



CSE411

Software Engineering and Information System Design

Project Report

Project Topic: Car Management Service

Project member:

Md. Mihal

2018-1-60-020

Md. Muyeen – Ul – Islam

2018-1-60-022

Arina Islam Ahona

2018-1-60-179

Submitted to:

Dr. Mohammad Salah Uddin

Assistant Professor, Department of Computer Science & Engineering,

East West University, Dhaka, Bangladesh.

Introduction

Cars are one of the most used transportation systems in the whole world. In Bangladesh, there are 3419884 registered motor vehicles, and a huge portion of it are cars and the number is growing. We use cars as our private and rental transportation services. But to use this huge market, a customer often faces many hurdles to rent or buy a car. Because of this, an effective one stop “Car Management Service” can solve many problems such as finding the best car in range, rent a suitable car, having an option to find the best suitable car for a person, and the latest news about the car world.

Problem Statement and Solution

Buying and renting cars by going to the places physically is sometimes a hassle. It would be much easier if booking cars for buying and renting could be done online. Our Car management System will be a one stop solution where an individual can buy a car, rent a car for temporary use, find the latest update of different cars online. There will be three kinds of users, admin and registered users and guest users. Guest users will have the permission to view the cars, get the latest information, find suitable cars by sorting them and contact the admin for any queries. Registered users will be able to create and login via registration page, find details about a car and book it, buy any car as per choice, check booking status, manage his/her account and update it and send message to the admin. Registered users will also have the permissions of a guest user. Finally, admin will have all the previous permissions and more, such like admin login, add and manage new vehicles, edit the details of a vehicle, manage booking details, manage registered users, update any page content, add new cars for sell, view transactions for both buying and renting. Admin will have a dashboard that will allow him/her to manage all these tasks.

Feasibility Study

1. **Technical feasibility:** this project is technically feasible. To create this software we will mainly use PHP, MySql and HTML. JQuery, Javascript and bootstrap will be used if needed. For hosting we will use Xampp server. All the tools we are using is easily accessible and well within our budget. By using these tools we can ensure the customer requirements within budget and time.

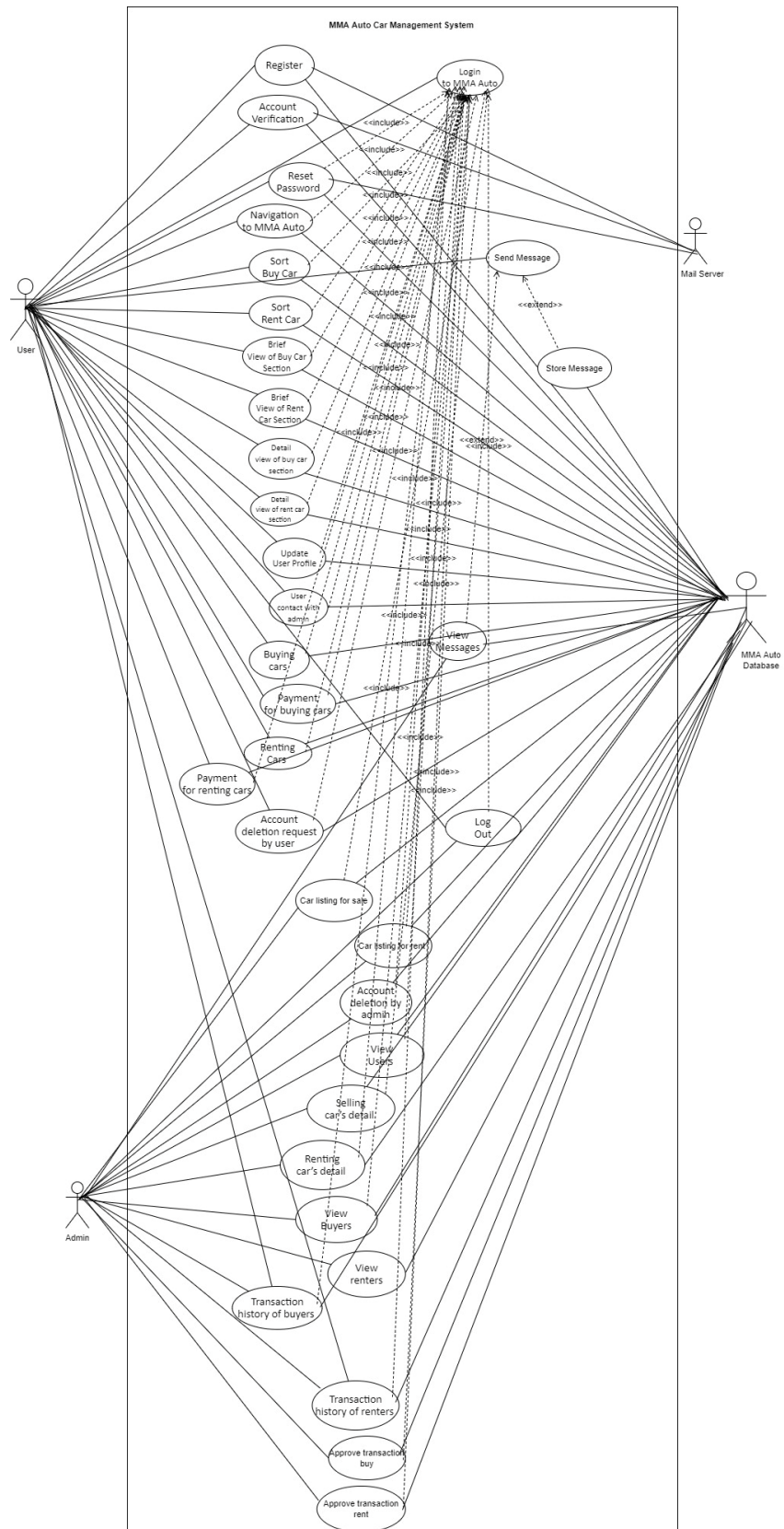
2. **Operational feasibility:** this project will allow the guests to simply register to the website and the registered using can easily book cars for buying and renting and also can make payment for the car easily. The software will also help the admin to efficiently manage all the tasks from the website. The sensitive data are also encrypted which will ensure the security of the users and admins. The project is user friendly and even a novice can use the website without any prior knowledge.
3. **Economical feasibility:** this project is economically feasible. The admin can easily handle all the tasks from the website which will reduce the amount of manpower needed to manage the system. All the data are stored in a database so there will be no need to keep any recording books which will reduce the amount of effort needed to keep record and also no manpower needed. So, these will save costs from various angles. That being said, this project is economically feasible.

Requirement Analysis:

1. Vehicle information for guest users.
2. Contact/Query section for guest users.
3. User registration/sign up option.
4. User login option.
5. Buying option of listed cars for registered user.
6. National ID and phone number input for registered user while confirming a transaction.
7. Booking a car for rent option for registered user.
8. Cancel booking anytime until the admin approves the request for registered user.
9. Registered user can sort cars by price/name.
10. Registered user can view their profile.
11. Registered user can update their existing information.
12. Registered user can change their password.
13. Password encryption security for registered user.
14. Registered user can send a request to permanently delete his/her account.
15. OTP verification for registered user.
16. Registered user can send direct message to admin.
17. Registered user can ask any question.

