Draw it or Lose it Web Application

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** 4
- **Recommendations** 6

Document Revision History

Version	Date	Author	Comments
1.0	12/12/2021	Alex Metzger	Finalized report for The Gaming Room Client

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

The Gaming Room wants to expand their Android app, *Draw it or Lose it*, to a web-based game that serves on multiple different platforms. This will require the current code for the Android application to be transformed so that multiple, oftentimes different, platforms can communicate with one another when the game is running. This will ensure that the game will run smoothly across multiple platforms via the web.

Design Constraints

- The web-application must mimic the already functioning Android application
- The application will need server-side assistance to help maintain a fluid management of the many different teams across many different platforms
- In order to fulfill the overall design of the application, an Object-Oriented Language must be used to code the application. In this case, Java will be used.

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

The UML diagram below illustrates the overall design and flow for programming the application. Inheritance is demonstrated by the Game, Team, and Player classes in that they all share attributes, particularly the getID(), getName(), and toString() methods. These methods will be defined in the parent class, named Entity, and the three classes mentioned previously will be subclasses of this class.

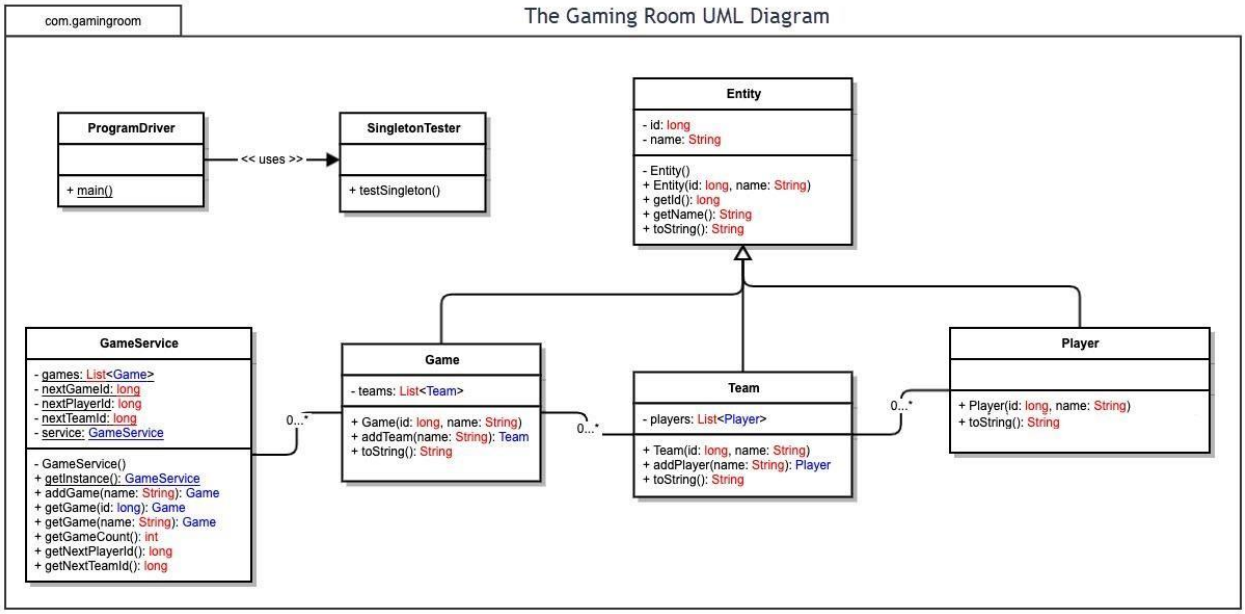
Each of these subclasses will be related to each other in 0…* (or 0 to many) form. This means that There can be 0 to many instances of the Game class, which will contain 0 to many Team classes, which will contain 0 to many Player classes.

Note: The word ‘contain’ does not imply inheritance. All three of these classes are only inheriting the attributes of the parent Entity class.

The only class that will always be present is the GameService class. This class will contain the Game instances, also in 0 to many form. Again the 0 to many form implies that there can either be 0 or many Game classes, which means the GameService class can either be a stand-alone class or contain many instances of the Game class.

