

Chat Away

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc111364065)

[Table of Contents 2](#_Toc111364066)

[Document Revision History 2](#_Toc111364067)

[Executive Summary 3](#_Toc111364068)

[Design Constraints 3](#_Toc111364069)

[Domain Model 4](#_Toc111364070)

[Evaluation 6](#_Toc111364071)

[Recommendations 7](#_Toc111364072)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/05/2022 | Charles Adkins | Added Executive summary & design constraints |
| 1.0 | 07/17/2022 | Charles Adkins | Added Domain Model, Evaluation & Recommendations |
| 1.0 | 07/28/2022 | Charles Adkins | Added Evaluation |
| 1.0 | 8/14/2022 | Charles Adkins | Added final Recommendations |

## [Executive Summary](#_sbfa50wo7nsh)

Chat Away, a social media company, is looking to expand its digital presence by outsourcing the development of a mobile application. The organization anticipates that this expansion will market their product to a larger customer base and increase revenue. To maximize the impact of the expansion, the mobile application will be available in the app store for both iPhone and Android smartphones.

## [Design Constraints](#_2et92p0)

Chat Away desires to enhance their marketing reach by developing a mobile application that expands upon their current footprint, which is currently limited to a website. Given that the website has proven to be a success, elements from the website will be incorporated into the mobile application, with small design and functionality alterations being made to increase the user experience within the new environment.

In terms of development requirements, given that iPhone and Android phones operate using different languages, we will either need to fully understand and implement the programming languages for each operating system or use a cross-platform language. Such languages include, Java for Android, Objective-C or Swift for iPhone or Xamarin, which is the cross-platform language. Determining which language to use, based upon personnel expertise and time constraints, will be a critical element to finalize for project success. Using separate languages to develop the application for each platform will be more time consuming, but development personnel may be limited in their knowledge of the cross-platform language Xamarin.

Additional design constraints to consider include determining whether or not to incorporate all the aspects of the website into the mobile application. Including all elements of the website may be time consuming and technological difficult, depending on the complexity of the content, but it will need to be further investigated. Having a slim version of the website, showcasing the most valuable and user-appreciated aspects of Chat Aways’ product, within the mobile application, may be a way to focus on the elements that draw in the most business, while also streamlining the project. Lastly, one large design constraint may include networking and storage limitations. Given that this will be a mobile social media platform, the networking and storage requirements will need to be separate from the website, as demand may overload the current infrastructure. Allocating dedicated servers and databases for the mobile application will ensure that the application can maintain performance as it continues to grow. However, if separate infrastructure is not desired, the current website infrastructure can be expanded upon, with unique identifiers being used to differentiate between website and mobile traffic, which can then be used for performance and business analytics reviews later.

## [Domain Model](#_8h2ehzxfam4o)

In our UML diagram, we see various relationships amongst the classes, which ultimately shapes and underpins the functionality of the game itself. More specifically, looking at the Entity class, we notice a relationship between itself and Game, Team and Player – this relationship is inheritance, meaning that each of the latter classes will inherit data from the Entity class, such as id and name. Additionally, looking at the classes, we see that GameService has a reference of Games, Games has a reference of Team and Team has a reference of Player, meaning that it’s an instance of itself and has a reference to another class. Lastly, we see that the classes have cardinality to their relationships, where GameService can have zero to many Games, Game can have zero to many Teams, and Team can have zero to many Players.

**"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

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Fully capable of hosting web-based applications with terminal bash commands making updates to the server.  Mac allows for several upgrades | Open source, free and accommodates several software options. Downside is that performance and security, as they’re lacking is robustness. | Fully capable, popular and trusted by many. Performance may be scattered, many software packages are tied solely to Microsoft, which makes changes in software an issue. Prone to viruses | Highly requested by users, as portability is important. Cost effective if only utilizing one device type, but potentially heavy investment for multiple mobile device types. Security may also be an issue. |
| **Client Side** | Cost and time required are medium, as it will come down to development team expertise. Cross functionality across web browsers must be considered. | Cost and time will be extensive unless development team is well-versed. Same considerations for cross functionality for web browsers. | Cost and time may be minimal, as Windows OS may be more common knowledge for developers. Same cross functionality considerations for web browsers. | Cost and time may vary, dependent on platforms and developer expertise. |
| **Development Tools** | Notepad++, Visual Studio Code, Atom, Xcode, NetBeans, etc. along with all languages, as well as libraries, not limited to Java, Python, Ruby on Rails, C, C++, PHP, Swift for backend and general use, but also HTML, CSS, Javascript, etc. for frontend and general use. | Visual Studio Code, Eclipse, Atom, NetBeans, etc. along with all languages, as well as libraries, not limited to Java, Python, C, C++, Perl, Go, Javascript for backend and general use, but also HTML, CSS, Javascript, etc. for frontend and general use. | Visual Studio Code, Eclipse, PyCharm, Atom, etc. along with all languages, as well as libraries, not limited to Java, Python, Ruby on Rails, PHP, C#, Perl, Javascript for backend and general use, but also HTML, CSS, Javascript, etc. for frontend and general use. | Android languages (Java, Xamarin) and Objective-C or Swift for iPhone, as well as libraries, not limited to Java, Python, Ruby on Rails, Javascript, Kotlin, C++, C#, PHP for backend and general use, but also HTML, CSS, Javascript, etc. for frontend and general use. |

## Recommendations

1. **Operating Platform**: Objectively looking at the details of all operating systems and the business and technical requirements, I would suggest utilizing Windows as the operating platform. The cost, time and required expertise needed for and by the development team will provide the project with the necessary means to accomplish the task quickly and accurately.
2. **Operating Systems Architectures**: Windows can run applications that use ASP, .NET, Access and MS SQL databases, which are common frameworks and database systems for developers, which easy-to-use interfaces and functionalities.
3. **Storage Management**: GoDaddy is a Windows-based host that uses grid hosting which limits downtime and is cost effective, specifically 10GB of storage, unlimited bandwidth, 100 email accounts, 10 MySQL databases and the ability to run .NET, ASP and other technology stacks for just ~$57/year (Long, 2020). Establishing a backup program and schedule might also be a unique feature that can be implemented to combat the instance of system or network failure, which can ultimate restore application runtime and gameplay in a reasonable amount time.
4. **Memory Management**: Memory will allow for thousands and thousands of pictures to be stored for game use, which can be accomplished through the use of databases, which GoDaddy and other hosts provide, either freely or for a minimal fee.
5. **Distributed Systems and Networks**: Development tools for cross-platform deployment will allow the game to be distributed and played on various platforms, Windows, Mac, Linux, mobile. These engines include, Unity, Unreal, LibGDX, Godot and CryEngine, all of which are free to use, with the option to upgrade to applicable premium features and respective fees (Shareef, 2021). Network latency and uninterrupted performance will be key in providing quality gameplay, so ensuring servers can handle the traffic and power sources will be established with backups in mind, will be important.
6. **Security**: Using third-party security software, in addition to the default Windows security software, will more securely protect the game space, users, as well as client-side and server-side information and data. Additional security features to be considered and or implemented might include a VPN to protect user-to-game connection, multifactor authentication to protect user identity and authorization, implementing and enforcing the principle of lease privilege,

Resources:

Long, Robert. (2020). Omegaweb. Windows Hosting Vs Linus Hosting. Retrieved July 14, 2022 from <https://www.omegaweb.com/windows-v-s-linux-v-s-mac-web-hosting/>

Shareef, Tashreef. (2021). Windowsreport. 5 best cross-platform game engines for game developers. Retrieved July 15, 2022 from https://windowsreport.com/cross-platform-game-engines/