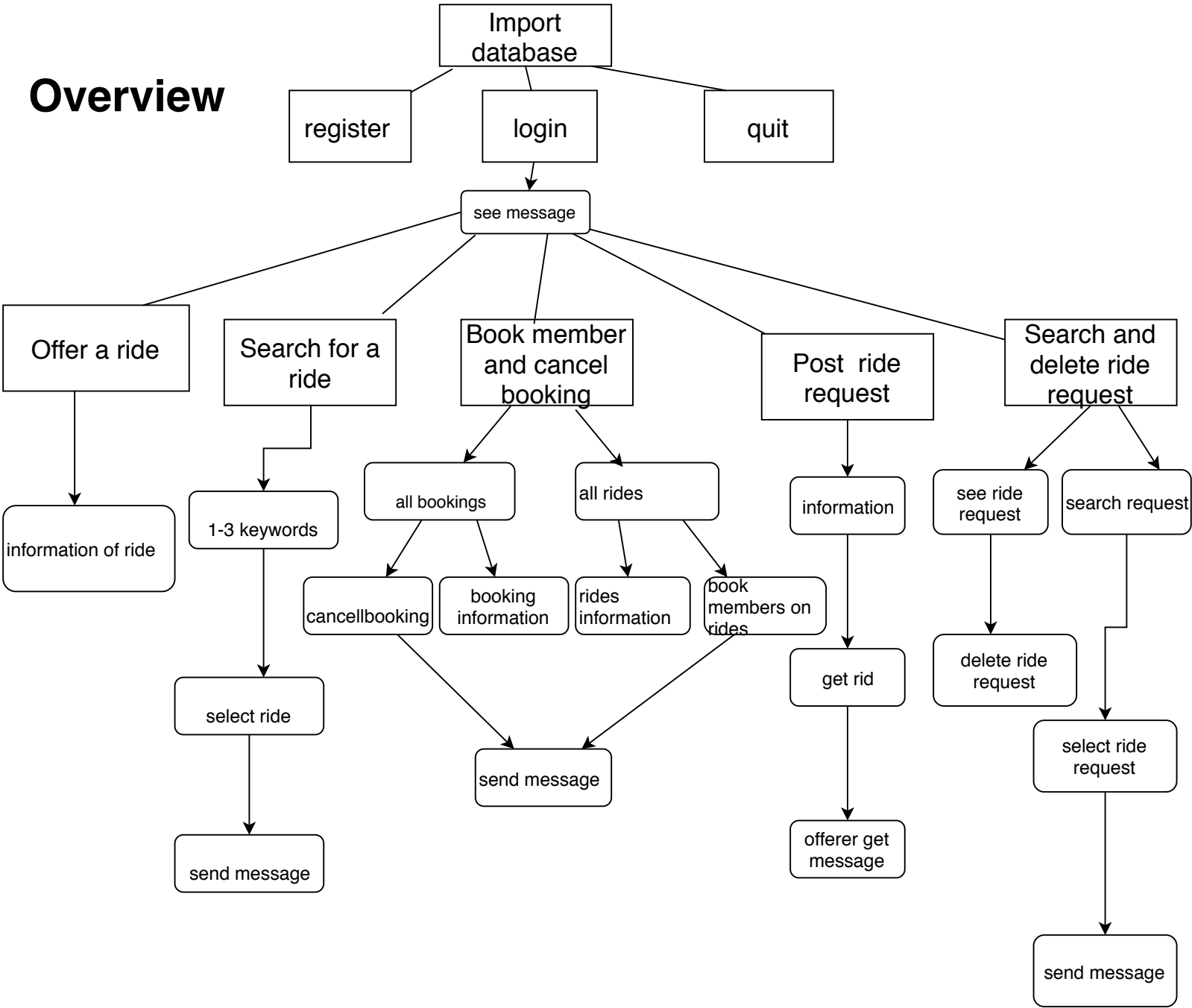


Overview



CMPUT 291 Mini project1

Overview

Our group project is designed for users to offer and book rides. Users just need to follow the instructions to choose which function you want to use.

Login

When user enter the system, there will show a login interface. If it is your first time to use our system, you should register first by enter "r". Offer unique email, name, phone and password to finish signup. If you have already registered before, then choose login by entering "l". Providing valid email and password and you login our system successfully. After log in, user can see unread messages. User also can logout by entering "q" to exit the program.

Offer a ride

User can offer ride by providing a date, the number of seats offered, the price per seat, a luggage description, a source location and a destination location. user also have the option to add a user's car number and any set of enroute locations.

Search for rides

Users can search rides by entering 1-3 location keywords. And you can select a ride and send messages to the one who post ride.