18. Register user can file a complaint.
19. When buying a car, registered user can see the stock if that is very low.
20. Admin login option.
21. Admin can see registered users.
22. Admin can delete the id of a particular user.
23. Admin can see listed cars.
24. Admin can update the information of an existing car.
25. Admin can see the bookings.
26. Approval system of bookings for admin.
27. Admin can see number of queries.
28. Admin can answer the questions asked by a registered user via email.
29. Admin can verify the bookings.
30. Admin can see the booking history.
31. Admin can update the contact information section.

Use Case Diagram:



Use Case Description:

Use case #1	Register	
Goal in Context	This use case allows new user to register to MMA auto in order to get the most benefit by creating an account	
Preconditions	The server is up and is fully functioning	
Success End Condition	User successfully creates their account	
Failed End Condition	User was unable to create an account by not giving necessary information	
Primary Actors	User, potential buyer or renter or both	
Secondary Actors	The database where information is being stored	
Trigger	When user clicks the register button and navigates to register page	
Description	Step	Action
	1	User will have to navigate to register page
	2	User will have to fill the form with their name, email, password
	3	Database will store the information
	4	A verification mail will be sent to users given mail
	5	When verified user will be asked to Login using the credentials
Extensions	Step	Branching Action
	1	Invalid information to fill up the form

Use case #2	Account Verification	
Goal in Context	This use case allows a user to verify their account creation in order to be successfully login to MMA auto by clicking the confirmation link sent to their email	
Preconditions	User must Sign up and the server should be functioning properly	
Success End Condition	User successfully verifies their account creation	
Failed End Condition	User wasn't successful to verify their account	
Primary Actors	Users who want to get registered	
Secondary Actors	Database and auto mailer	
Trigger	User fills the register form properly and clicks the register button	
Description	Step	Action
	1	User fills the registration form and clicks the register button
	2	A message shows that a verification link has sent to the users email that they are using for registration
	3	User navigates to his email account and clicks the confirmation link sent by MMA auto
	4	If account verification is successful user will be redirected to login page and success message shows up
Extensions	Step	Branching Action
	1	User unable to find the confirmation otp link containing email

Use case #3	Login to MMA Auto	
Goal in Context	User can Login to their account in order to buy or rent cars	
Preconditions	User have to create an account first and server should be up and running	
Success End Condition	User successfully logs in to their account	
Failed End Condition	User was unable to log in to their account due to incorrect credentials given by them	
Primary Actors	Registered user, potential buyer or seller both	
Secondary Actors	MMA Auto Database	
Trigger	When a user clicks on the login button after they have registered successfully	
Description	Step	Action
	1	User will go to the login page after registering in MMA Auto
	2	User inputs his email and password that were used to create his account
	3	User clicks the login button
	4	Database checks the information given if they are correct or wrong
	5	User is redirected to MMA Auto homepage if login credentials are right
Extension	Step	Branching Action
	1	Invalid information given when login to MMA Auto

Use case #4	Reset Password	
Goal in Context	User can reset their password in case they forget it by clicking the Forgot Password option when Login to MMA Auto	
Preconditions	User must be registered and have an account in MMA Auto so that there are existing email of that user the they used to create the account	
Success End Condition	User successfully reset their password	
Failed End Condition	User was unable to reset their password and could not login to MMA Auto	
Primary Actors	User who forgets their password	
Secondary Actors	Database, auto mailer	
Trigger	User clicks the forgot password option	
Description	Step	Action
	1	User navigates to the Forgot Password option
	2	User inputs their email
	3	Database checks the given email if it exists or not. If any random email is given that was not used to create the user account, a message will show up to state that email does not exist
	4	If correct email is given a verification will sent to the email and a notification message will show up
	5	User navigates to their email and click the verification link
	6	User redirected to change password page where they are asked to enter a password and confirm that
	7	If password and confirm password matches password will be changes successfully and redirects to Login page
Extension	Step	Branching Action
	1	Error message shown for random email input
	2	Error message shown for unmatched password

Use case #5	Navigation to MMA Auto	
Goal in Context	User can navigate in MMA Auto very easily by clicking the specific options provided in the navigation bar	
Preconditions	The server is up and running and user have to visit the homepage of MMA Auto	
Success End Condition	User successfully browses each section of MMA Auto car management service	
Failed End Condition	Navigation menu fails to navigate user in different sections	
Primary Actors	User, potential buyer or renter or both	
Secondary Actors	HTML structure	
Trigger	User clicks to different options for example home, about, buy, rent, login, logout. Profile	
Description	Step	Branching Action
	1	User navigates to the homepage of MMA Auto
	2	User clicks the option that will take them to that specific section
Extension	No extensions	

Use case #6	Sort Buy Car	
Goal in Context	User can sort the cars according to price of the car (increasing or decreasing order) and Name of the car(ascending or descending order) in case of buying any	
Preconditions	Database should be functional	
Success End Condition	User sorts the cars according to their choice and the sorted result displayed properly	
Failed End Condition	Sorting does not work for user	
Primary Actors	User, potential buyer	
Secondary Actors	Database	
Trigger	User navigates to the buy car section and selects their choice of sorting and clicks the sort button	
Description	Step	Action
	1	User navigates to buy car section
	2	User select the option they want to sort the cars to
	3	From Name(A-Z), Name (Z-A), Price(Increasing), Price(Decreasing) options user select an option and click sort button
	4	Sorted cars are displayed according to choice
Extension	No Extensions	

Use case #7	Sort Rent Car	
Goal in Context	User can sort the cars according to renting per hour price (increasing or decreasing order) and Name of the car (ascending or descending order) in case of buying any	
Preconditions	Database should be functional	
Success End Condition	User sorts the cars according to their choice and the sorted result displayed properly	
Failed End Condition	Sorting does not work for user	
Primary Actors	User, potential renter	
Secondary Actors	Database	
Trigger	User navigates to the rent car section and selects their choice of sorting and clicks the sort button	
Description	Step	Action
	1	User navigates to rent car section
	2	User select the option they want to sort the cars to
	3	From Name(A-Z), Name (Z-A), rent per hour price(Increasing), rent per hour price(Decreasing) options user select an option and click sort button
	4	Sorted cars are displayed according to choice
Extension	No Extensions	

Use case #8	Brief View of Buy Car Section	
Goal in Context	User can have a quick look at the specification of a car they want to buy in floating product card format of HTML and CSS	
Preconditions	Database should be up and running and proper information of a car should be stored in the database	
Success End Condition	User gets a quick overview of the car specification that might interest them to buy it	
Failed End Condition	Information from the database did not show up	
Primary Actors	User, potential buyer	
Secondary Actors	Database, Admin who entered the specification of the car	
Trigger	User gets a look at the specification of cars just under the sorting option as soon as they visit the buy car section of the homepage	
Description	Step	Action
	1	User visits the homepage
	2	User navigates to buy car section
	3	Cars available to buy are displayed in separated tiles
Extension	Step	Branching Action
	1	Specifications of cars those are stored in database are shown
	2	If there is any empty field of specification that will now show in the card tiles

