The Gaming Room

# Software Design Template

Version 1.1

## Table of Contents

[Software Design Template 1](#_Toc21086455)

[Table of Contents 2](#_Toc21086456)

[Document Revision History 2](#_Toc21086457)

[Executive Summary 3](#_Toc21086458)

[Requirements 3](#_Toc21086459)

[Design Constraints 3](#_Toc21086459)

[Rationale 3](#_Toc21086460)

## Document Revision History

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 06/06/21 | Howard Bratcher | Template for new games for the company |
| 1.1 | 06/20/21 | Howard Bratcher | Additions to game template. |

## Executive Summary

The gaming room is looking for options for their new platform. They need to know he better options for their new software, where it is best developed, and where it will best be able to service multiple clients at one time.

## Requirements

The client needs specific applications for such a game, it will need server-side support for back end programming and control and a front end system for users to interface with. The core requirements that the client needs are as follows:

* Adding other users as friends.
* Playing with other users, point-to-point communications, instant messages.
* Forums for meeting and greeting people.
* Network mechanics and security users to be paired up to play.
* Security concerns for the users.

## Design Constraints

1. Security on the network for end user information.
2. Privacy concerns for users who do not want to have profiles seen in the public.
3. Have a moderation AI to prevent unsavory content.

## Rationale

The client is looking to solve their problem of finding different applications to run a server on and that work with clients’ applications for a multiplayer game. This can be solved by creating the game to be allowed on multiple platforms. The idea is to market it to as many users as possible, in order to raise profit and a user base. We will be able to prevent alienation of other people simply based on the system they are using. Security concerns on the network should be an upmost priority. Ensuring that user’s private information, used to connect to the servers is kept in a secure manner. Another security concern is protecting the users while in the game from various outside hackers that intend to cheat or break the game in other ways. There should also be various types of physical security around the servers to prevent intrusion.

Two other basic things needed for the game are going to popular in game additions. Being able to add your friends for later gaming and different types of communications. The first is in order to play with your friends you need them in a list to find. Next the idea is to have text or voice messaging to communicate with them. Without these options, a multiplayer game is pointless because people are going on to play with others, not alone. There could also be an option for a forum that allows users to communicate with everyone about the game. These are popular on today’s age because it allows people to group up with new people and become friends.

Lastly the network must be able to sustain users of all types, especially those whit weaker connections because that will become inevitable. The network bandwidth will fluctuate and cause connection problems. It should be noted that there needs to be ways to combat this without completely dropping all connections. Security on this network is important because even though cheaters in game are annoying only to the player, we do not want them exploiting something that could cause servers to crash or them to steal information. They may even be able to place malware known as ransomware on the server and cause problems for the company.

**Server Side:**

When dealing with the server side of the application, I would choose Linux over any other operating system. Windows has plenty of utilities that can cover a wide variety of issues and MAC OS is also viable however the costs of these two products are way more expensive than a free server version of Linux. Microsoft will charge a per processor fee and only give so much power for X amount of cores on a processor. This means that in the process of building a server for a game you may become limited due to versions of windows that can cost a lot of money. MAC OS can also be expensive; however, it does offer the same similarities as Linux considering the back ends of both operating systems are identical. Mobile servers would simply fail in their job and not be able to perform to the needs of the company. In my opinion Linux will offer the best possible outcome because it allows us to properly communicate with the server.

Linux is important because it will allow someone to edit server settings, commands, and other details for a game while it is running through the back end of its shell. This kind of communication is important for developers to think about considering it is imperative to keep the server healthy for thousands of users. The days are also in the past that Linux is incompatible with other software’s as well and this makes it a prime choice for being chosen as a proper server suite. And lastly, Linux will serve its purpose well considering there are plenty of free options to choose and a community that is very supportive for arising issues. It should be noted that most mobile platforms run on Linux but are still bad choice for running a sever simply because there is not enough power output for thousands of players.

