



The Gaming Room
CS 230 Project Software Design Template
Version 1.0

Table of Contents

CS 230 Project Software Design Template	1
Table of Contents	2
Document Revision History	2
Executive Summary	3
Design Constraints	3
System Architecture View	3
Domain Model	3
Evaluation	3
Recommendations	5

Document Revision History

Version	Date	Author	Comments
1			
1.0.0.01	01.23.2021	Arison O'Hara-Hulett	Updated executive summary, design constraints, domain model UML Recommendations

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

Currently “The Game Show” application is only available on java enabled web browsers and android based devices. The client would like their game room game to be based off of the 80s television game show. “Draw it or lose it”.

Design Constraints

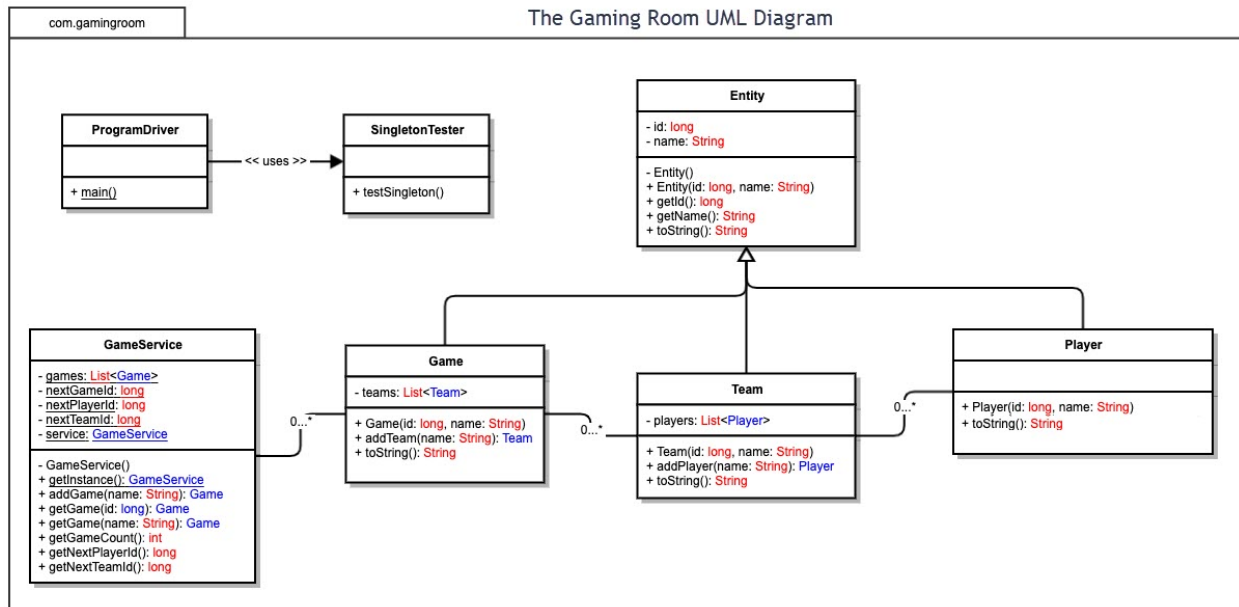
The Design constraints lie in the memory, connectivity, security. Assuring that there will be sufficient bandwidth for the files, images and player interaction, along with consistent connectivity, and security of maintaining and allowing unique gaming sessions. Making sure all of the images are compliant with any licensing. Having the functionality of the program maintain consistent gameplay with server interaction as well as making sure the game is not overly exciting and seizure inducing, does not contain material that is sensitive in nature *with out adequate warnings, and the general appearance of the game and the navigability.

System Architecture View

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

Entity is the ‘game room’ instance, this instance has a few accessor methods for retrieval of ID & Name along with a transcription (toString) method. By communicating to the child classes: Game Service, Game, Team & Player, Entity will invoke a running game room session. Each game from Entity will have a game ID and name, those have being refactored for unique game sessions. Further in depth each child class has its own methods of operation to produce a team and player abiding by ‘The Game Rooms’ session rules. All classes are called upon by the Program Driver class in subordinate clauses currently under testing by the Singleton Tester class.