Use case #9	Brief View of Rent Car Section	
Goal in Context	User can have a quick look at the specification of a car they want to rent in floating product card format of HTML and CSS	
Preconditions	Database should be up and running and proper information of a car should be stored in the database	
Success End Condition	User gets a quick overview of the car specification that might interest them to buy it	
Failed End Condition	Information from the database did not show up	
Primary Actors	User, potential	
Secondary Actors	Database, Admin who entered the specification of the car	
Trigger	User gets a look at the specification of cars just under the sorting option as soon as they visit the rent car section of the homepage	
Description	Step	Action
	1	User visits the homepage
	2	User navigates to rent car section
	3	Cars available to rent are displayed in separated tiles
Extension	Step	Branching Action
	1	Specifications of cars those are stored in database are shown
	2	If there is any empty field of specification that will now show in the card tiles

Use case #10	Detail view of buy car section	
Goal in Context	User can have a detail look at all the specification of a car they want to buy in an organized manner when they click on the image of the car or the buy now button	
Preconditions	Server must be functional and user must be registered and logged in	
Success End Condition	User gets a full look at all the specification of a car they want buy or explore	
Failed End Condition	Information from the database did not show up	
Primary Actors	Registered user, potential buyer	
Secondary Actors	Database, Admin who entered the specification of the car	
Trigger	User clicks on the image of the car they want to buy or clicks the buy now button for that specific car	
Description	Step	Action
	1	A registered user logs in to their account
	2	Logged in user goes to the buy now section
	3	User clicks on the image of the car they want to buy or clicks the buy now button
	4	User redirected to another page that contains all the information for that car.
	5	User can scroll to see all the details specifications of that car
Extension	No extensions	

Use case #11	Detail view of rent car section	
Goal in Context	User can have a detail look at all the specification of a car they want to rent in an organized manner when they click on the image of the car or the buy now button	
Preconditions	Server must be functional and user must be registered and logged in	
Success End Condition	User gets a full look at all the specification of a car they want rent or explore	
Failed End Condition	Information from the database did not show up	
Primary Actors	Registered user, potential renter	
Secondary Actors	Database, Admin who entered the specification of the car	
Trigger	User clicks on the image of the car they want to rent or clicks the rent now button for that specific car	
Description	Step	Action
	1	A registered user logs in to their account
	2	Logged in user goes to the rent now section
	3	User clicks on the image of the car they want to rent or clicks the rent now button
	4	User redirected to another page that contains all the information for that car.
	5	User can scroll to see all the details specifications of that car
Extension	No extensions	

Use case #12	Update User Profile	
Goal in Context	User can update his profile by changing username and password	
Preconditions	User must be registered and logged in and server must be up and running	
Success End Condition	User successfully change their credentials	
Failed End Condition	Credentials were not changed successfully	
Primary Actors	Registered user	
Secondary Actors	Database	
Trigger	A registered user logs in to their account and navigates	
Description	Step	Action
	1	User logs in to their account
	2	Logged in user navigates to profile menu
	3	Options are available for changing the username and password
Extension	No extensions	

Use case#13	User contact with admin	
Goal in Context	Any user (registered or non-registered) can send any message to admin via contact option	
Preconditions	Database server must be up and running	
Success End Condition	Admin successfully received user's message and success message showed up	
Failed End Condition	Message was not sent successfully due to server issue	
Primary Actors	User	
Secondary Actors	Database, admin	
Trigger	User scrolls down to contact option and writes a message and click the submit button	
Description	Step	Action
	1	User goes to MMA Auto homepage and scrolls down to contact option
	2	User writes what they want
	3	User submits their message
Extension	Step	Branching Action
	1	A success message showed up saying that "Message was sent successfully to admin".

Use case#14	View Messages	
Goal in Context	Admin can see the messages sent by the system's user in a detailed manner	
Preconditions	Server must be up and running and Admin must be logged in	
Success End Condition	Admin successfully see's the message sent by user	
Failed End Condition	Admin did not find any messages although message was sent to admin	
Primary Actors	Admin, database	
Secondary Actors	User	
Trigger	Admin logs into dashboard and click the view messages option	
Description	Step	Action
	1	Admin logs into their account
	2	Admin scrolls down to view messages option and clicks it
	3	Messages are shown including the sender's email
Extension	Step	Branching Action
	1	Message delete option available for admin
	2	Admin can click the delete icon to delete the message sent by user
	3	Deleting removes the messages permanently from database

Use case#15	Buying cars	
Goal in Context	User can buy any car they want to	
Preconditions	User must be registered and the car any user wants to buy must be in stock	
Success End Condition	User successfully entered their information and redirects to payment page	
Failed End Condition	User couldn't redirect to payment page	
Primary Actors	Buyer	
Secondary Actors	Database, admin	
Trigger	User clicks on buy now option of the car they want to buy and then clicks proceed	
Description	Step	Action
	1	User logs into their account
	2	User navigates to buy a car section and clicks buy now option of the car they want to buy
	3	User clicks to proceed option if that car is in stock
	4	User enters their information on the client information form, information stored in the database
	5	User is redirected to buy status page where they can pay for the car
Extension	Step	Branching Action
	1	User can watch their buying status
	2	For payment methods, user clicks the proceed to pay button
	3	To cancel order, user click the delete button

Use case#16	Payment for buying cars	
Goal in Context	A registered user can pay for the car they bought in a convenient manner	
Preconditions	Server must be up and running and user must be registered and logged in	
Success End Condition	Successful transaction of payment	
Failed End Condition	Incorrect amount given	
Primary Actors	Buyer	
Secondary Actors	Database, admin	
Trigger	User proceeds to payment or user navigates to buy status from profile option	
Description	Step	Action
	1	User proceeds to payment option after buying a car
	2	User name and national ID is showed from database
	3	User enters the amount to be paid
	4	System checks if the user input amount and amount to be paid are matched or not
	5	Error message shown if incorrect amount is entered
	6	User is thanked by redirecting to another page if transaction is successful
Extension	Step	Branching Action
	1	User can pay by navigating to Buy status from profile option
	2	User can see their buying status

Use case#17	Renting Cars	
Goal in Context	User can rent any car they want to	
Preconditions	User must be registered and the car any user wants to rent must be available	
Success End Condition	User successfully entered their information and redirects to payment page	
Failed End Condition	User couldn't redirect to payment page	
Primary Actors	Renter	
Secondary Actors	Database, admin	
Trigger	User clicks on rent now option of the car they want to rent and then clicks proceed	
Description	Step	Action
	1	User logs into their account
	2	User navigates to rent a car section and clicks rent now option of the car they want to buy
	3	User clicks to proceed option if that car is in stock
	4	User enters their information on the client information form, information stored in the database
	5	User is redirected to buy status page where they can pay for the car
Extension	Step	Branching Action
	1	User can watch their renting status
	2	For payment methods, user clicks the proceed to pay button
	3	To cancel order, user click the delete button

