

CS 230 Project Software Design Template

Project 3



Connor Corson

2/18/22

Southern new hampshire university



Creative Technology Solutions

# **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 | 1/21/22 | Connor Corson | Summary, constraints, architecture, model, eval, recommendations have been edited and revised |

**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)

Creative Technology Solutions has asked that I build a web-based game that serves multiple platforms based on their current game, Draw It or Lose it. They require that the game can involve numerous teams, multiple players to each team, unique name requirements, and only one instance of the game to be established at once. This could get tricky when developing the game on numerous OS’s. Mac is different from Windows, and that could cause error. However, using cross-platform software, we can make it happen!

## [Design Constraints](#_2et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

There are a few constraints that can be spotted right away. Issue with time constraint can be a factor depending on the due date. The transportation of the game to numerous platforms can cause quite a problem. Without the necessary skills in cross-platform programming, this project would almost be impossible. And we must include how the game’s graphic properties may vary depending on the OS that it is currently running on.

## [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)

**"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.**

So to my understanding, Game, Team, and Player are all “Entity.” They are kind of in a “is a” relationship with Entity. This is called inheritance. Player and team is an Aggregation, meaning one has a reference of an instance of another class. This would also be an example of GamesService and Game, and Team and GameService. This relationship is like each class “has” each other in themselves.

## [Evaluation](#_2o15spng8stw)

**Server Side:** The client has asked you to create a web-based application. This implies a server-style configuration for hosting the website and allowing it to scale up to thousands of players. What does this mean for your ability to host the software application on each operating platform listed above?

* **Evaluate various platforms for their characteristics, advantages, and weaknesses for hosting a web-based software application.** Consider the following in your evaluation and articulate your findings in the software design template:
  + Does each of the operating platforms offer a server-based deployment method where the website will be hosted?
  + What are the potential licensing costs to the client, The Gaming Room, for the server operating system?

**Client Side:** The client wishes to move beyond their current Android-only application to supporting players on iOS and Android mobile platforms, as well as traditional desktop-based operating systems. The application must be delivered as a modern, responsive HTML interface running inside the web browser for desktop clients (Linux, Mac, Windows), as well as on mobile platforms. Each will be capable of communicating with the back-end web application running on the server.

* **Determine the software development considerations (cost, time, expertise) that are necessary for supporting multiple types of clients.** Consider the following in your evaluation and articulate your findings in the software design template:
  + What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices?

**Development Tools**

* **Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deploying on each operating platform.** Consider the following and articulate your findings in the software design template:
  + What impact do these technical requirements have on a development team? Consider whether multiple development teams may be needed.
  + Are there licensing costs related to the development tools?