**Client Side:**

As far as client-side software is concerned, Windows will beat most competitors by default. This is a more expensive route than Linux, however there are more compatible game created for it, it has become more secure over the years for files and data, and most importantly it is easy for most users to operate. Windows shines like a star because for this category because it allows users to download and install files automatically and has multiple clients that can keep the files organized into their own organizational units. Steam is a popular gaming platform, along with other ones that do most of the work for the user. These platforms have their own protection for gaming online and windows has plenty of other protections as well such as its NTFS file system. This allows the user flexibility and comfort for most of their needs, they can also easily install other applications such as VPNS to help protect their data online. Linux is a free option for people who have more experience in the field and is slowly growing in popularity for developers to create applications for and MAC OS is just as expensive, if not more expensive for users because you must purchase apple hardware. Mobile devices will work well for some simple on-the-go games but do not have enough power to reach the potential of its forefathers on the desktop computer yet.

**Development Tools:**

There are tons of different developer tools to use on the market such as Visual Studio for C++ and Java coding. There is also eclipse, Dr. Java, and tons more of different applications that allow us to code for any device. The compiler will allow us to write the code we want and compile however we need to for any device. There are even online tools accessible from any web browser such as “repl.it”, these work amazingly well for devices because they are accessible from anywhere. The tools used by a team are important because they will determine how well the product turns out. If the users can’t operate on the applications, then we run into the issues where nothing gets done or its not written correctly simply because people are unable to use the tools.

**Operating Platform:**

The best operating platform for the game is windows as an operating system background. This is because it is one of the most used systems for anyone with a standard computer. Most end-users do not use Linux because they believe it takes a lot of know-how, ad macs tend to be more expensive. Different mobile operating systems can be unstable and harder to code considering the amount of different OS patches they have with different types of hardware architecture. Windows is the most user-friendly OS and because of this, making the game for this platform would get it to more players.

**Operating System Architecture:**

Each operating system has different types of architectures, most main OS has features to it. The two major types are 32 bit and 64 bit. These each provide the end user with two major aspects, however the 64bt is the most important. These allow the user to use ore ram and the creator to use more of it in order to run the game better. The more memory, the more the program can run.

**Distributed Systems and networks:**

Distributed systems are simply a network of computers. Within this network we are able to have multiple systems connect for various reasons. For the game, this allows us to have our users connect to each other over the game and play with each other over a network. There are two different ideas on these networks, the groups of players and the groups of servers for the game. Ideally these two groups are spread over a wide area, and this includes the different groups of players connecting over the different groups of servers. This can help with network issues that may arise in the case of players being too far from each other. This helps resolve issues of poor connection to any device whether it is the server itself or another player.

These systems will have to implement different ideas as well considering players need to communicate with each other or other servers on the network. One important aspect is the network architecture, its placement on a geographical location for user to connect too. Another is where it is located for access of security reasons and interference. Another idea to factor in is a shared database for users to use for their data or even a shared database for developers to use for game data. These systems are spread across large networks in order to provide access to multiple different people.

**Memory Management:**

Memory management is an important concept in any application being created for any device, especially an application that is meant to be run on multiple different devices. This is something that should be paid attention to because memory is what is being used for any currently running application. As a quick run-down, an application that is running is doing so in the RAM of the device. If we use windows as an example, Applications may run in the background, but if that process is doing something it is going to consume part of the RAM. Your application must also be able to manage the RAM it uses to function properly; this means handling the application in a way so that unused portion of memory from your application are freed up. Eventually, the RAM can be filled up and this is where memory management comes into play.

When you are creating an application, the larger it is and the more it does will inevitably use more RAM. Two factors are important to remember during these periods. How much processing power will your application need for what it does, and once it is done using a portion of memory, can you free that space up for more use of your application or another one. Any client using your software will not want a slow computer or other problems coming into play. Considering that the computers performance is dependent on RAM then we need to ensure that the application being created does not soak up memory. When coding this means clear, concise, and clean code. It should do what it needs to do and then delete any unnecessary trash. If this trash builds up it can cause various issues such as computer crashes, memory leaks, and more. Creating your application with only the necessary code and code that works well together will inevitably allow you to make a smaller cleaner running application that will be able to manage its memory well. Also, investigate the operating system that the application will be running on. If it can work with the Operating system well, then that OS can help manage the memory for you too.