Use case#18	Payment for renting cars	
Goal in Context	A registered user can pay for the car they rent in a convenient manner	
Preconditions	Server must be up and running and user must be registered and logged in	
Success End Condition	Successful transaction of payment	
Failed End Condition	Incorrect amount given	
Primary Actors	Renter	
Secondary Actors	Database, admin	
Trigger	User proceeds to payment or user navigates to rent status from profile option	
Description	Step	Action
	1	User proceeds to payment option after renting a car
	2	User name and national ID is showed from database
	3	User enters the amount to be paid
	4	System checks if the user input amount and amount to be paid are matched or not
	5	Error message shown if incorrect amount is entered
	6	User is thanked by redirecting to another page if transaction is successful
Extension	Step	Branching Action
	1	User can pay by navigating to rent status from profile option
	2	User can see their renting status

Use case#19	Car listing for sale	
Goal in Context	A user specifically only the admin can add cars for sale for the users to buy. Admin can add all the specifications of the car, image of the car and manage the stock	
Preconditions	Server must be up and running and Admin must logged-in	
Success End Condition	Information successfully stored in the database and are shown to the users in buy car section	
Failed End Condition	Information was not stored and no data displayed in the buy car section	
Primary Actors	Admin	
Secondary Actors	Database, potential buyers	
Trigger	Admin logs into his account and navigates to Add a car for sell option	
Description	Step	Action
	1	Admin logs in to the system
	2	Database checks if the user admin or not
	3	Admin navigates to add a car for sell option
	4	All the specifications are inserted by admin including a unique csku
	5	Database checks into the buy_car table if the csku is unique or not, if passes then information is stored into database
	6	Stock quantity entered by admin
	7	Submitting the specifications stored them into database
Extension	Step	Branching Action
	1	Listings are displayed into buy car section on the homepage for users to buy

Use case#20	Car listing for rent	
Goal in Context	A user specifically only the admin can add cars for rent for the users to rent. Admin can add all the specifications of the car, image of the car and manage the stock	
Preconditions	Server must be up and running and Admin must logged-in	
Success End Condition	Information successfully stored in the database and are shown to the users in rent car section	
Failed End Condition	Information was not stored and no data displayed in the rent car section	
Primary Actors	Admin	
Secondary Actors	Database, potential renters	
Trigger	Admin logs into his account and navigates to Add a car for rent option	
Description	Step	Description
	1	Admin logs in to the system
	2	Database checks if the user admin or not
	3	Admin navigates to add a car for rent option
	4	All the specifications are inserted by admin including a unique csku
	5	Database checks in the rent_car table if the csku is unique or not, if passes then information is stored into database
	6	Stock quantity entered by admin
	7	Submitting the specifications store them into database
Extension	Step	Branching Action
	1	Listings are displayed into rent car section on the homepage for users to rent

Use case#21	Account deletion request by user	
Goal in Context	A registered user of MMA Auto can request to permanently delete their account and all their information will be removed from database excluding their transaction history if the have any	
Preconditions	User must be logged in and server must be up and running	
Success End Condition	Account deletion request was successful	
Failed End Condition	Account deletion request was not successful	
Primary Actors	Registered user	
Secondary Actors	Database, admin	
Trigger	User clicks the request to delete id option under the profile menu	
Description	Step	Action
	1	User logs in using their credentials
	2	Database checks the credentials
	3	User navigates to Profile Option
	4	Request to delete ID is available there for the user
	5	Requests are stored in the database for the admin to view and take action

Use case#22	Account deletion by admin	
Goal in Context	Admin can check the deletion requests sent by users and delete their account	
Preconditions	Server must be up and running and admin must logged-in and there should exist a request to delete ID	
Success End Condition	Account deletion successful and a email is sent to the deleted users email to notify	
Failed End Condition	Account deletion was not successful	
Primary Actors	Admin, database	
Secondary Actors	User	
Trigger	Admin clicks the delete button in the action column	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to view deletion requests via dashboard
	4	Clicking on the delete button under action column deletes the account of user
	5	A notification email is sent to deleted users mail about the deletion
Extensions	No extensions	

Use case#23	View Users	
Goal in Context	Admin can see the list of registered users of MMA Auto and manage them	
Preconditions	Server must be up and running and Admin must logged-in	
Success End Condition	Admin successfully watched the list of users and delete any of them if they want	
Failed End Condition	No data showed in the list	
Primary Actors	Admin, Database	
Secondary Actors	User	
Trigger	Admin clicks the see user option from dashboard	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to see user's option via dashboard
	4	All the users are fetched from database and showed on the list
Extension	Step	Branching Action
	1	Admin can delete any user's account by clicking the delete button under action column

Use case#24	Selling car's detail	
Goal in Context	Admin can view the detail information of the cars that are on the selling list and edit or delete any car entry if they want to	
Preconditions	Server must be up and running and admin must logged-in	
Success End Condition	Admin can see the list of cars details that are in sale	
Failed End Condition	No entries are shown in the list although they exist	
Primary Actors	Admin	
Secondary Actors	Database	
Trigger	Admin clicks the view car for sell option from dashboard	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to view cars for sell option via dashboard
	4	All the car's data from buy_car table are fetched from database and showed on the list
Extension	Step	Branching Action
	1	Admin can edit any car's information showed on the list by clicking edit button
	2	Clicking edit takes admin into a from that contains that specific car's information
	3	Admin can delete any car's entry if they want

Use case#25	Renting car's detail	
Goal in Context	Admin can view the detail information of the cars that are on the renting list and edit or delete any car entry if they want to	
Preconditions	Server must be up and running and admin must logged-in	
Success End Condition	Admin can see the list of cars details that are in rent	
Failed End Condition	No entries are shown in the list although they exist	
Primary Actors	Admin	
Secondary Actors	Database	
Trigger	Admin clicks the view car for rent option from dashboard	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to view cars for rent option via dashboard
	4	All the car's data from rent_car table are fetched from database and showed on the list
Extension	Step	Branching Action
	1	Admin can edit any car's information showed on the list by clicking edit button
	2	Clicking edit takes admin into a from that contains that specific car's information
	3	Admin can delete any car's entry if they want

Use case#26	View Buyers	
Goal in Context	Admin can view all the information of the list of buyers	
Preconditions	Server must be up and running and admin must logged-in	
Success End Condition	Admin successfully view the buyers list including their transaction information	
Failed End Condition	No entries are shown in the list although they exist	
Primary Actors	Admin	
Secondary Actors	Database	
Trigger	Admin clicks the view buyers option via dashboard	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to view buyers option via dashboard
	4	All the entries from client_buy table are fetched from database and showed on the list
Extension	Step	Branching Action
	1	Admin can delete any buyers history by clicking the delete option under action column