This GameService class will use the Singleton Design Pattern to ensure that only one instance of the class will be created at one time. This will ensure that memory will not be overloaded and that new game instances will be added to the same GameService instance.

To ensure that every Team and Player instance has a unique name within a game, the iterator pattern will be implemented. This iterator will iterate through all taken names and determine whether or not the name is already taken. If it is taken, an error message will be displayed. Otherwise, the name will be added to the list of team/player names.



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.

Development Requirements	Mac	Linux	Windows	Mobile Devices
Server Side	Mac is an expensive option for server side operations. Mac has terminal commands to deal directly with server-based operations.	Like the Mac, the terminal has commands that directly deal with server-side operations. Also a cheaper alternative to both Windows and Mac.	Like the Mac, this option can be expensive when compared to Linux. Much easier to navigate the platform through its userfriendly GUI. Has access to many server specific commands via the command prompt.	Most mobile devices will not be able to support the amount of traffic that will be desired for the server-side of a web-based application. Weak specs compared to other OS's.
Client Side	User-friendly GUI to aid clients, so requires less expertise and saves more time than Linux. Most expensive option for clients.	Least expensive option for clients. Difficult GUI to navigate with a steep learning curve. Most amount of time and expertise required.	Least amount of time and expertise required to accurately navigate the OS. Moderate expense compared to Linux and Mac.	Specs usually make web-based applications designed for other platforms, especially games, perform poorly. Takes a significant amount of time and expertise to run these types of applications on mobile devices.

Development Tools	Supports all necessary languages for web development including HTML, CSS, Javascript and Java. Supports common IDE's such as Visual Studio, IntelliJ IDEA, and Eclipse, and Atom. Exclusively supports Xcode. Uses the terminal for commands.	Supports all necessary languages for web development including HTML, CSS, Javascript and Java. Supports common IDE's such as Visual Studio Eclipse, and Atom. Uses the terminal for commands. Open source which gives access to many extra development tools for free.	Supports all necessary languages for web development including HTML, CSS, Javascript and Java. Supports common IDE's such as Visual Studio, IntelliJ IDEA, and Eclipse, and Atom. Uses the command prompt for commands.	Supports all necessary languages for web development including HTML, CSS, Javascript and Java. Does not support IDE's. Limited tools for development.

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**: I recommend using the Windows operating system for the development of this application given that Windows performs well on server-side functionalities which will be immensely important given this is a web-based application. There will be no constraints in terms of development tools or IDE preferences when using this platform. This platform is also reasonably priced compared to the MacOS and requires less of a learning term for clients when compared to Linux.

2. **Operating Systems Architectures**: The Windows platform provides a hierarchy of systems and services with the lower-level functionalities of the hierarchy including hardware and driver functionalities, all performed by the system's kernel. This allows the user to focus on higherlevel functionalities such as server-side service and development.

3. **Storage Management**: If Windows is used to develop this application, local storage management is greatly simplified using the GUI that comes with the Windows platform. Windows also provides many built-in tools to aid in disk space management within the system.

   As far as sharing files between local machines, Windows provides access to many popular cloudbased storage options such as Google Drive. The operating system comes with an application called File Share which can make sharing files much easier. Github is always a great option when it comes to file sharing as well.

4. **Memory Management**: Windows has evolved its memory compression techniques to provide optimal memory management within its systems. The operating system is very good at partitioning RAM space on its own through underlying functions provided by the platform's kernel. All of these techniques are seamless to the user, due to the system's hierarchy architecture chain.

5. **Distributed Systems and Networks**: One of the most important aspects of the Windows operating system in the use of developing web—based applications such as Draw it or Lose it is the fact that it is available in a distributed version. This means that memory and tasks can be split up between multiple servers to optimize performance. By using Windows to develop this web-based application, the development team will gain access to this distributed style of operation and will ultimately deliver higher performance standards.

6. **Security**: Windows has a multitude of security features pre-built into the operating system to help keep user data safe, which is always a priority. Windows Information Protection is one of these features that specializes in preventing data leakage within a company across many sharing methods, including cloud sharing. Windows also has a feature called Device Guard that is essentially an anti-malware application that comes pre-installed in the platform. These features along with many others will ensure that user data is as safe as it can be. Though there is always a threat to security in any platform, Windows is among the top competitors in this area.