Evaluation

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 API, Server-Security Documentation	Linux API, Server-Documentation	Windows API, Server-Documentation	Mobile Devices API, Server-Doumentation
Server Side	Mac is easy set up and low maintenance, having customer support with industry leading professionals is a plus. The server interaction is done through XCode	Linux Servers are well equipped and monitored along with industry leading professionals Linux comparable to both Windows and Mac (Documentation provided for: Red Hat Linux) Recommended language is Python	Windows is fairly low maintenance as well however picking the right server version could be an ordeal as not all servers provide the same services. Swapping to a new version in the future is a possibility, if the game loses or needs an update. *potentially a fallout if deprecation of the current server.	The Mobile server chosen is currently Atlassian : Confluence. *The API & is through Android Which is low maintenance and easy to set up. Currently the only supported server. For Android the recommended environment is through Android Studio
Client Side	A final web based version of this game requires minimal expertise. The device should be java enabled, and connected to the internet.	A final web based version of this game requires minimal expertise. The device should be java enabled, and connected to the internet.	A final web based version of this game requires minimal expertise. The device should be java enabled, and connected to the internet.	The device at this time should have access to the Google Play store. For greater availability, porting a version over to apple via Xcode is a possibility.
Development Tools	Depending on the form of application, A host such as Apache, or android studio, a container such as Kubernetes or docker is an acceptable database management system Xcode is recommended	Depending on the form of application, A host such as Apache, or android studio, a container such as Kubernetes or docker is an acceptable database management system. Any text editor. VI	Depending on the form of application, A host such as Apache, or android studio, a container such as Kubernetes or docker is an acceptable database management system. Microsoft Visual C++ version 6.0 or later with the Uni-code MFC libraries	Depending on the form of application, A host such as Apache, or android studio, a container such as Kubernetes or docker is an acceptable database management system. Xcode is recommended

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:** Available for all systems that run Java applications. Mac OS, Windows, All Java compatible Linux distributions. Mobile devices running compatible android version, running a Java enabled web browser. Java version dependent.
2. **Operating Systems Architectures:** Mac has now developed enhanced graphic and IO handling providing an enjoyable safe gaming experience. Macintosh is running a proprietary CPU on some of its newer devices and the Kernel is XNU, here is the link to Mac mini and its technical specifications [MAC-MINI](#) the systems architecture can be found [OS Architecture](#). Windows XP source code was leaked on to the internet earlier this year. However the general system architecture can be found [here](#) Windows can be run on any computer that meets its minimum requirements for installation although having more RAM CPU and hard drive space is recommended.
3. **Storage Management:** Any Java compatible storage, Kubernetes is preferable at the time Cloud storage is an option via the server connectivity, from apache. *[Kubernetes Docs](#)
4. **Memory Management:** The Game Room uses a series of enumerated files pertaining to the storage of session ids username ids and other authentication methods storing and managing credentials. Variables are stored currently as long, string, and a series of unique gaming instances. The incrementers in place are form validation to provide a secure environment while gaming to avoid collision of data streams. Authentication and security management is handled in majority by the basic security libraries.
5. **Distributed Systems and Networks:** Utilizing current thread capacities, running a continuous stream on the hosting web browser to a dedicated server will provide access to the game session. Storing input from the end user's machine and uploading on retrieval from the machine to the server. Allowing for continuous game play with minimal interruption ie crossing streams and de-authentication of current gaming sessions. Through the server's encrypted connection management system allowing a secure seamless gaming experience.
6. **Security:** Along side with IP and device masking. A server provided VPN will provide a more secure level of gaming. Users whose accounts have been expired will need to re-authenticate and acquire certifications through latest security protocols. Handling of 2 factor authentication can be handled through some API's that provide those libraries are [authy](#) and [fusionauth.io](#)