Use case#27	View renters	
Goal in Context	Admin can view all the information of the list of renters	
Preconditions	Server must be up and running and admin must logged-in	
Success End Condition	Admin successfully view the renters list including their transaction information	
Failed End Condition	No entries are shown in the list although they exist	
Primary Actors	Admin	
Secondary Actors	Database	
Trigger	Admin clicks the view renters option via dashboard	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to view renters option via dashboard
	4	All the entries from client_rent table are fetched from database and showed on the list
Extension	Step	Branching Action
	1	Admin can delete any renters history by clicking the delete option under action column

Use case#28	Transaction history of buyers	
Goal in Context	Admin can see all the transaction made by buyers	
Preconditions	Server must be up and running and Admin must be logged in	
Success End Condition	Admin had a view at the transaction history of buyers	
Failed End Condition	No transaction history showed	
Primary Actors	Admin	
Secondary Actors	Database	
Trigger	Admin clicks the view buy transactions option from dashboard	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to view buy transactions option via dashboard
	4	All the entries from buy_transactions table are fetched from database and showed on the list
Extension	Step	Branching Action
	1	Admin can delete any transaction history of buyers by clicking the delete option under action column

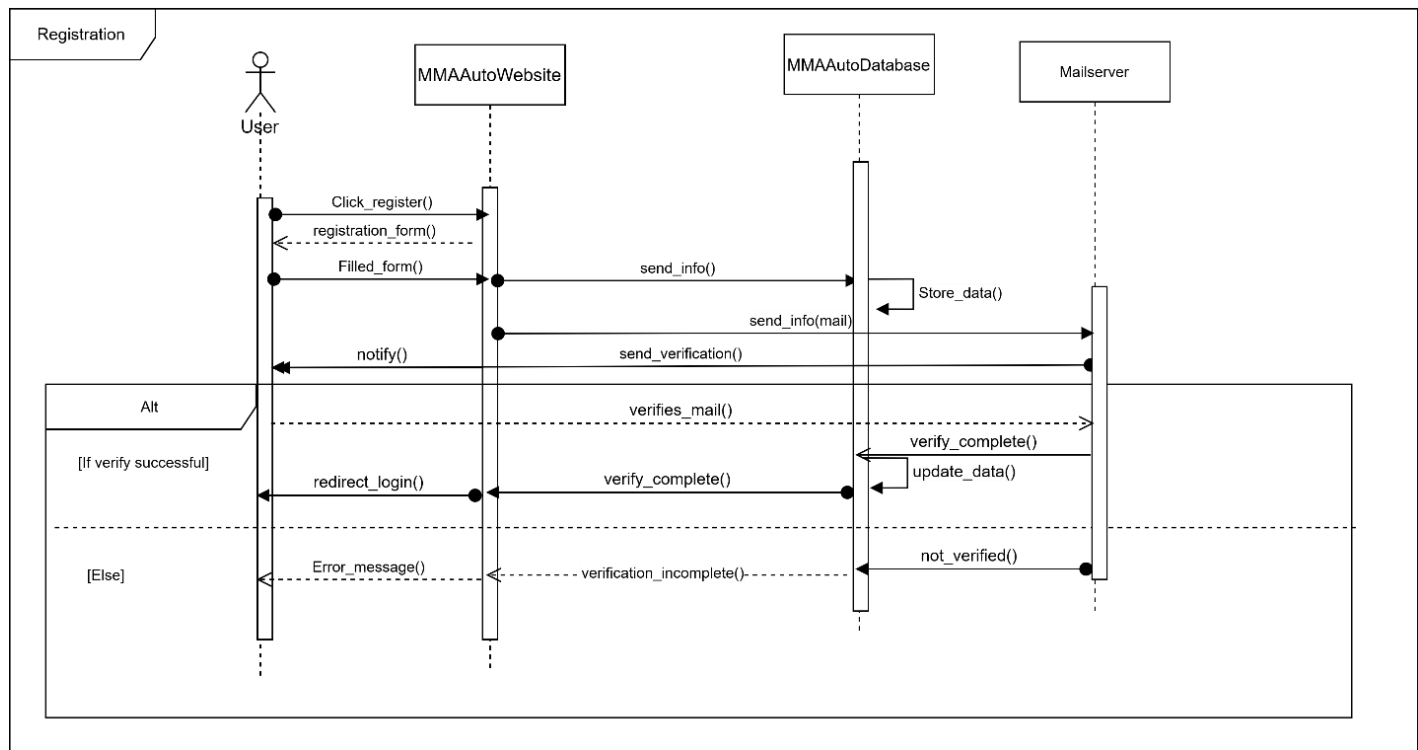
Use case#29	Transaction history of renters	
Goal in Context	Admin can see all the transaction made by renters	
Preconditions	Server must be up and running and Admin must be logged in	
Success End Condition	Admin had a view at the transaction history of renters	
Failed End Condition	No transaction history showed	
Primary Actors	Admin	
Secondary Actors	Database	
Trigger	Admin clicks the view rent transactions option from dashboard	
Description	Step	Action
	1	Admin logs into their account
	2	Database checks the credentials if the user is admin or not
	3	If successfully logged in, admin navigates to view rent transactions option via dashboard
	4	All the entries from rent_transactions table are fetched from database and showed on the list
Extension	Step	Branching Action
	1	Admin can delete any transaction history of renters by clicking the delete option under action column

