

CS102**Spring 2017/18**

Instructor:

Aynur Dayanık

Assistant:

Sevil ÇalışkanProject
Group**G4****~ GShare ~****Me && The Boys****Güven Gergerli, Çağrı Eren, Bora Çün, Tuna Dalbeler, Kaan
Ateşel, Onur Korkmaz**

Criteria	TA/Grader	Instructor
Presentation		
Overall		

Detailed Design Stage Report

(v1)**28 November 2019**

1. Introduction

GShare is a sharing application that allows people to find borrowers and lenders near them. The application uses a virtual currency called “G”, so that the users earn G when they lend, and pay G when they borrow.

2. Details

2.1 Some Descriptions

ChatCollection: sortChat() Takes the chat with the latest message to top.

Chat: typing is true if the other person is typing at the moment, and false if not.

Chat: getStatus() gets the status of the chat (Not agreed, Terminated, Agreed, Waiting for return, Returned).

Notice: location indicates the location where the notice is posted.

Notice: finish() When a transaction is complete, this method does the necessary procedures.

Notice: agree(User noticeTaker) Activated when both parties click agree. Does the necessary procedures.

Location: isInRange(int Range) Returns true if an object is in a specified range.

PrivateProfile: getQuestsToDisplay() Allows the user to see the badges the user displays.

G: compare(User user1, User user2) Compares the G values of two users

QuestCollection: sortQuestCollection() Gets the completed quests to bottom.

NotificationsScheduleManager: setAlarm(String date) When two users agree, this method sets the appearance time of the notification.

NotificationList: selectNotices() Selects the notices to be shown on top.

Notice: computeTimeLeft() Computes the time left to return the item.

Notification: canDelete True if notification is deletable

Server: run() Starts the connection.

Server: getConnection() Gets the connection.

Registration: validate() Validates the user from database according to the info the user entered.

Registration: doesExist() When a new user tries to register, this method checks if this user exists on database.

Registration: hashPassword() Creates a hash of provided password for security purposes.

Client: run() Establishes connection with the server

Quest: total The number of times the user has to do a thing to achieve the quest.

Quest: progress The number of times a user has done a thing in the quest.

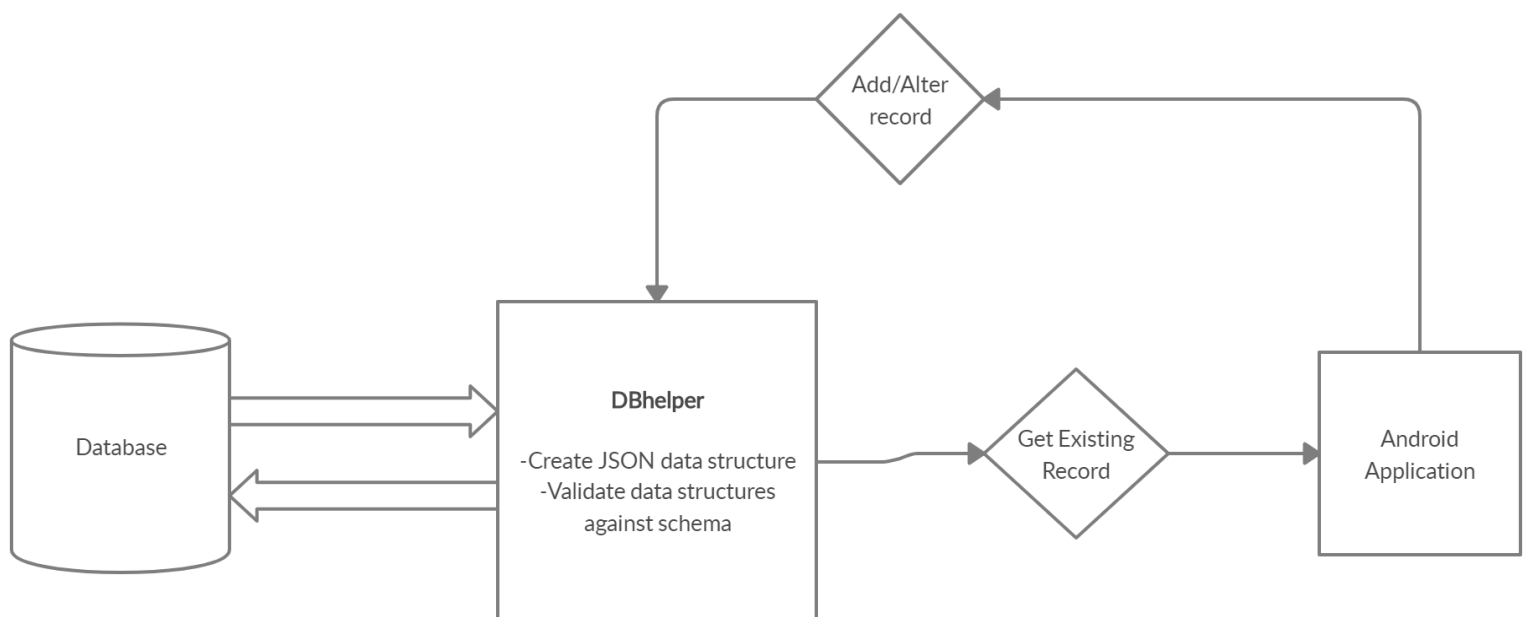
QuestCollection: questList An arraylist to hold all the available quests.

