

Nicholas Bartlett

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

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

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

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

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

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

[**System Architecture View**](#_heading=h.3dy6vkm)3

[**Domain Model**](#_heading=h.1t3h5sf)3

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

[**Recommendations**](#_heading=h.3rdcrjn)5

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

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.2 | 08/14/2022 | Nicholas Bartlett | Updated evaluation and reccomendations |

**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](#_heading=h.2et92p0)

The client, Chat Away, is requesting a mobile app to market their product on both Apple and Android smartphones. The application needs to be available in the app store of both devices in order for the client to increase revenue.

## [Design Constraints](#_heading=h.tyjcwt)

Chat Away has a few design constraints first being that the application needs to function on both Apple and Android smartphones. Second, the application needs to be developed within their budget which was not provided. Third, they require that the application be available in both the Apple and Android app stores. Last, they are outsourcing the development of the application rather than developing it themselves which will require communication between both companies.

## [System Architecture View](#_heading=h.3dy6vkm)

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.1t3h5sf)

Entity is the parent super class and makes a relationship between all other classes; Game, Team, and Player. They will all inherit properties from Entity which can be seen in the UML design. Each class will share certain common attributes such as their names and id number. Aggregation will be used as demonstrated by GameService having a reference to Game which has a reference to Team which has a reference to 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](#_heading=h.2s8eyo1)

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** | **Characteristics**  Popular for web hosting  **Advantages** Upgradable and has a variety of options for different web hosting **Disadvantages**  Not preferable to the other choices for web hosting | **Characteristics**  Very secure, and preferred  **Advantages** Best option for security as flaws are generally discovered before becoming a problem and is likely the preferred choice for web hosting **Disadvantages**  Less availability to support web hosting | **Characteristics**  Dominant compared to other platforms  **Advantages** Fast loading times and familiarity to users **Disadvantages**  Worse technical support than other options and vulnerable to malware | **Characteristics**  High portability and popular  **Advantages** Has a wider reach than other platforms, is more compatible and cost efficient  **Disadvantages**  Very poor security and is selective to device OS |
| **Client Side** | Cost, time, and expertise requirements will be compatible with Windows and linux. Since the app will be running on all major operating platforms dedicated servers will be a better solution than peer-to-peer | Time, cost, and expertise requirements are the same as with Mac. Dedicated servers are still the best option for client usability and stability. | Time, cost, and expertise requirements are the same as with Mac. Dedicated servers are still the best option for client usability and stability. | Time, cost, and expertise is increased here. Unlike with PC OS’s we have the apple as well as android OS to consider so development time will be larger. Dedicated servers are still the best choice over peer-to-peer. |
| **Development Tools** | Java/HTML/CSS/Javascript/PHP would be the best options for languages for frontend development. This promotes compatibility across all platforms and should reduce overall cost. Python, Java, as well as PHP are all good options for the backend. Eclipse would be a good IDE choice as it is supported on all operating platforms. | Java/HTML/CSS/Javascript/PHP would be the best options for languages for frontend development. This promotes compatibility across all platforms and should reduce overall cost. Python, Java, as well as PHP are all good options for the backend. Eclipse would be a good IDE choice as it is supported on all operating platforms. | Java/HTML/CSS/Javascript/PHP would be the best options for languages for frontend development. This promotes compatibility across all platforms and should reduce overall cost. Python, Java, as well as PHP are all good options for the backend. Eclipse would be a good IDE choice as it is supported on all operating platforms. | Java/HTML/CSS/Javascript would be the best options for languages for frontend development. This promotes compatibility across all platforms and should reduce overall cost. Python, Java, as well as PHP are all good options for the backend. Eclipse or android studio would be a good IDE choice as it is supported on android. |

## 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**: My recommendation is that The Game Room use Microsoft Windows as its platform. It is the most user friendly environment and has a plethora of IDE’s available. This increases the developer’s options for their preference of IDE, so they aren’t stuck using something they are unfamiliar or uncomfortable with which will increase productivity and overall reduce cost. Linux would be a lower cost option if budget constraints are a large concern.
2. **Operating Systems Architectures**: Windows uses Graphical User Interfaces (GUIs) which makes it more user friendly to access system resources.
3. **Storage Management**: During development of this software a resource like GIT would be an excellent choice for storage management. It allows for great version control and for multiple developers to work on the same project. It also automatically detects when two developers have modified the same file and tries to upload it, forcing a manual user review of changes to ensure there are no errors. After development Cloud based storage will likely be the best option as it allows The Game Room to only pay for the storage they need without incurring the running cost of managing their own storage on a server. It is also easily expandible for future use.
4. **Memory Management**: Managing memory should be done by databaseing the large index of images that will be required. Only when images are needed should they be moved for mass storage into memory. To improve load times between games you could pre-load the next games image into memory but on modern hardware this should not be a concern. This method will be especially important for users on mobile and lower end PC hardware as it keeps their limited resources freed up for the OS to use.
5. **Distributed Systems and Networks**: Peer to peer networks would not be ideal here as there are many different operating systems communicating. By using dedicated server(s) to run the game and allowing clients to connect to it, not only will all platforms be able to communicate easily but it should reduce connection issues, provided The Game Room has a stable connection and robust server. The server can create and end games as needed determined on the number of clients connecting to play the game.
6. **Security**: Windows comes with built in security that is regularly updated by microsoft when holes/breaches are discovered. To best protect clients personal information it should be stored separately on the server from the game files. Having users register on an HTTPS webpage then using a hash table to authenticate clients who log into the game will ensure that their personally identifiable information (PII) is never transmitted over the network. There are several encryption methods that will protect users from brute force attacks such as RSA. Two-Factor authentication could be made available to clients who opt into it, further increasing their security without inconveniencing those who do not want to be forced into higher security. Administration users with access to server files should be required to use two factor authentication to protect the files that normal users would not have access to.