EDAF50 Project Rapport

Niklas Hedström
dat15nhe@student.lu.se

Daniel Jannemyr Åhlin
dat15dah@student.lu.se

Simon Hyttfors
dat15shy@student.lu.se

Emil Wihlander
dat15ewi@student.lu.se

22 maj 2019

1 System design

An overview of the system can be seen in figure 1. In designing our system we wanted to have a structural and logic way of setting it up. Therefore, we chose to do a struct containing information about Newsgroups and one containing information about Articles. Each newsgroup have an article counter which handles the unique value for each article in a newsgroup. Furthermore we have one class, MessageHandler, that takes care of the message handling between the server and the client.

1.1 Server

Named YourServer, is the main program for the server. The server connects to the `server.cc` given from start. For all the different commands available, there is a method that takes the Database and the MessageHandler as inputs and does the logic for each command. We chose to breakout all of the methods from the switch-case due to better readability.

When starting the server, you have to type in 1 to launch the in memory database and 2 to enter the disk database as argument.

1.2 Client

The client has a number of different methods that makes the code easier to understand. A switch case is used to make it easy to choose what you want to do and is a clear way of interacting with the server.

1.3 Database

The Database is a pretty basic class, we wanted an abstract class such that we easily could implement the same type of methods in both our in memory-variant and our disk-variant of the database.

1.3.1 In memory database

The in memory database uses a map that store all the newsgroups with their respective ids as keys to make it easy to find a specific newsgroup.

1.3.2 Disk database

Connected with SQLite3 to fetch information from the database. All functionality is there except that all articles get numbered uniquely in the large system and not uniquely in the newsgroups.

There is also a couple of help functions to make sure that the communication with SQLite3 works as expected.

2 Classes

client.cc The class describing the client.

database.h The abstract class of the Database.

diskdatabase.cc/h
The class that communicates with the SQL-server to fetch information from our database.

inmemdatabase.cc/h
The class responsible for the in memory version of the database.

messagehandler.cc/h
The class respnosible for take in information and transform it to byte code and transform byte code back to information such as ints and strings.

newsgroup.h A header file containing the struct for both NewsGroup and Article.

youserver.cc Uses the class server to take in information and do something with it and send it back.

3 A day in the life of the News system

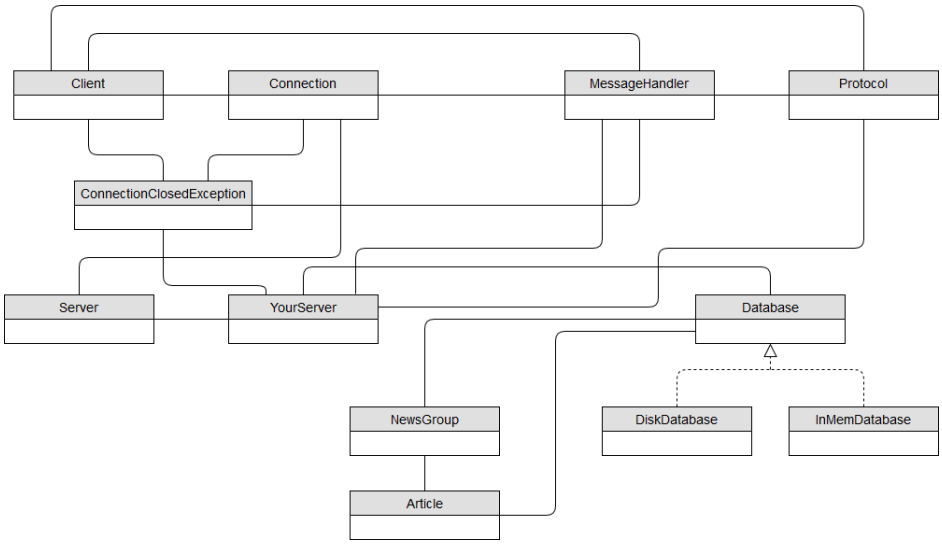
- The server receives a command
- The message handler receives the byte, interpolates it and give the information to yourserver.
- Yourserver gets the information and, through the swich-case, decides what method to call.
- Once the method has been callen upon, the message handler sends an answer to the client and performs the given command and sends more information to the client if needed.
- The message handler sends an answer end to the client so that the client now can get the information.

4 Conclusion

We have fulfilled all the requirements for the project. We think that the project is in a good shape for the future and that nothing really should need any changes. From start we found the project description a bit overwhelming and hard to grasp, but after some deliberation we solved it eventually.

Functions for SELECTING from the database would have to be implemented, this would be possible by further studying how to use SQLite along with C++.

Appendix A



Figur 1: An UML of our system