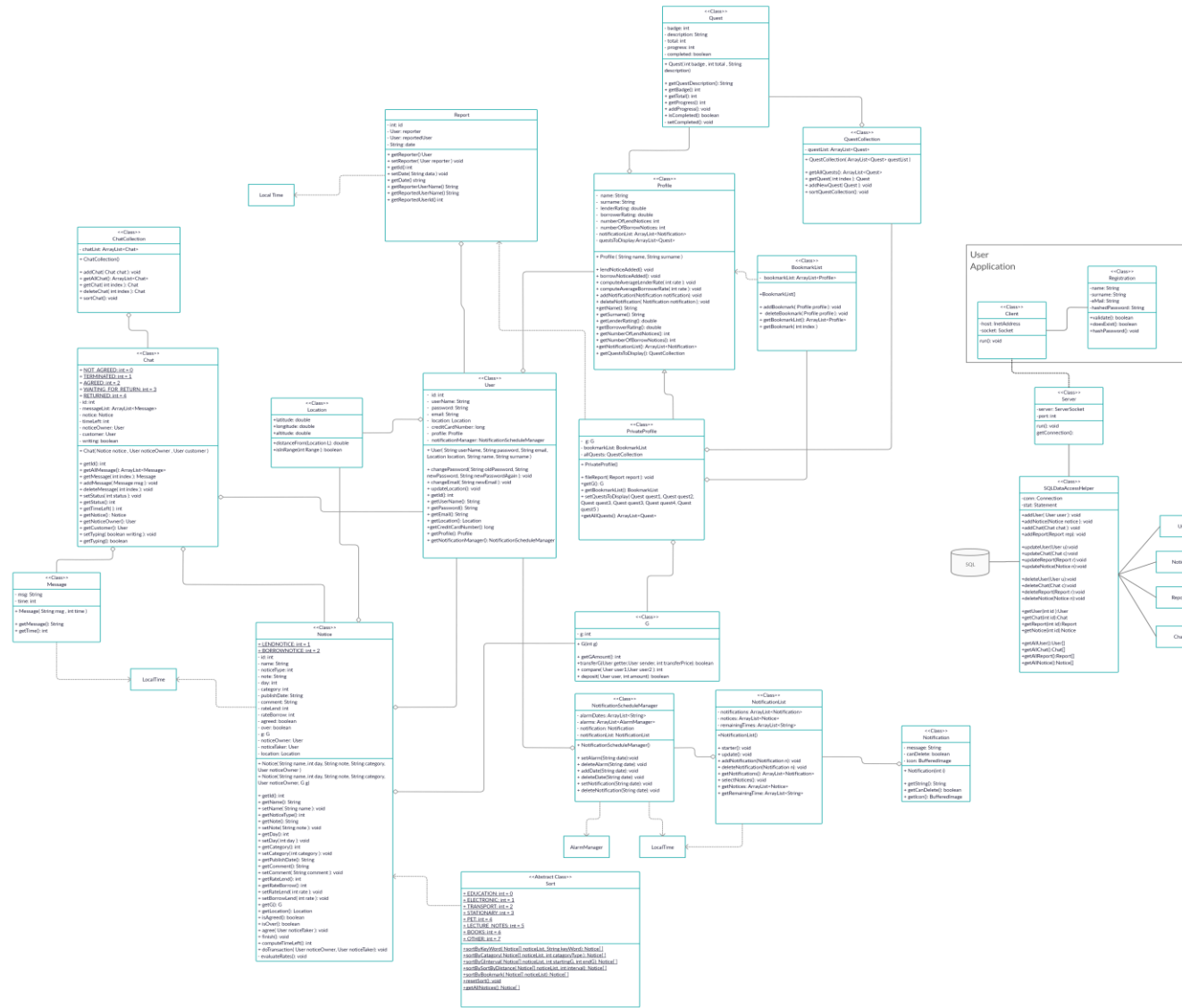
Asdasd

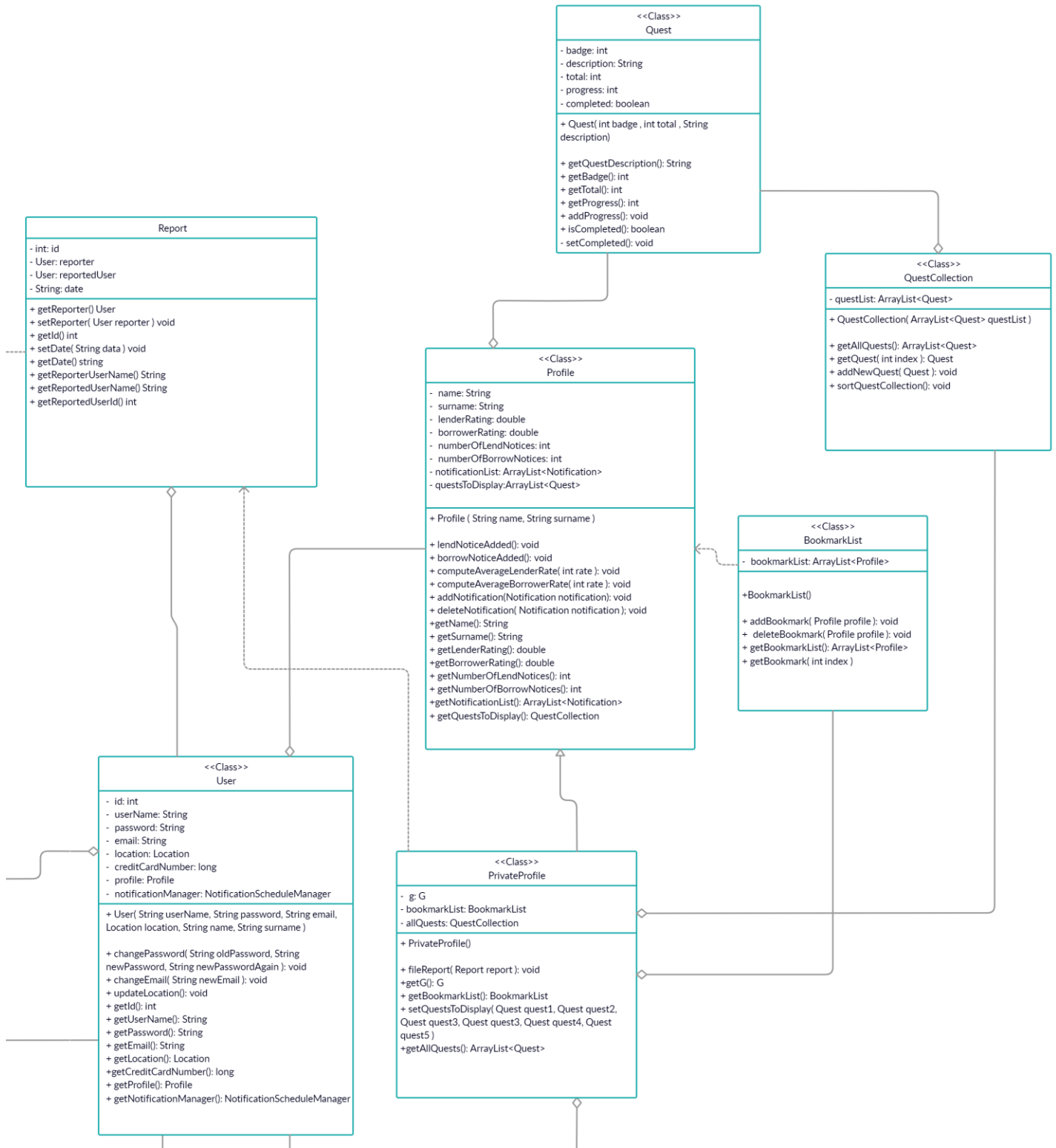
2.2 Used Technology

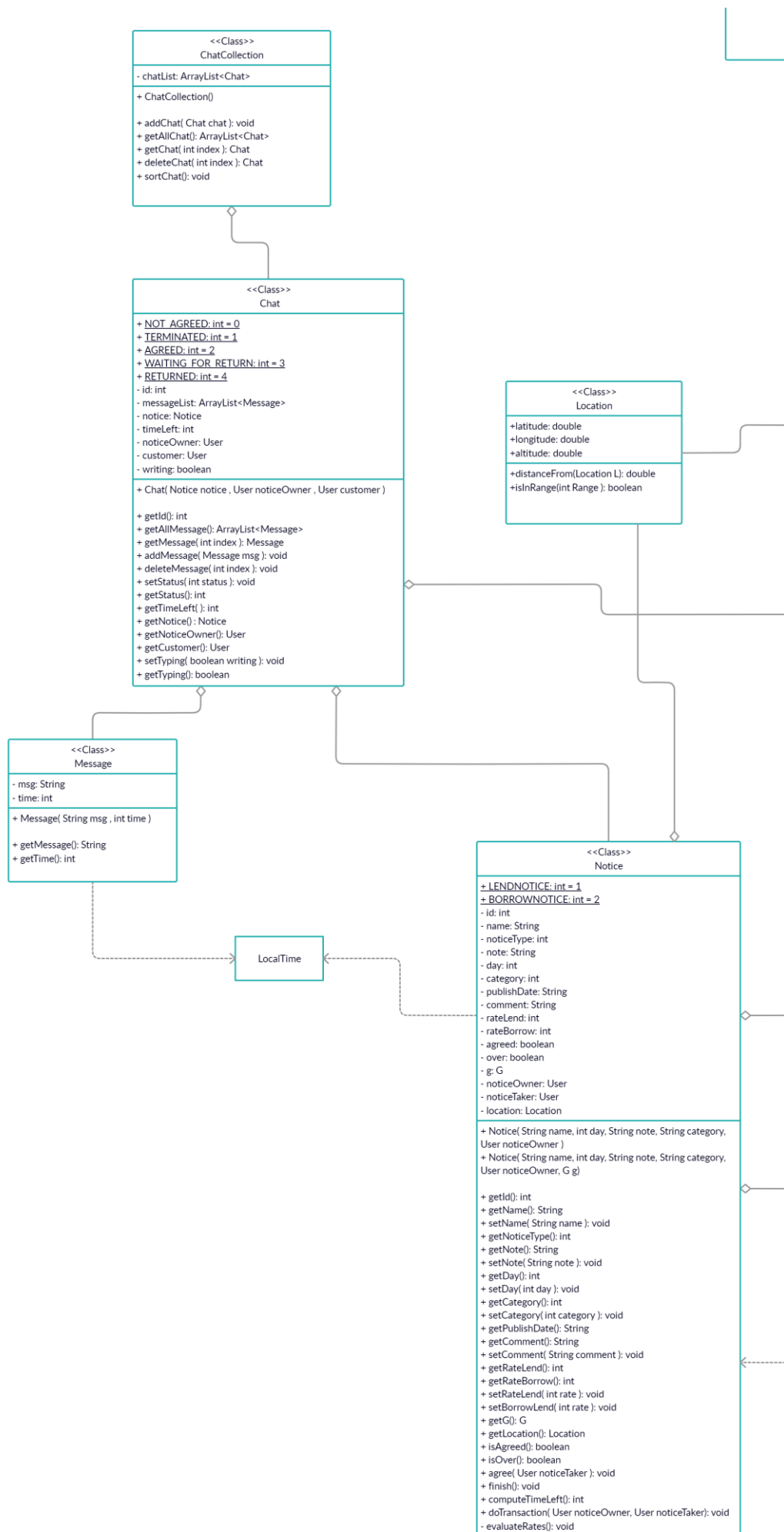
Our application will be an android application that will be written in android studio. Our application will have a database and our database will be created with MySQL application. We will also use a server for containing the database. We may use socket.io to program the chat of the application. We will also use github for version controlling and sharing codes. As it is the primary programming language that we learned, our code will be written in mostly java.

