

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 3.0

## 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 | 05/29/2022 | Courtney Maxwell | Initial draft – updated table and information in brackets |
| 2.0 | 06/06/2022 | Courtney Maxwell | Update for Project Two |
| 3.0 | 06/21/2022 | Courtney Maxwell | Update for Project Three |

**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 wants to create a web-based game called “Draw It or Lose It” that can be used on multiple platforms. Currently, the game is only on Android devices.

## [Design Constraints](#_2et92p0)

* Requires more than one team to play
* Each team must have multiple players
* Team names must be unique
* Only one instance of the game can exist at any time
* Must be able to run on multiple platforms

## [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)

Entity creates a relationship between Game, Team, and Player class. Each class inherits information from Entity. The UML below displays this inheritance. Each class shares common references like “id” and “name” as displayed in the diagram. GameService has a reference of Games, Games has a reference of Team, and Team has a reference of Player.

**"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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Characteristics:  Commonly used for web hosting  Advantages:  Upgradeable  Several options for each web hosting requirement  Weaknesses:  Not as widely used as other operating systems for web hosting services | Characteristics:  Most preferred due to security  Open source  Advantages:  Security-most bugs are caught before any damage is done  Most preferred for web hosting  More cost-friendly than Mac  Free  Weaknesses:  Not as widely used so it is more difficult to find applications to support needs of web hosting | Characteristics:  Dominant to other operating systems  Closed platform  Proprietary  Advantages:  Doesn’t take much time to load  Widely used  Weaknesses:  Security against viruses is not as good  Poor tech support  Expensive | Characteristics:  More popular  Easily portable  Advantages:  Wider reach of service  More compatibility  More cost effective  Weaknesses:  Poor security  Depends on the mobile device |
| **Client Side** | Cost:  Similar cost as Windows  Expensive  Time:  Moderate time required  Expertise:  Moderate expertise required | Cost:  Minimum cost  Time:  Maximum time required  Expertise:  Maximum expertise required | Cost:  Similar cost as Mac  Expensive  Time:  Minimum time required  Expertise:  Minimum expertise required | Mobile devices allow flexibility to clients  Can see updates anywhere  More difficult to implement than other platforms |
| **Development Tools** | Macs use Swift most often with Notepad++ but all languages can be run on Macs. HTML, CSS, JavaScript are used to support Java, Python, PHP, and others.  Objective-C-programming language | Visual Studio, Eclipse, Notepad++ are all used.  HTML, CSS, and JavaScript are used to support Java, Python, among others.  Visual Basic – programming language | Visual Studio, Eclipse, HTML, CSS, JavaScript are used to support Java, Python, PHP, and other languages.  Notepad++ is also a useful tool.  C++ - programming language | An infinite number of apps can be created for mobile devices. Languages and software are run on mobile devices and consist of HTML, CSS, and JavaScript to support Java, Python, PHP, and more. |

Desktop apps provide a better experience for the user and are more responsive to downloads, updates, and other services that can slow the app down. Compared to a web app, a desktop app must be downloaded before it can be played or used making desktop apps less ideal than web apps. Web applications are easy to update for the developer and the user. Web apps are less responsive, especially when they rely on data from the server. Performance can vary because one cannot control the Internet. Reliability on web apps depends on the status of the Internet connection.

## 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**: Because Windows is more widely used, I would recommend starting with devices that support Windows. The software used is more available and requires minimum expertise and costs.
2. **Operating Systems Architectures**: Using Windows allows one to access several Windows applications that show a GUI, or Graphical User Interface. The applications access system resources, graphics, messaging, file exploring, web services, and more. User accounts and specific servers are used for the services.
3. **Storage Management**: Windows allows one to view and manage files on your hard drive as well as how much space you have in storage. You can also choose what file locations you want to save your apps in. You still have access to the cloud to save data. Because the storage is built-in to the system, creation and placement for larger projects is easier preventing it from getting lost or deleted.
4. **Memory Management**: Games require some form of memory management to store pictures, files, programs, and more. A database for memory management allows one to store large projects in one secure location.

1. **Distributed Systems and Networks**:
2. **Security**: Windows has built-in security software that detects malware, viruses, and security threats in real time. As threats change, the system updates to keep it and the user information safe.