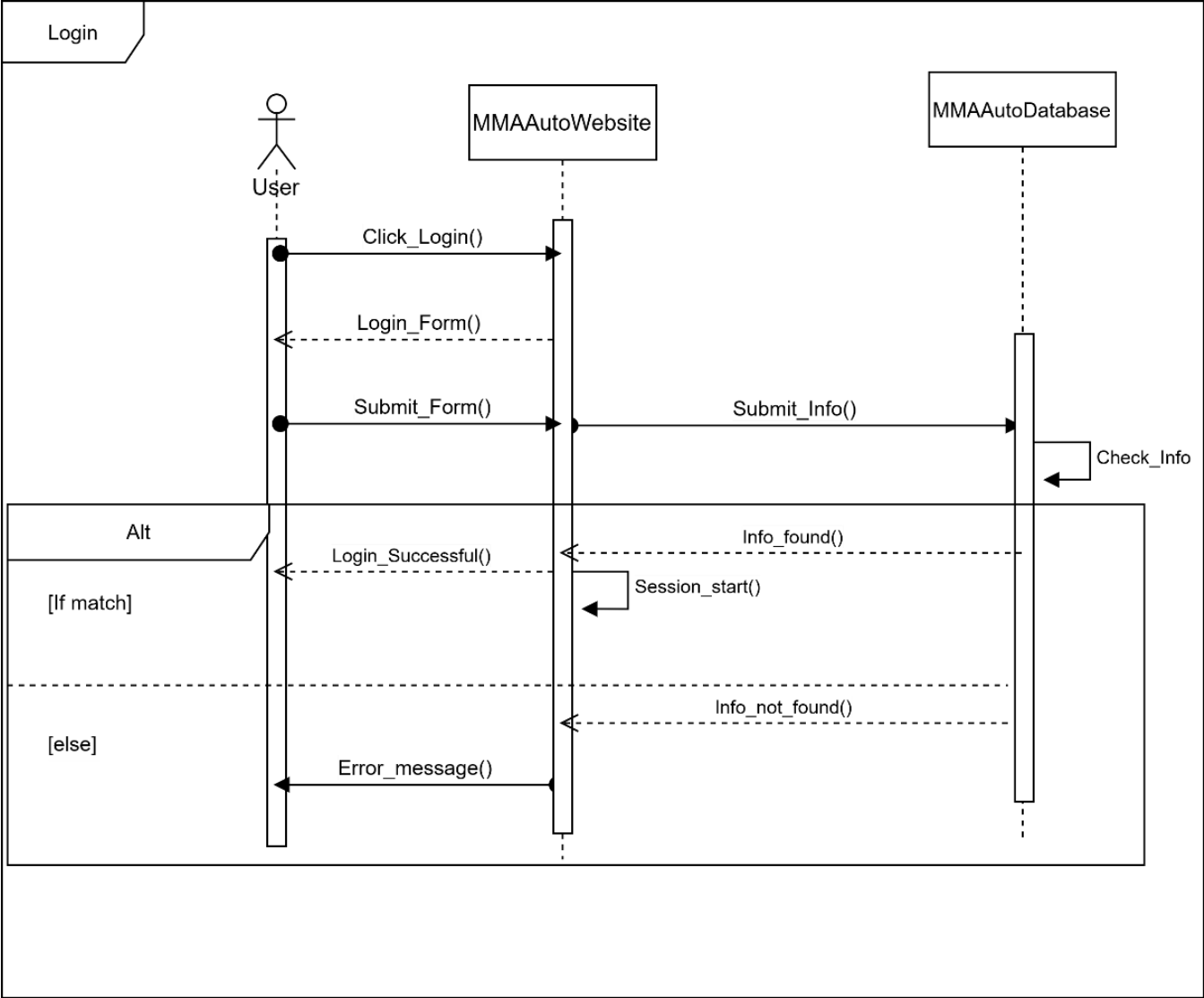
Use case#30	Log Out	
Goal in Context	Both registered user and admin can log out from the system	
Preconditions	User must be logged in	
Success End Condition	User successfully logs out from the system	
Failed End Condition	Log out not successful	
Primary Actors	User	
Secondary Actors	Database	
Trigger	User clicks the logout option	
Description	Step	Action
	1	User navigates to Logout option
	2	User clicks the logout option
	3	Session ends for user
Extension	No Extension	

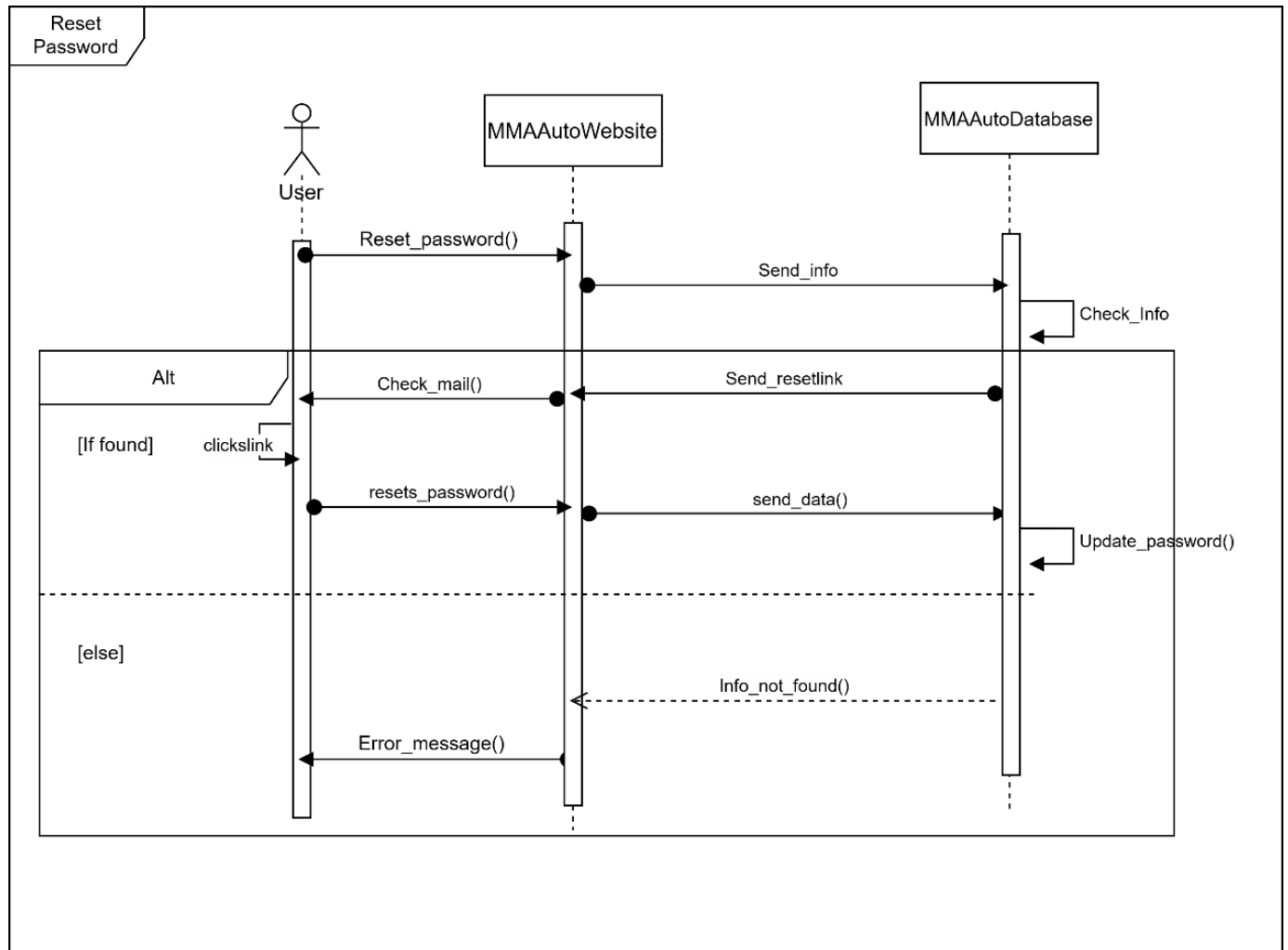
Use case#31	Approve Transaction Buy	
Goal in Context	Admin can confirm any sale by approving buyer transaction	
Preconditions	Server must be up and running and admin must be logged in	
Success End Condition	Transaction successfully approved and buyers can see the approval	
Failed End Condition	Transactions were not approved	
Primary Actors	Admin, database	
Secondary Actors	Buyer	
Trigger	Admin clicks the approve button	
Description	Step	Action
	1	Admin logs into their account
	2	Admin navigates to dashboard
	3	Admin navigates to view buyers option
	4	Admin clicks approve option from the action column if they want to approve transaction of buyers
	5	Database is updated with the action
Extension	Step	Branching Actions
	1	Admin can delete transactions
	2	Buyers can see the approval from their buy status option

