application development process required for the system to run smoothly would be that the website should be able to run on various types of screen sizes on Linux-installed devices and Linux web browsers like Firefox, Chromium-based, and other downloadable browsers. | Cost: Same cost as MacOS  Time: Decent time required.  Expertise: Decent experience required.  The application development process required for the system to run smoothly would be that the website should be able to run on various types of screen sizes on Windows-installed devices and Windows web browsers like Internet Explorer, Chromium-based, and other downloadable browsers. | Cost: budget-friendly  Time: Decent amount of time is required.  Expertise: Decent experience is required.  The application development process required for the system to run smoothly would be that the website should be able to run on various types of screen sizes on all the mobile devices like all different versions of iPad and iPhone for IOS and different versions of Android mobile devices. |
| **Development Tools** | MacOS 'most popular language is Swift but it can also support HTML, CSS, JavaScript, Java, Python, PHP, and Ruby.  The IDE that could run on MacOS are Eclipse, Netbeans, and Visual Studio Visual are popular among developers. The IDE allows multiple developers to work on their parts simultaneously. Different teams of developers might be required if the customer wishes to create a system in specialized devices or platforms.  Licensing fees might require for different pay service IDE. But Netbeans, Eclipse, and Visual Studio are open sources, therefore at no cost. | Linux can work with visual studio, eclipse, along with notepad++.  The OS can support HTML, CSS, JavaScript, Java, Python, PHP, and Ruby.  The IDE that could run on MacOS are Eclipse, Netbeans, and Visual Studio Visual are popular among developers. The IDE allows multiple developers to work on their parts simultaneously. Different teams of developers might be required if the customer wishes to create a system in specialized devices or platforms.  Licensing fees might require for different pay service IDE. But Netbeans, Eclipse, and Visual Studio are open sources, therefore at no cost. | Easier to use than Linux but can run the same as it. So visual studio, eclipse to name a few of the many languages. The OS can support HTML, CSS, JavaScript, Java, Python, PHP, and Ruby.  The IDE that could run on MacOS are Eclipse, Netbeans, and Visual Studio Visual are popular among developers. The IDE allows multiple developers to work on their parts simultaneously. Different teams of developers might be required if the customer wishes to create a system in specialized devices or platforms.  Licensing fees might require for different pay service IDE. But Netbeans, Eclipse, and Visual Studio are open sources, therefore not at cost. | You can create countless apps using Android and Swift. Both languages and software can be run on all three machines.  The OS can support HTML, CSS, JavaScript, Java, Python, PHP, and Ruby. |

## Recommendations

1. **Operating Platform**: *The Linux operating system will be a popular choice to host a website, the system is secure, affordable, offer a wide range of developing tools to create websites, and is highly customizable. The program already existed for Android using that as a template which could reduce the time and tackles one of the Linux disadvantages.*
2. **Operating Systems Architectures**: *The architecture of Linux compose of multiple components, it starts with the very base where the hardware resides, this area contains physical devices that support the system like RAM, CPU, hard drives, motherboards, graphics cards, and more. Linux systems have kernel space, the space is the core component of the operating system. It is responsible to communicate between hardware and software, files managements, processing information, memory, user access control, and resource management. User space is where the user can interact with the system by providing commands via inputs, user can see applications, databases, and files that contain within the system. The user space and kernel space memories are separated to provide more stable and reliable system functions, this ensures if the userspace fails the kernel space could provide a backup. By access to the kernel space from user space the user can access it through the shell, applications, and system libraries.*
3. **Storage Management**:  *One appropriate storage management system to be used with the recommended operating platform is the use SSD, which manages information that is stored in the SSD properly will help the system run smoothly. Storage management phases include allocations, recovery, compact, and reuse. The process is usually run within the Kernel space of the system, where the command is to store, retrieve, recover, and more to the system. The Gaming room required various pictures in the library, by using SSD it could render the data more efficiently compared to the traditional storage.*
4. **Memory Management**: *The game is required to store the game and team names so each game and team can be unique. A file of pictures is also required to be used during the game, the file can be stored externally from the IDE, and imported and used during the program process. Linux operating system is ideal for memory management, the memory can be manipulated by the paging process, the process helps access memory quicker.*
5. **Distributed Systems and Networks**: *There are many software on the market that allows the program to operate on multiple platforms, there is Codename One a cross-platform framework, that is simple to use and can operate on iOS, Android, Desktop & Web apps. To prevent the outage we need to discuss the limitation and backup in case the current system can not handle it. Lastly the networks and system connectivity by using REST API, the software architecture styles support the system by maintaining the performance even system is increased in complexity, it can also be scaled up, as being stateless make it simple to modify and it’s standardized amongst developers.*
6. **Security**: *To keep the system secure a good firewall is required for all the system devices, access control by authorizing users privilege and command the system to authenticate users when they request access. Secured communication between devices is also required to be implemented by encrypting messages using SSL and TLS. A standard system and employee policies also need to be developed properly, the system also required routine maintenance, update, and backup when needed.*