Book members or cancel bookings

Users who post the ride can see all the bookings, available seats on all rides and can cancel any bookings. For users whose bookings have been cancelled will get a message. User who offer rides can add user who send booking request with necessary booking information. However, users will get warning if the ride is overbooked.

Post ride request

For users who want to book a ride can search their request with valid booking information. After post ride request, user will get unique rid number. User who offer ride can get request.

Search and delete ride requests

For the users offer ride can see all the ride requests sent to you and can delete any of them and they can also send message to the one who post request.

Detailed design

Our main project document is project.py which connect database and contains the login function. Other 5 documents which are offer.py, search.py, book.py, post.py, requests.py have been imported in project.py. Each document attribute to one option in the project.

In project.py, our main function is used to ask developer to enter the name of database and import it. Welcome_screen function used to show user interface to login, register or exit.

Sign_up function is used to register by providing valid email, name, phone and password.

Log_in function for user to log in with valid email and password. Check_email and check_log

function are used to check if the log in email and password have existed in database. Menu

function show 6 options interface for user to enter. If user have unseen message, mail function show them after login successfully.

Offer.py is used for 1. offer a ride. Offer function ask user to input valid information, and select matching location code from database. Car function is an option for user to choose whether they want to enter car number.

Search.py represent 2.search for rides. Search function ask user to enter 1-3 keywords and find the match rides from database. Select function ask user to select which ride they want to send message.

Book.py is option3. List_bookings function is used for list all the bookings and can cancel any booking. List_rides function can list the ride that user offer and book members who send request to.

is represent 4.Post ride request option. Post function ask user to enter valid request information including date, pick up location, drop off location, price that they want to pay and insert information into request database. Check_valid is used to check whether all the input information are valid otherwise invalid message will show.

Requests.py represent 5.Search and delete request. Search_delete_requests function can show all the ride requests or select a request that want to delete. Search_request function search request by matching lcode entered by user and user can send message.

Testing Strategy

Our testing strategy generally is import the database provided on eclass. For example, we use mess@marky.mark and password mpass to login. And we want to test task 5 search and delete request. We found there are two requests which email is mess@marky.mark After test 5, we can see whether they show all of these two requests. And after delete one request , is there only one left?

Every time we ask user to input information, we have to check whether input is valid. If it is invalid, the invalid message will show and re-input. At first, we don't add check_valid function, so when testing, there exist foreign key constraint failed message. This is our common bugs when we test.

Another specific test case example is after send a request and post a booking, user will get an unique rno or bno. We at first use count the whole bookings and requests number and then +1 to get unique number. However when testing, there will show unique constraint failure. Because we didn't consider user can cancel requests and bookings. So we change our method to debug.

In second question: search for rides testing. We have to test all kinds of input keywords, like two or three same keywords, two or three different kinds of keywords like one is lcode, one is date... If we get the correct output according to database.

We also need to check if we send message successfully. We use one member to login, test one function like book member and send message to the one. And we login another user to see if get message successfully including all information from inbox table.

Group Work Strategy

Our group is made up by 3 members: Ziyuan Liu (zl8), Jianfei Ma (jianfei2), Qi Yang(qy5). Each member in our group lift their weight, try best to finish this project and help each other. Ziyuan is responsible for login, 1.offer rides, and 2.search for a rides. Jianfei is responsible for 3.Book member or cancel bookings and 4. Post ride requests. I am responsible for 5.search and delete rid requests and report documents.

Our group spent around 3 days for this project. The first day we allocated every member's task and start to work on it. The second day we discuss the problems we meet when we do tasks and work on it to fix them. I started to write document. The third day we still fix some problems. And we get our project together, test them then hand in finally.

Here are problems we met:

In first task, Ziyuan thought show 5 matches at a time means it show 5 matches at the first time and if user want to see more, the all the matches can show. But I think it means every time user can only see 5 matches until all the matches shown. After discussed, we think my understanding is correct.

In the second task search for rides, we found that after entering 2 or 3 keywords, all the matches should display. However if one match contain both these keywords, so it will display more than just once. After discussing, we think we should remove duplication by splitting in three situations.

In third task, Jianfei understand that if user who offer ride can book member who have sent the message rather than just random users after discussion.

In forth task, Jianfei didn't check the valid at first. So if user enter the invalid information, the project will crash. So he added it after discussion.

In fifth task, I don't know how to do show 5 matches each time at first. Because Ziyuan had same problem, so we work together to fix both of us problems.

For login function, I found out we should make password non-visible and we should ask for the name of database to import database rather than import in main function. So we still can run our project without changing code. We also fix these two problem together.

The method of coordinate is we did our own task separately at first. When we have problems, we will discuss in group to solve them together. After finish, we test together to handle different situations as much as possible.