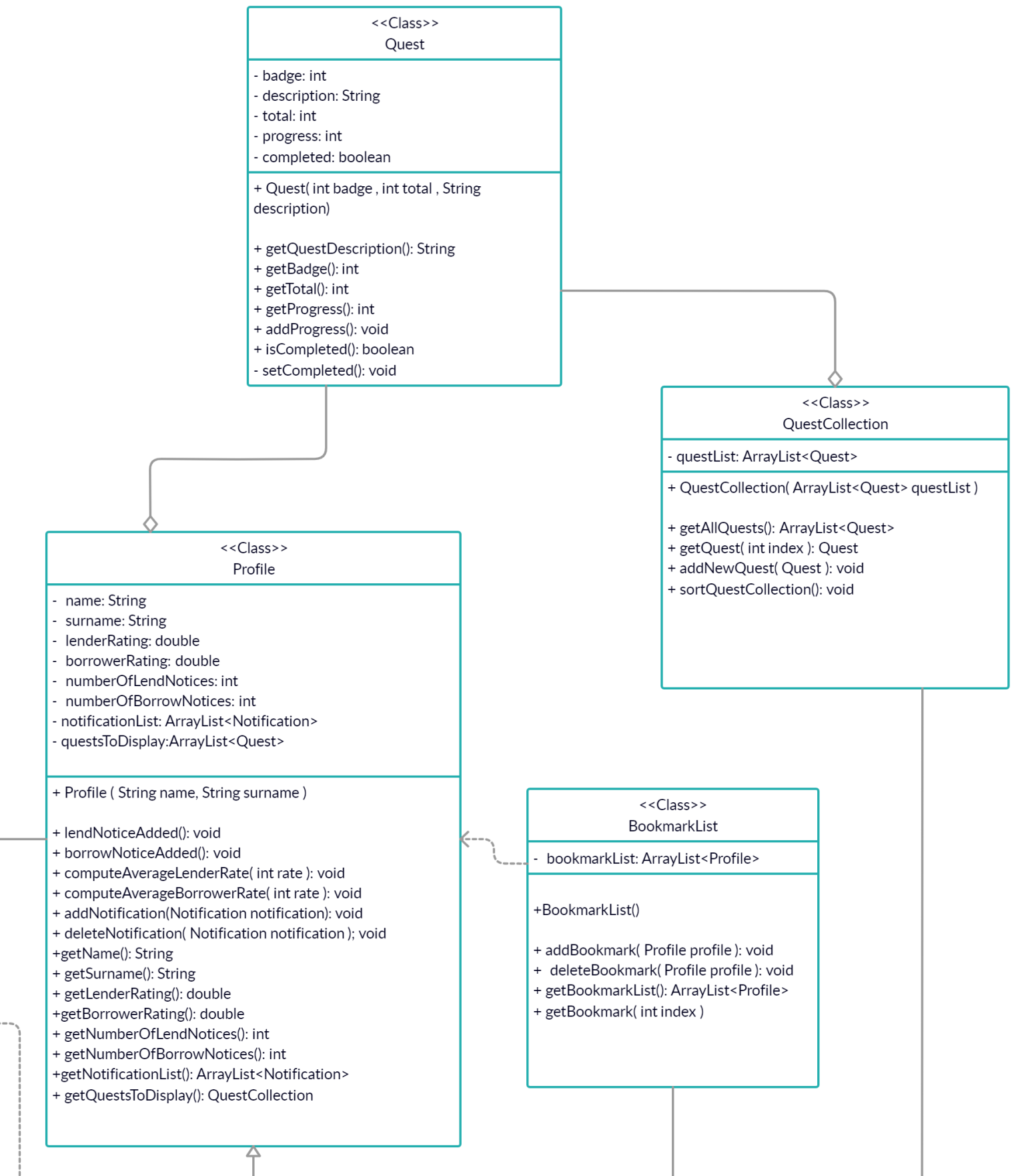
2.3 UML Diagrams

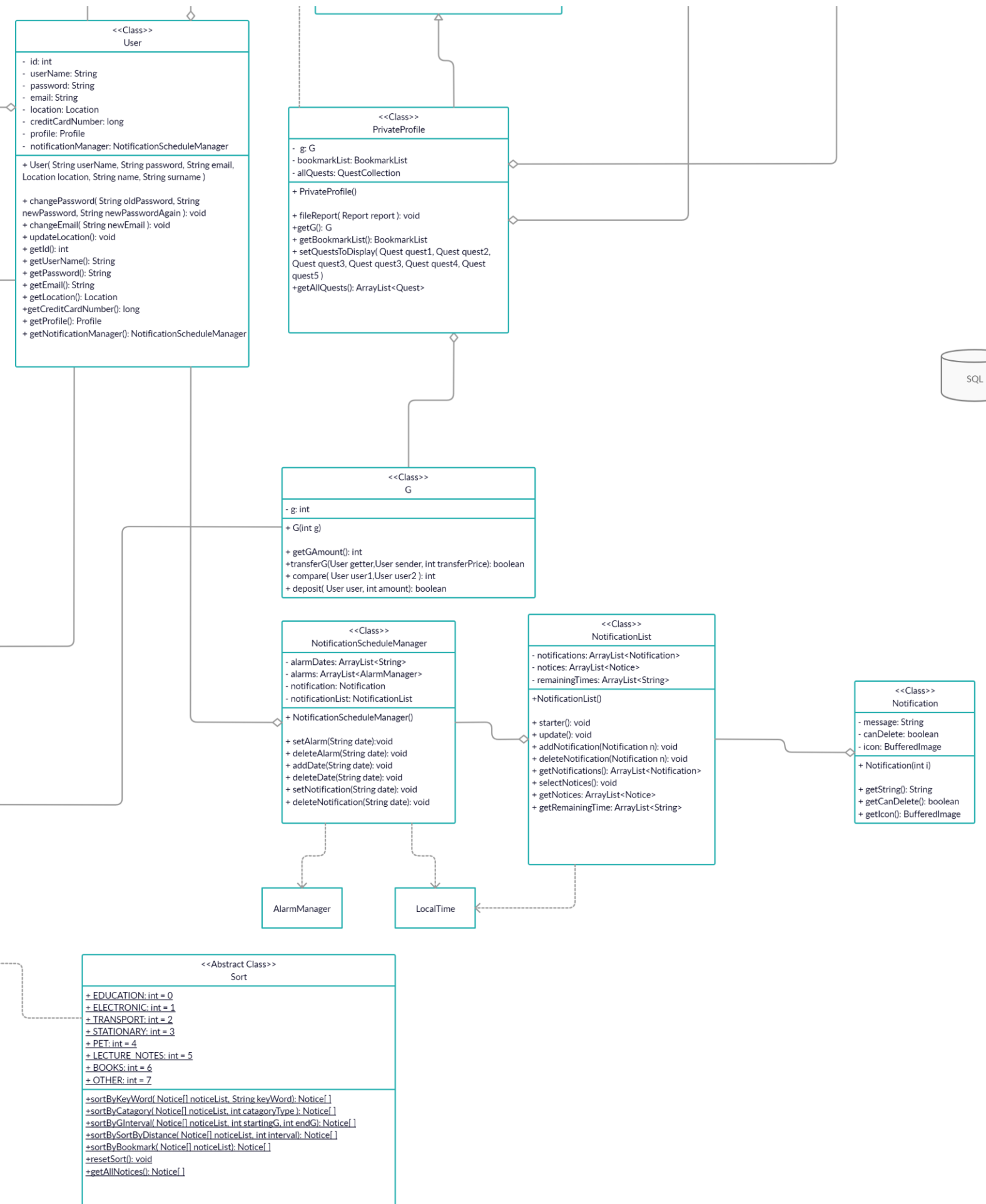


