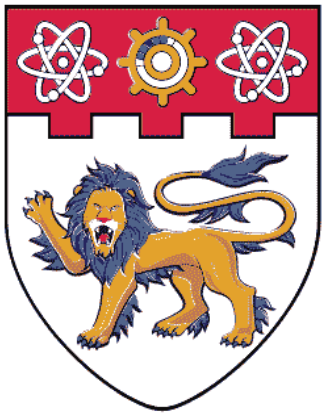


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Drawdiculous

Design Report on Software Maintainability

Version 1.2

Prepared by Team Luck

Bian Hengwei	U1923732B
He Yinan	U1922693C
Jin Han	U1922733A
Lee Yu Jie Melvin	U1922533G
Renganathan Ramasamy	U1922494L
Sannabhadti Shipra Deepak	U1822459L

Document Change Record

Revision	Description of Change	Approved by	Date
1.0	Initial Template	Sannabhadti Shipra Deepak	14/10/2021
1.1	Add Design Strategies	Sannabhadti Shipra Deepak	14/10/2021
1.2	Add Architectural Design Patterns and Software Configuration Management Tools	Jin Han	17/10/2021

Table of Contents

1. Design Strategies

1.1. Introduction

The ‘Drawdiculous’ application is a multiplayer game that is designed for widespread appeal. The current prototype is built on restricted resources, and is only able to handle a limited number of players at any given time. Given these limitations, the nature of this application makes it imperative to design the game with scalability in mind. Therefore, the project planning phase also included a great emphasis on future robustness and maintainability.

The Client-Server architecture model and layered architecture model was used in the construction of this application as its advantages include modularity, simplicity, maintainability, flexibility, scalability, portability, robustness and implementation stability.

1.2. User Focus

The application was created to specifically target the elderly user group. However, due to protocols and safe-distancing measures through the COVID-19 pandemic we were unable to hold focus groups to have our target user base provide their inputs. In order to circumvent this, the ‘Drawdiculous’ team looked into the existing research and expert analysis provided by reliable independent studies. The team had extensive brainstorming and testing sessions to discuss the relevant features of the application, and provide continuous feedback on its design and usability.

1.3. Feature Management

In order to effectively manage and roll out application features, there was an emphasis on testing during the design process of the application. The overall strategy was to implement features atomically, and test them individually, to allow for easy error detection and correction, and integrate the features and conduct integration testing.

1.4. Maintenance

Our design goals specify that we intend for ‘Drawdiculous’ to be easily adapted to new operational environments. Additionally, we intend to implement continuous perfective maintenance by detecting issues and correcting them as quickly as possible, well after product delivery. This is in order to reduce maintenance costs and ensure quick turnaround. The application is well-documented, with readable code, and extensive modularity to uphold our commitment to upholding quality in both process and product.

2. Architectural Design Patterns

"Drawdiculous" uses the Client-Server architecture and the Layered architecture within the server and client. The application is decomposed into server and client whereby they are further decomposed into groups of subtasks, each of which is at a certain level of abstraction. Each layer provides services to the next layer above, allowing reusability.

Server

The three layers are:

1. Interface layer

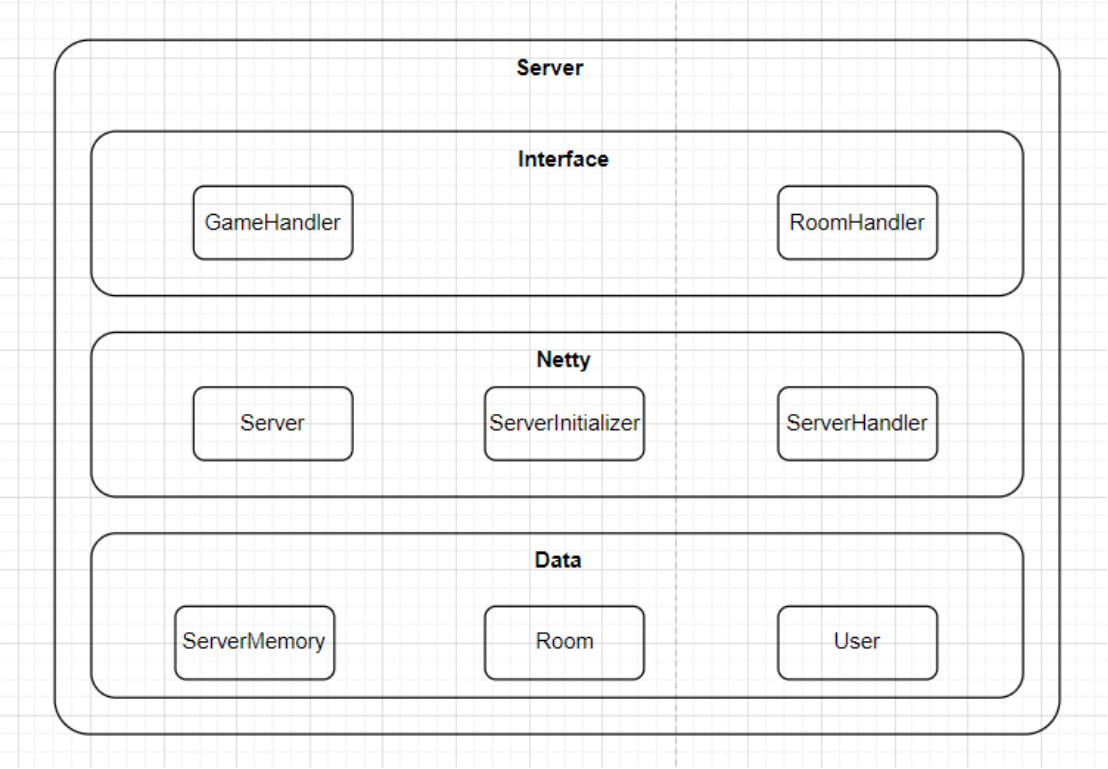
Provide an interface for clients to call and use server features.