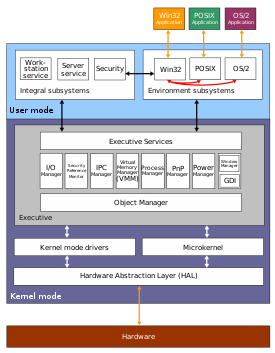
| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | * Advantages: upgradable, It is reliable and able to run the same software locally, comparable to Windows and Linux with commands and OS, very good security, and safer to use, web and email hosting, Jenkins and Xcode Server Integration * Disadvantages: ***Moderate Cost,*** only compatible with apple Hardware, not open source * OS has flexible terminal commands, can easily make changes, popular in webhosting. | * From what I have read, Linux and Windows have been battling for the top seed in Server-based solution. * Advantages: **Often cheaper**, most preferred, system administration gives any admins major benefits, Cyber safe, rare security errors, few demands on hardware, integrated remote function. * Disadvantages: complex operation, 3rd party programs installed only by admins, update process can be complicated, long-term support not offered to all versions, tons of pro programs do not work with Linux. * .NET servers .Linux comes with 6 deployment steps that are necessary for a server to be started, and have their own commands: 1) installed “yum” 2) configured “vim” 3) started “start” 4)persistent “ststemctl” 5)tested 6) reconfigured “restart” | * From what I have read, Linux and Windows have been battling for the top seed in Server-based solution. * Advantages: Beginner friendly, up to date drivers and quickly and easily available, supports a ton of different 3rd party apps, tech problems fixed easily with a system recovery, long term support is 100% guaranteed, very compatible with various programs * Disadvantages: **High license cost,** security errors are common, malware vulnerability, takes up a ton of resources, not good for multi-user systems * Windows offers the Microsoft Application Virtualization Management, which is a client that lets users interact with applications | * Advantages: Reliability when existing on interconnected machines, remote management for devices, you can choose what you use for space, processing power, and security, different OS like IOS and Android, data backups, mobile, **cost-efficient** * Disadvantages: system integration, stability, very selective depending on the phone, bad security * From my reading, I honestly cannot say that mobile devices have a solid and secure server-based deployment method. I cannot find anything that proves me incorrect. |
| **Client Side** | Cost: Moderate  Expertise: Moderate  Time: Moderate  I would say the development process would require a ton of testing on different OS. This testing will ensure that everything works, throughout every OS in question. Cross-platform apps will need a coding language that supports cross-platform development like Java, C++, C#, Ruby, etc. | Cost: Low  Expertise: Maximum  Time: Maximum  I would say the development process would require a ton of testing on different OS. This testing will ensure that everything works, throughout every OS in question. Cross-platform apps will need a coding language that supports cross-platform development like Java, C++, C#, Ruby, etc. | Cost: Moderate  Expertise:Minumum  Time: Minimum  I would say the development process would require a ton of testing on different OS. This testing will ensure that everything works, throughout every OS in question. Cross-platform apps will need a coding language that supports cross-platform development like Java, C++, C#, Ruby, etc. | Cost: Low  Expertise: Maximum  Time: Maximum  I would say the development process would require a ton of testing on different OS. This testing will ensure that everything works, throughout every OS in question. Cross-platform apps will need a coding language that supports cross-platform development like Java, C++, C#, Ruby, etc. |
| **Development Tools** | Languages: HTML, CSS, JavaScript, Java, Python, PHP, Ruby  Tools: Notepad++  Safari  Free to use.  With correct training and understanding on these languages, it will give the development team less of a problem. Both safari and Notepad are very easy to guide through and will make it an easy project for the development team. | Languages: HTML, CSS, JavaScript, Python, PHP, Ruby  Tools: VS, Eclipse, Notepad++  Free to use  With correct training and understanding on these languages, it will give the development team less of a problem. VS and Eclipse may be a challenge to the development team at first. They are both great tools, but have a lot packed into them, that they will need to explore and learn as they go on. | Languages: HTML, CSS, JavaScript, Python, PHP, Ruby  Tools: VS, Eclipse, Notepad++  Free to use  Windows will be the easiest to guide through, just slightly easier than Linux. Same goes for the tools. | Languages: HTML, CSS, JavaScript, Python, PHP, Ruby  Tools: android, swift,mimo, udemy  Some subscriptions may apply  With mobile being a whole different os than the rest, it will come as a challenge. However, with a phone, there are several apps and programs that can help guide and teach the development team. |

## 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**: To compare the four operating systems, they all have pros and cons. Each platform has their own unique way of the use of games, the operators, and how the games function. I however, would recommend Windows to be the most ideal operating server platform for Draw It or Lose it. After some research, I have concluded that the cost for the OS is moderately fair, it does not take a strong amount of expertise to figure out the OS, and it is time friendly. Windows has a lot of different languages that can support communication through different OS, like Java, C++, C#, ect. I say this because many other OS use these coding languages. Windows has a great deal of security features to keep a server safe and secure, has the capability to be compatible with other software, development tools, and the OS in question, and actually has the most software available to make sure the game is at top care. Its only disadvantages include support, and can be vulnerable to malware attack, if not careful. I believe with using Windows OS, it will be the easiest way to expand our game to other environments.
2. **Operating Systems Architectures**:

“Operating systems architecture refers to the overall design of hardware and software components and their operational effectiveness as a whole.” (Katzan,1970)



I took some time to study Windows OS’s architecture. It is comprised of two modes, User and Kernel mode. User mode is comprised of many dynamic-link libraries and system-defined processes. It includes a Win32 environment subsystem, an OS/2 environment subsystem, a POSIX environment subsystem, and a security subsystem. User mode is designed to be able to communicate well with applications written by other OS. Kernel mode is in charge of the hardware and system resources. It can run code in a protected area. To keep it simple, kernel mode is like the admin of the modes. Any user function requires permission from the kernel.