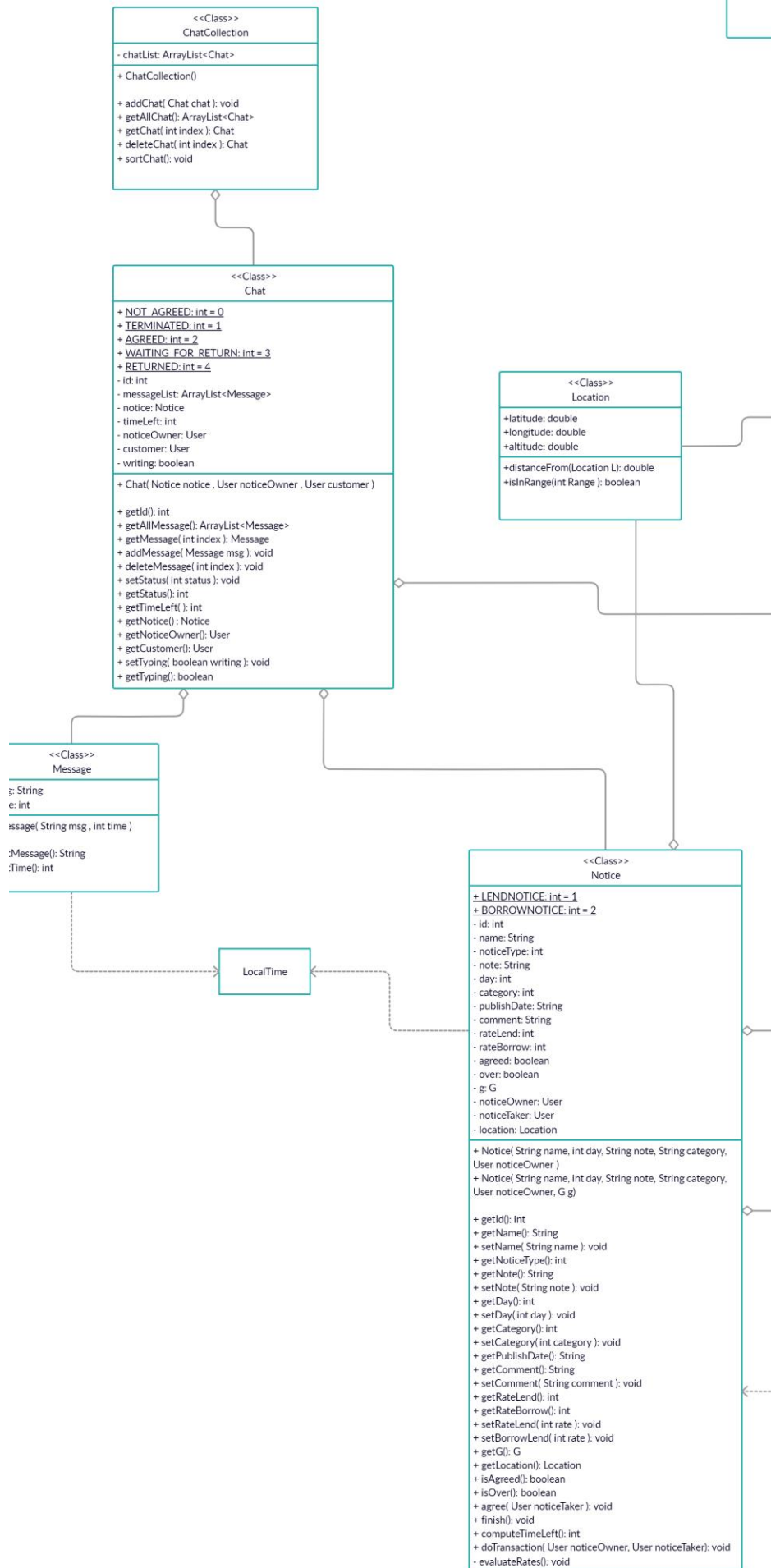


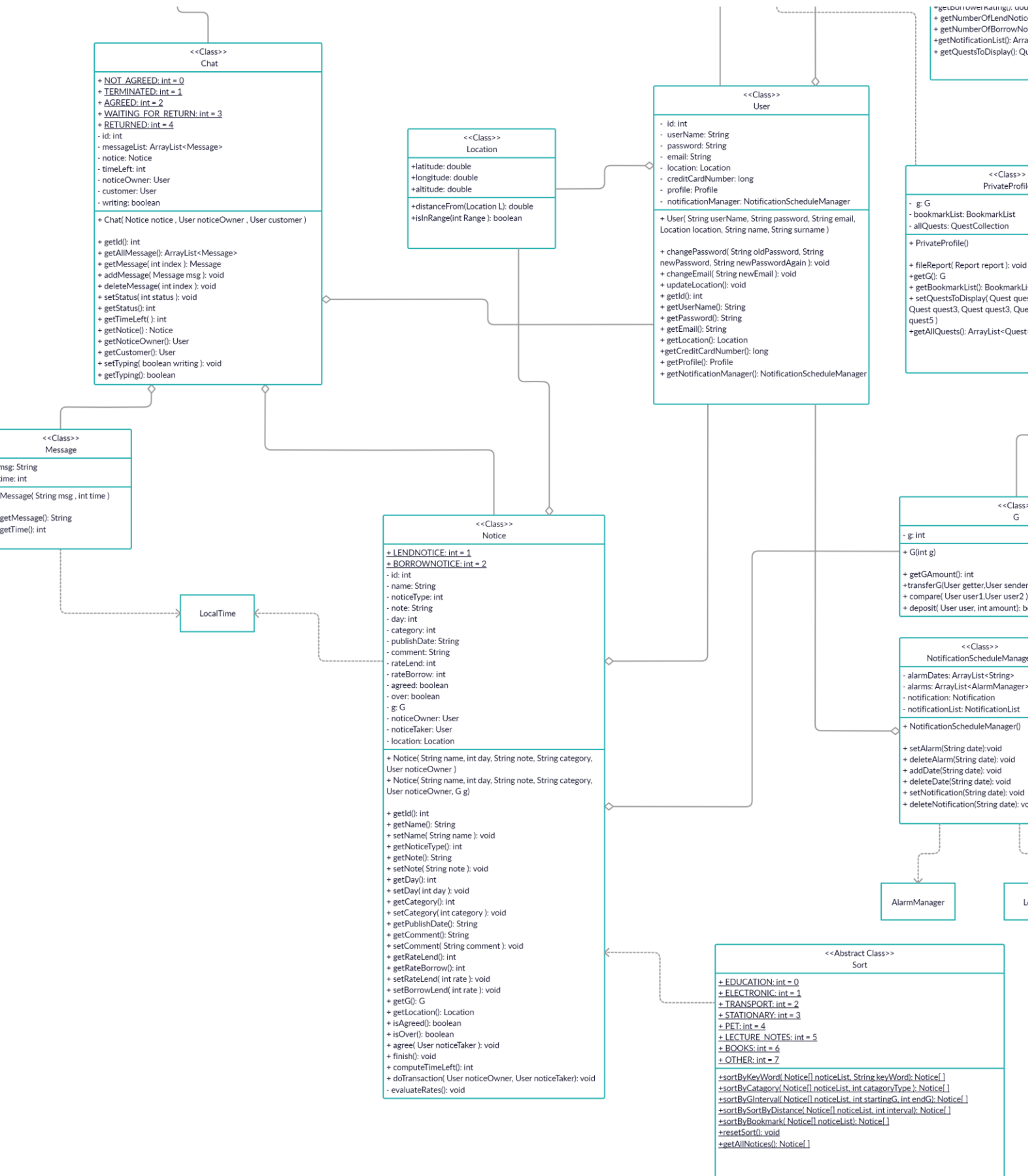


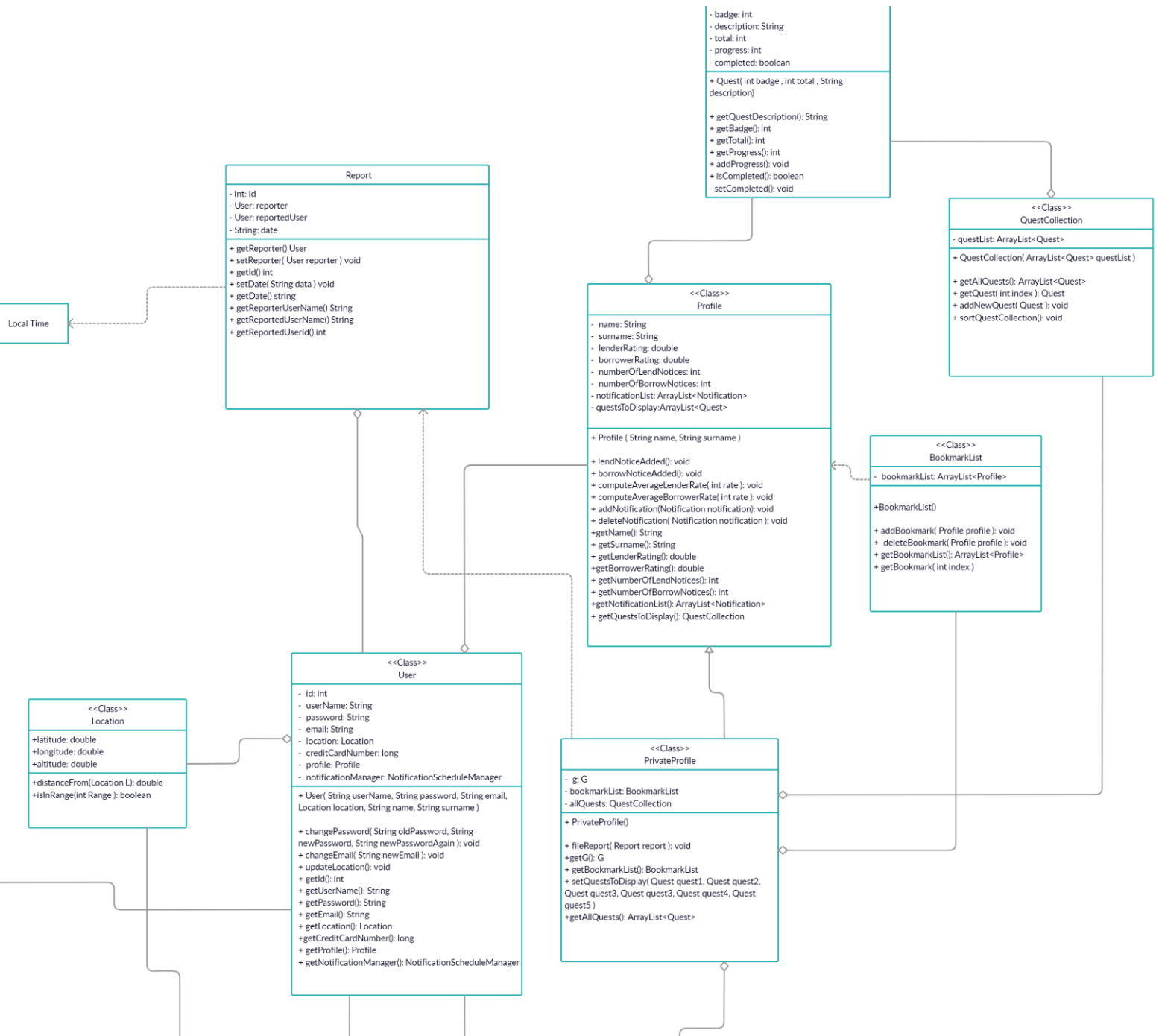


