

<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 | <11/05/2022> | <Nathon Chavez> | <Updated Recommendations> |
| 2.0 | <11/18/2022> | <Nathon Chavez> | <Added in module 5 reqs> |

**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 looking to develop a web-based application. Although the application is exclusively on Android, they want to optimize the game so that it runs smoothly on all platforms. The game is a lose interpretation of the 80s game *Wind, Lose or Draw*, in which teams compete to guess what is being drawn. Currently, The Gaming Room does not know how to set up the environment, therefore the client has asked to create a design document and begin working on the application. >

## [Design Constraints](#_2et92p0)

< For the constraints and requirements, the application needs each team to have multiple people, game and team names must be unique, needs one or more teams in the game, and is optimized for various platforms. These requirements, for the most part, are mostly simplistic in that Java is a universal language and can run on all the platforms we are targeting. Given that we are developing a game for multiple platforms, we will also need to optimize the UI for each platform since screen sizes and other general hardware varies per platform. Additionally, we will also need unique IDs for games and team names, which can be accomplished through iterators and memory allocation. >

## [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)

<Below, we can see that all the classes connect or rather all lead to the entity class. We prove this later in the code by extending the Game, Team and Player classes to the Entity class. In other words, the arrow indicates that all the subclasses will inherit the attributes and methods from the superclass or parent class. Although we will not be doing anything this week with the ProgramDriver, we can see that it is pointing towards the SingletonTester to test the code. The *uses* arrow indicates that we will need the SingletonTester to test the restrictions we have for the game. The GameService class is the backbone of the application as it has the most methods and objects, >

**"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)

To ensure that the application is optimized for various web browsers we would need to know what browsers we want the app to be optimized for and test for those web browsers. The testing can be done either via emulators, virtual machines, or by using up-to-date software requirements like ES-6,7 or 8. What may be difficult is figuring out what plugins, dependencies, and libraries may or may not be compatible with the browsers we optimize the app for. For example, many features of Internet Explorer is not compatible with Bootstrap 5 because that library was not optimized with Internet Explorer in mind and IE was made on a much older platform. Additionally, it may put a strain on the development team because there may be some libraries or languages that some members of the development team may not be familiar with. There is also the chance of the requirement to use libraries that cost money because a certain library has a feature or gimmick that the client wants or needs. This, in turn, has the potential to further divide the development team because not everyone have may experience working on such a niche library.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | <Terminal commands that allow user to make changes, server access, or general configurations> | <Flexible general server configurations> | <Proprietary software, but comes preinstalled in most new mobile (laptop) and desktop devices> | <Game web applications are somewhat difficult to optimize on mobile devices due to their smaller screen size and the optimization of the browser used. However, groups such as Oracle are server-side implementation which can manage applications, servers, users and smaller or large platforms.> |
| **Client Side** | <Due to the large userbase and strong support there are many browsers that will help ease development due to their many development tools. It should take a medium amount of development time However, to develop for MacOS we do need an Apple device.> | <Large variety of well-supported and trusted development environments and tools with minimal cost and development time.> | <Similar to Linux, Windows has a wide variety of tools and supported development environments. Other than MacOS there is intuitive cross-development development with fast development and deployment.> | <It will be difficult to test and deploy on some browsers due to the smaller screen size and varied hardware. It will also take longer to develop and will require more knowledgeable developers.> |
| **Development Tools** | <We can run Linux and Windows through virtual environments and develop through there. However for applications, we need to be reviewed and approved by Apple. > | <Virtual machines can run Windows and MacOS. We also have Eclipse for Java and can deploy at any time. Also no license is needed and it is focused around being open-sourced.> | <We can deploy at any time and also has Eclipse for Java as well as VS Code. It can also run Linux through a virtual machine and can develop mobile applications with no extra cost.> | <Need to know Swift or just base Java for universal programming. Need approval from Apple for deployment and Xcode.> |

## 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**: <After heavy consideration, I decided the best course of action is to utilize Windows as the preferred OS for development and deployment. Windows has option of using virtual machine to test for MacOS, Linux and mobile systems. Additionally, there is large support and is compatible with many IDEs as emulators.>
2. **Operating Systems Architectures**: <Windows has the base and more intuitive user mode which is meant for the base user to maneuver and manage; it is where the user can browse and do their daily or recreational tasks. However, there is also the kernel options which provides the options of low-level operations such as scheduling threads, or routing memory or hardware. >
3. **Storage Management**: <Regarding storage, my recommendation would be to choose a cloud-based storage. With a cloud-based storage, we do not need to worry too heavily on local hardware, especially since hardware prices are currently higher than usual. Therefore, I would recommend using something Google Cloud Services or Microsoft Azure. The benefits of using Microsoft Azure comes with the fact that we are already using Windows to develop and deploy the applications. So, compatibility comes with the cloud service. >
4. **Memory Management**: <Since I am recommending Windows as the OS of choice, we will also need to use the latest version of Windows 11. However, the biggest drawback about using Windows 11 is that it is still a new operating system, so there are still many bugs, stutters, and is not yet as efficient as Windows 10 especially when it comes to development. However, it is constantly getting updates to fix the software and improve the general stability of the OS. With Windows, we have the option of using a virtual machine memory address space, but we could also focus on getting more SSDs for space to save our builds and RAM for faster processing. >
5. **Distributed Systems and Networks**: < The reason I recommended using a cloud service is due to their reliability and general stability when it comes to the ease of access of the projects and builds. For example, cloud based services offer email alerts, general service monitoring and safety features as well to prevent any potential attacks. >
6. **Security**: <When it comes to cloud services like Azure or Google, we do not need to worry about potential security breaches. These services offer data analysis and security monitoring so that developers and analyst can see what the user base is doing and find potential issues that may grow to become something more significant and tackle that issue before it becomes a bigger problem. >