1. **Storage Management**:

As storage goes, my opinion has not changed. An SSD is the best storage device available. An SSD supports high speeds, quick reading time, and is closer to the CPU of the computer, for peak performance. I have a table below comparing different categories of HDD’s and SSD’s.

|  |  |  |
| --- | --- | --- |
|  | HDD | SSD |
| Speed | * High latency * Longer to read | * Low latency * Faster reading times * Supports more IOP’s |
| Electricity use | * Uses more electricity | * Uses less electricity |
| components | * Moving parts * Thin layer of magnetic material | * No moving parts * Basically a memory chip * Integrated circuits * 3 components to keep it fast and simple: controller, cache, and capacitor. |
| Weight | * Heavy | * light |
| Vibration | * because of moving parts, it is weak to vibration and could cause damage | * Can withstand vibration of up to 2000Hz. |
| Defragmentation | * Constant need | * Not so necessary |

1. **Memory Management**:

I personally believe that Windows would require 16 GB of physical RAM to run everything with no issues. I personally have had a good experience with DDR4 ram. DDR4 is energy efficient, which allows better performance, and does not require cooling specifications. DDR4 also runs from 1066 to 2133 MHZ, compared to its earlier version DDR3, which runs from 400-1066 MHz.

Memory Management is the responsibility of managing a computer’s main memory. The OS is in charge of keeping track of both allocated and free memory available when in use. For memory management purposes, I would calculate the amount of memory our game would be taking up, and ensure my machine has close to double recommended. For example, if I know Draw it or Lose it requires at least 4 GB of allocated RAM, I would suggest having 8 GB of RAM available to ensure the machine has enough memory to manage everything. Another technique I found interesting was how OS uses Cache Memory to store tidbits of information. To summarize, Cache memory is a way for a machine to save small amounts of memory and use it quickly, in order for the machine to not have to store the tidbit of data in a storage device like an SSD or HDD. With it being right next to the CPU, it will execute at rapid speeds, giving you the quickest results. I find that really cool because it saves space, and keeps the machine running at top speeds.

1. **Distributed Systems and Networks**:

To ensure that it is possible for our game to be connective with all platforms, it would need a database that is shared among all participants. With this game being cross compatible, and online, I would suggest Artery. Artery is designed for online games, that provides excellent speed and performance.

1. **Security:**

Security will always have its ups and downs on any platform. Some have high end security, where other platforms can just “look” at malware and catch them! There are many ways we can keep our servers safe from ddos, ransomware, malware, hackers, and other components that can break our servers and systems. A common mistake people make is setting something really important up, and sealing it with a weak password that is easily guessable. Having a password that makes up of at least 10 characters, and has some form of special character like “!@#” will really help keep your stuff safe, and not as easily accessible by hackers. Some other tips include training people to detect the early signs of a malware attack or ransomware. Having the software used constantly updated and patched will help keep it secure. Making sure that you are using a reliable security application or firewall is a very good piece of advice. Windows has an excellent Firewall built into the OS, that I personally enjoy a ton! Ensure your code is well protected. Something I recently stumbled upon is an authentication protocol named SSL. SSL is like an electronic identification certificate. An SSL is a way to validate any code you are trying to access. It provides a great layer of security for any communication you find online. As of January 7, 2021, SSL 3.0 is supported through Windows 10.

***Resources for Project 2:***

<https://www.redhat.com/sysadmin/six-deployment-steps>

<https://digital.com/best-web-hosting/operating-systems/>

<https://www.ionos.com/digitalguide/server/know-how/linux-vs-windows-the-big-server-check/>

<https://digital.com/best-web-hosting/mac/>

***Resources for Project 3:***

* **Harry Katzan(1970,July) Operating systems architecture**: <https://www.semanticscholar.org/paper/Operating-systems-architecture-Katzan/6fd323024c6c19b9dc353b3d2e51ff448a480353>
* **Wikipedia(2022) Architecture of Windows NT:** <https://en.wikipedia.org/wiki/Architecture_of_Windows_NT>
* **“HDD vs SSD” Diffen.com (2022 ,February):** <https://www.diffen.com/difference/HDD_vs_SSD>
* **Lawrence Williams (2022, January) DDR3 vs DDR4: Must Know Differences** <https://www.guru99.com/difference-between-ddr3-and-ddr4.html>
* **Yaffet Meshesha (2021, June) How to protect your server from Hackers:** <https://www.wikihow.com/Protect-Your-Server-from-Hackers>