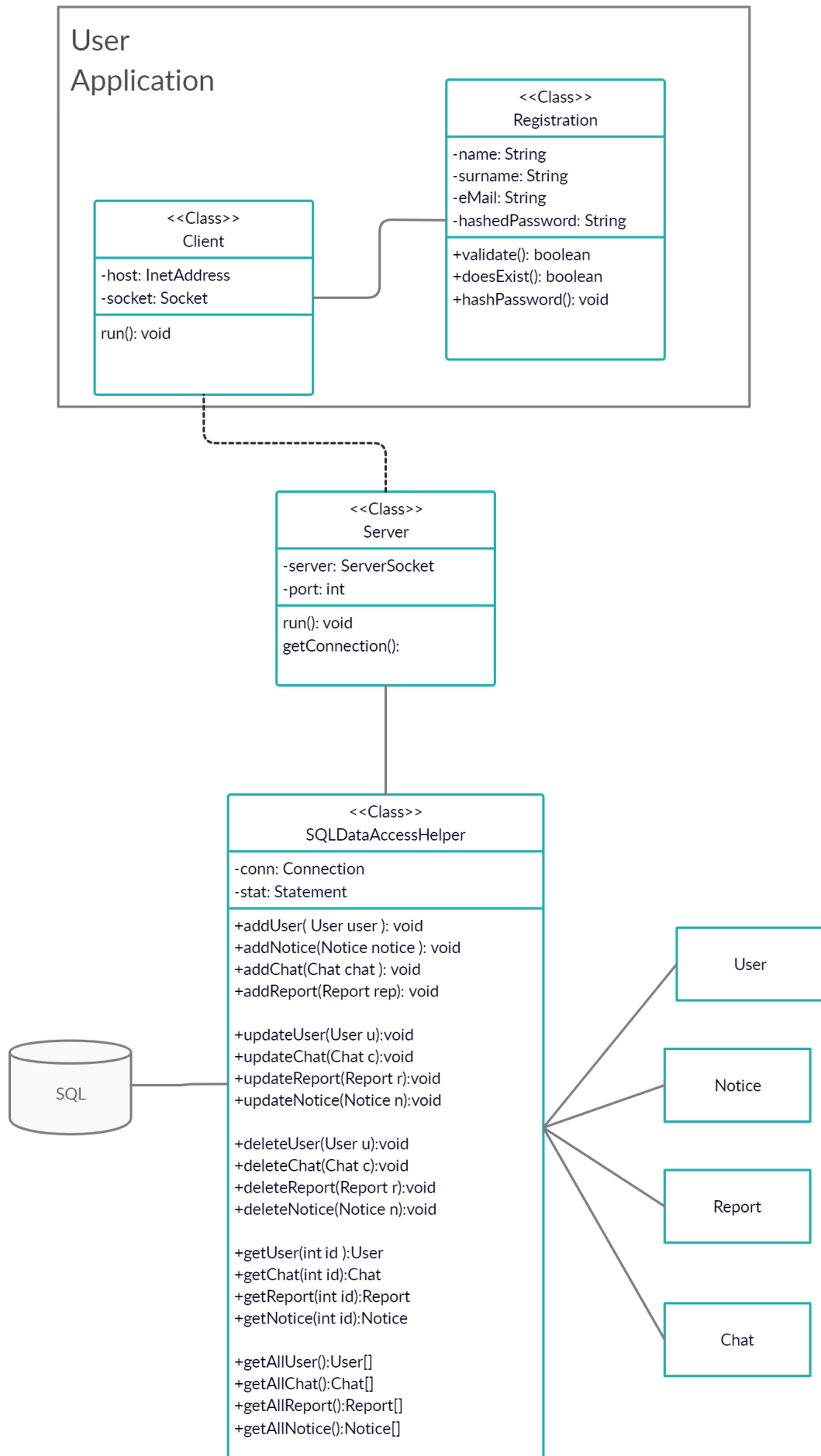












2.4 Task Assignment

We will separate the group for the sharing of the work as:

Kaan Ateşel will do the database part

Tuna Dalbeler will do the server part.

Onur Korkmaz will do the controller part.

Güven Gergerli and Çağrı Eren will do the modelling and Viewer part.

Bora Çün will do the testing part.