There are other ideas as well when managing memory, you can ensure that you are reusing memory in your code, allocate it where you need it and delete it when it is not being used. For example, if your using pointers, this could help eliminate a very lengthy piece of code if you’re not using them. We can point to other pieces of the application where data needs to be handled or stored instead of creating new data to hold values. Another of us of pointers is in the idea that instead of creating temporary variables to hold data, then equaling a variable to that data to overwrite it, you just point to it and overwrite it that way. This seems harder to do when coding, however, your code will add up and you will create a cleaner piece that runs better.

**Storage Management:**

System storage is another factor for your application. Different factors that are important to keep in mind are the size of the application and permissions the application needs. When considering storage, think about the devices being used today. Most people are opting for portability over desktops in their daily lives and most corporate business are opting to use thin clients in their office. A thin client is a device that only has barebones parts and runs mainly on a network. This means if someone wants to use your application, it should be as small as possible because they could run into installation issues. Older model computers that have bult in 32GB flash memory are running into issues in today’s age. They run great, I personally have one, but they cannot keep getting updates because the updates to the OS, specifically windows are just too large now. There are more than likely fixes coming to these, however there may not be, and this can render the device useless or a larger security risk for the users’ data considering it is out if date.

When considering other parts of storage management in creating an application, redundant data can be a big issue. If you are storing your data multiple times, but not s=deleting it then this can build up on a hard drive. For example, windows may create log files for certain processing tasks you can set. These logs can show what a processor is doing over time and help us find issues in the Information Technology field. However, if not set up properly it will fill up an entire hard drive with log files and h=then overwrite those files when storage runs out. No matter what the device is, there is always a limited amount of storage. If a user runs the application and it continuously saves trash data to their system but does not have a way to delete temporary files, then this can cause problems.

Another part of storage management are the permissions on the device. Every device may have specific permissions that will need to be granted towards your application. This allows you to store data on the device. If you do not create a way to properly access files in the operating system, this leads to issues such as security risks or loss of data. If the application does not know to create a new file name every time a log file is made, then this example shows that it will keep overwriting itself. If this is important data, then no client will want to use your application. Other different factors to consider in managing your memory storage are backups of data and how often they are done, Will the application be able to find the data once it is stored, and is the data secured properly; especially if it is sensitive information (Schiff).

**Security:**

Security over the networks is another large issue that can have widespread effects on the game if it is not taken seriously. One of the most major concerns is network security. In most games in today’s age, we are always in the need of giving up data for them. For example, on mobile devices, a lot of games are considered free, in reality, the game may be free, but they are going to profit off of our data. Also, it is good to note that you volunteer to give some data up when signing the end user license agreement, in other words, “Just checking the box”. Network security is paramount because if this data is being transmitted as you sign into the game and a hacker is able to intercept those packets, they now have your data. One email or one password used multiple times can be the end up being the way they get into your bank account. Data from the questions used to reset passwords can affect other logins such as your bank as well. Therefore, it is important to make sure we keep this data secure. Another factor of security is the physical side of it. Keeping servers physically locked or under guard, depending on the data, is important to keeping it safe from anyone with a simple usb stick.

**Works Cited**

Franklin, Curt, and Dave Coustan. "Memory Storage and Management." *How Operating Systems Work*, HowStuffWorks, computer.howstuffworks.com/operating-system7.htm.

Schiff, Jennifer L. "14 Things You Need to Know About Data Storage Management." *CIO*, CIO, 11 Sept. 2013, www.cio.com/article/2382585/14-things-you-need-to-know-about-data-storage-management.html.