2. Netty layer

Uses features provided by Netty framework to initialize and set up the server.

3. Data layer

Contains model classes and storage to store server side data.



Client

The three layers are:

1. Presentation layer

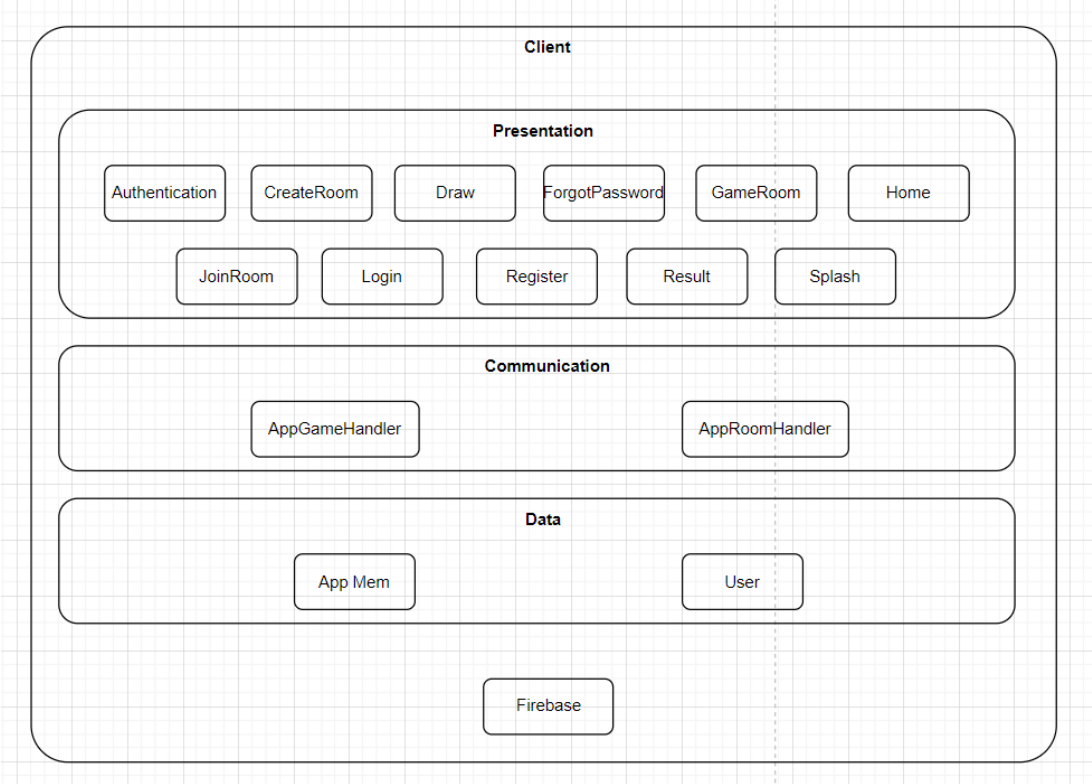
Contains boundary classes to provide user interfaces.

2. Communication layer

Communicates with the server by handling server responses.

3. Data layer

Contains model classes and storage to store client side data and runtime data.



3. Software Configuration Management Tools

This is where we discuss version control management, and track who made what changes and when.

1.1. MediaWiki

MediaWiki is a free and open-source application for document organisation. This service is very accessible and easy for beginners to pick up. There are many FAQs provided which can teach users the functions required by the users. There is a wide range of functions which allows users to create their information in different styles. It also allows users to concurrently edit the page at the same time. Hence, editing of the page will not result in a loss of information.

1.2. GitHub

GitHub is a source code hosting platform using the distributed version control and source code management Git. GitHub is chosen for its familiarity and support provided by various IDE applications. GitHub also supports issue tracking similar to a ticketing system. Whether it's a software bug, code enhancement or documentation, users can open an issue, label them appropriately and assign them for other team members to resolve. All users involved will receive timely updates on the progress of the issue.

1.3. Google Drive

Google Drive service is used as a file storage and for the backup of documents initially created. This service allows users to share and store files within the group easily. Furthermore, Google Drive allows users to edit documents using the full functionality of Google Docs Editors office suite. This service allows users to edit documents concurrently and supports version control.

1.4. Jira Software

Jira Software is a software application used for issue tracking and project management. It includes the Core tool which enables change requests, workflow approvals and general task management, which is designed for non-technical teams. Jira Software includes the Core features, as well as agile functionality, which is used for bug tracking, managing basic software development tasks and project management. This service helps the team to manage work for all kinds of use cases.

1.5. SVN

SVN is a software versioning and revision control system. All documentation is tracked and recorded using SVN. Different versions of documents are kept tracked of, and their history is maintained so that they can be rolled back when an issue arises. This service enables maintainability for the documentation.