Use case#32	Approve Transaction Rent	
Goal in Context	Admin can confirm any sale by approving renter transaction	
Preconditions	Server must be up and running and admin must be logged in	
Success End Condition	Transaction successfully approved and renters can see the approval	
Failed End Condition	Transactions were not approved	
Primary Actors	Admin, database	
Secondary Actors	Renter	
Trigger	Admin clicks the approve button	
Description	Step	Action
	1	Admin logs into their account
	2	Admin navigates to dashboard
	3	Admin navigates to view renters option
	4	Admin clicks approve option from the action column if they want to approve transaction of renters
	5	Database is updated with the action
Extension	Step	Branching Action
	1	Admin can delete transactions
	2	Renters can see the approval from their rent status option

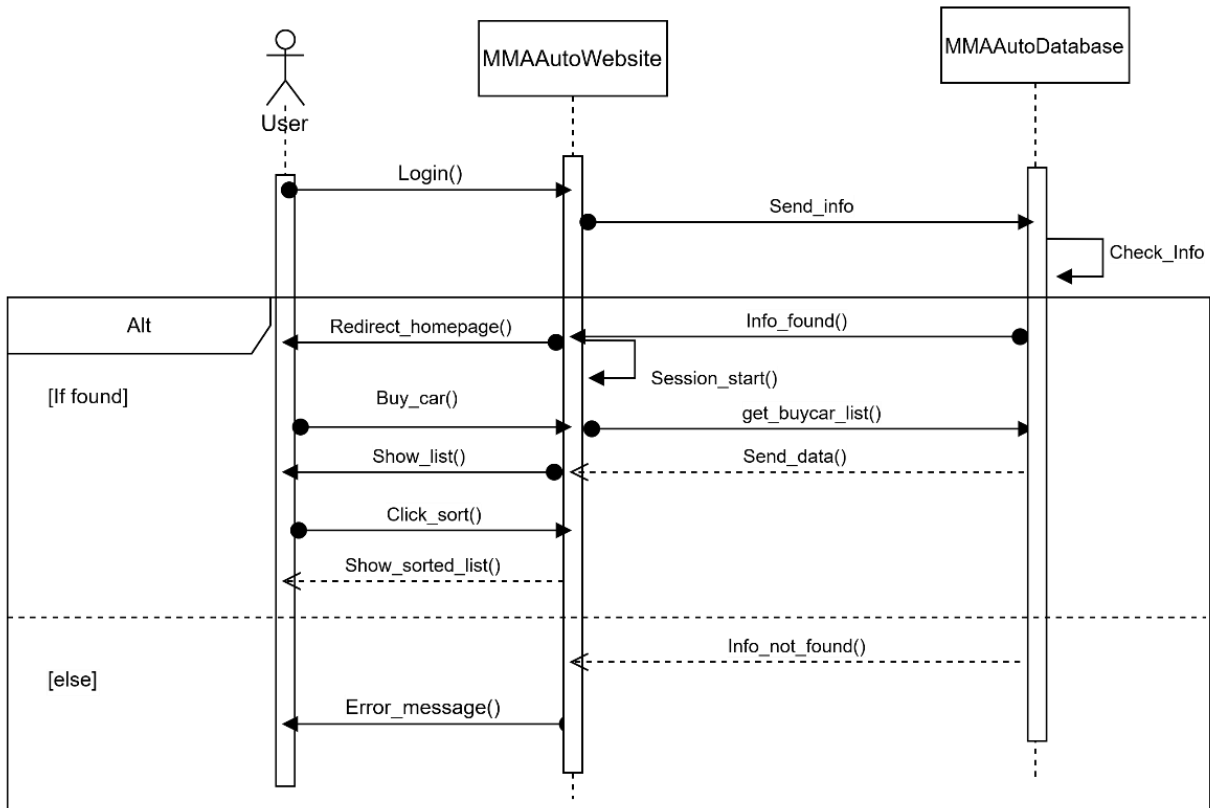
Sequence Diagram:



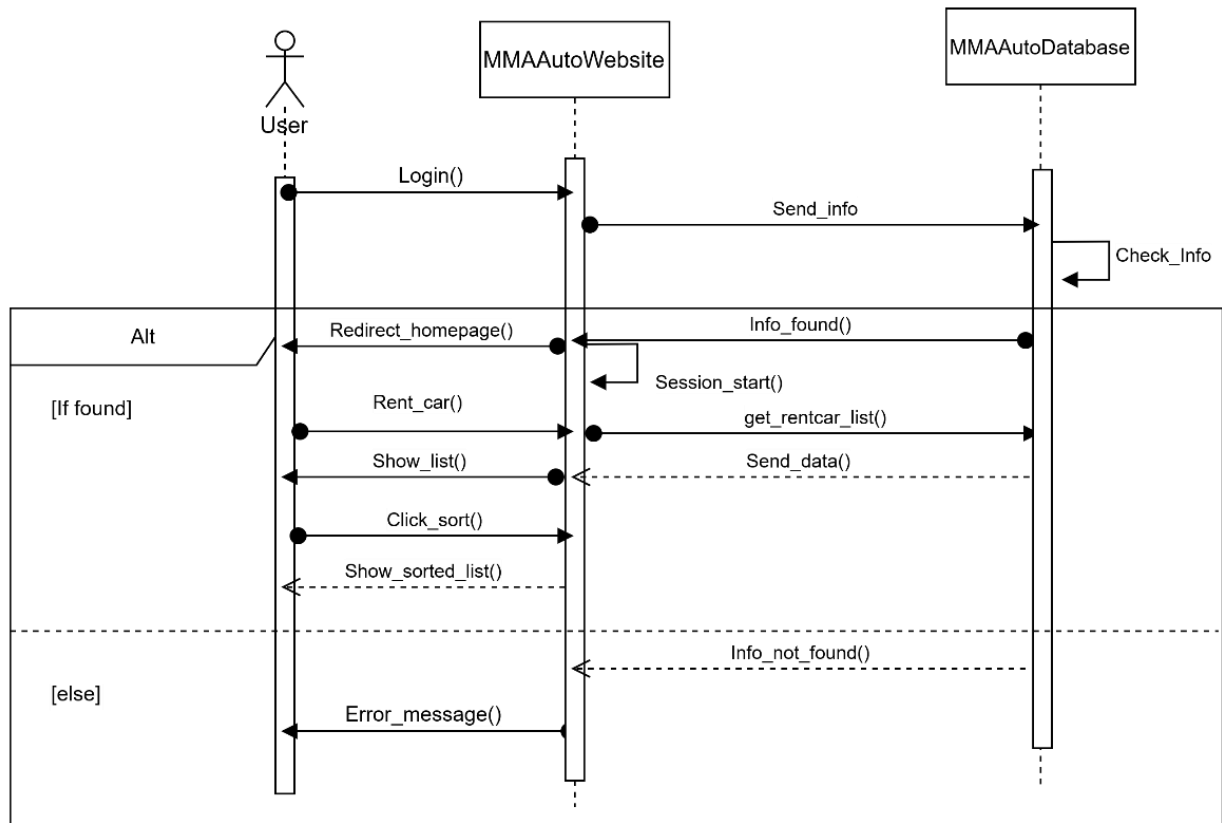




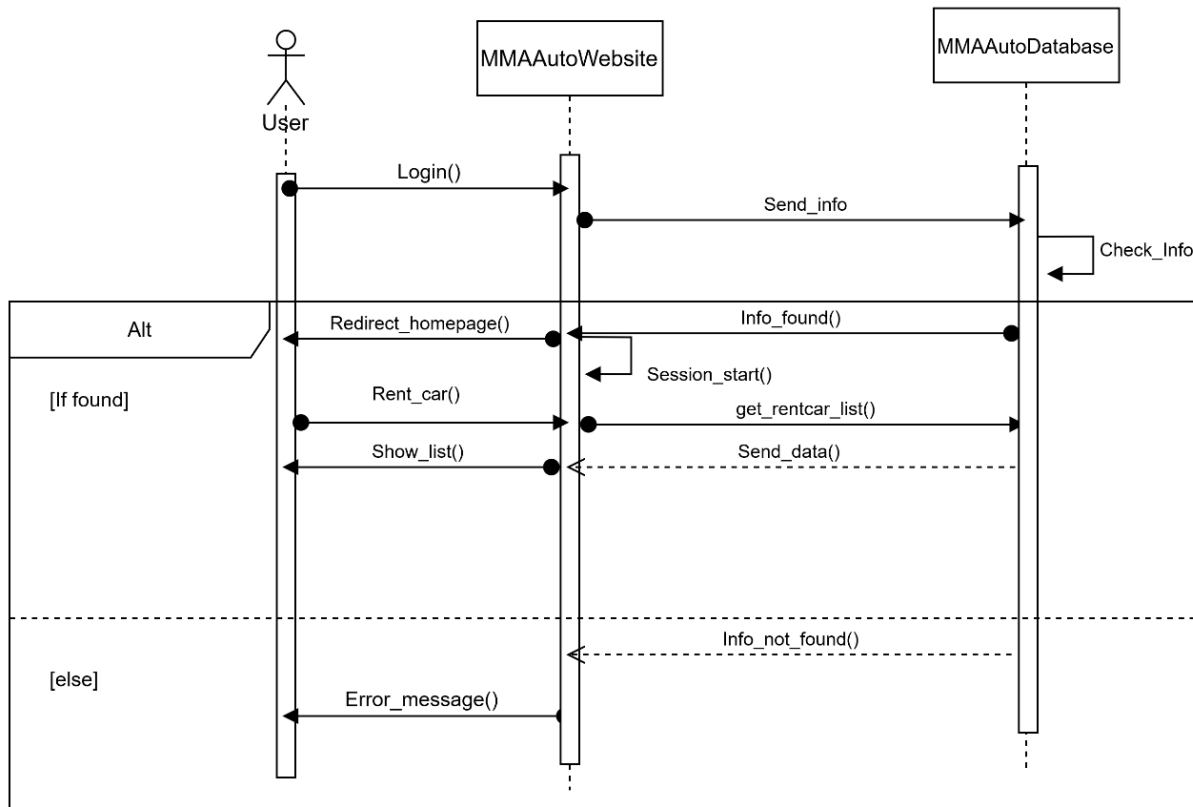
Sort Buy Car



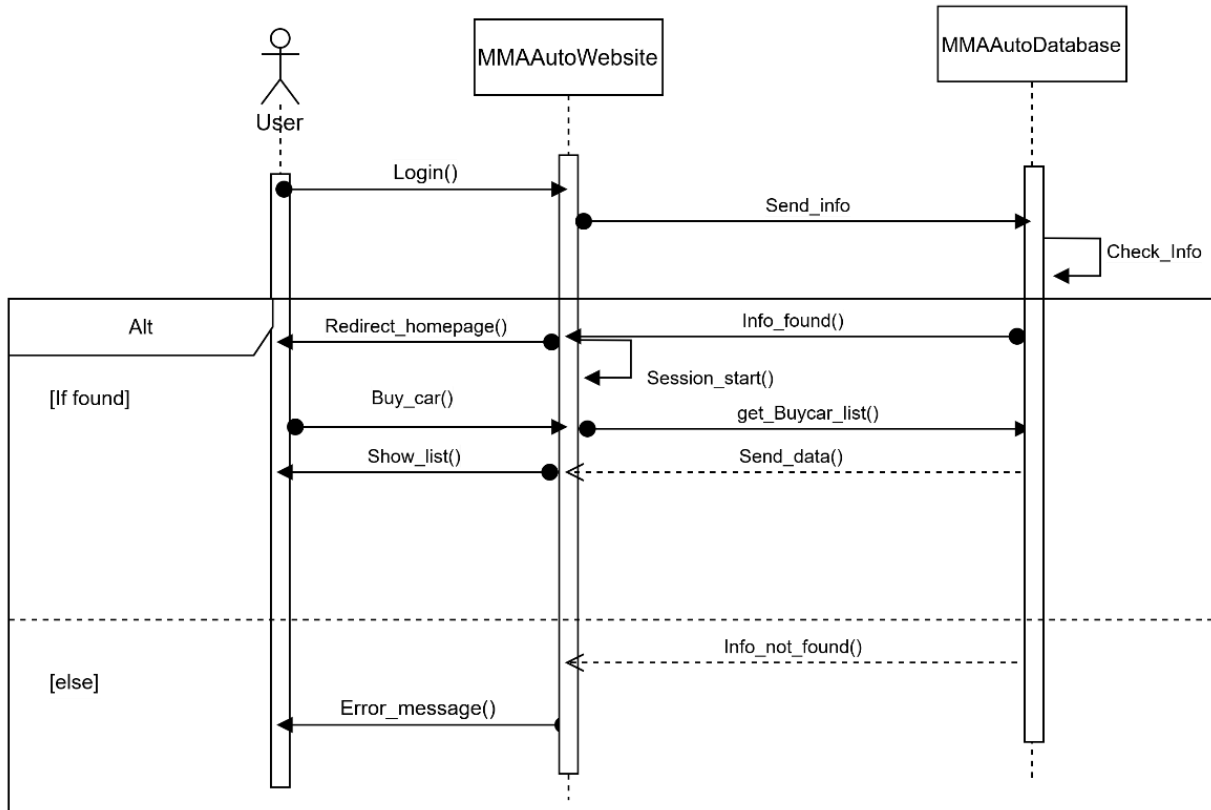
Sort Rent Car

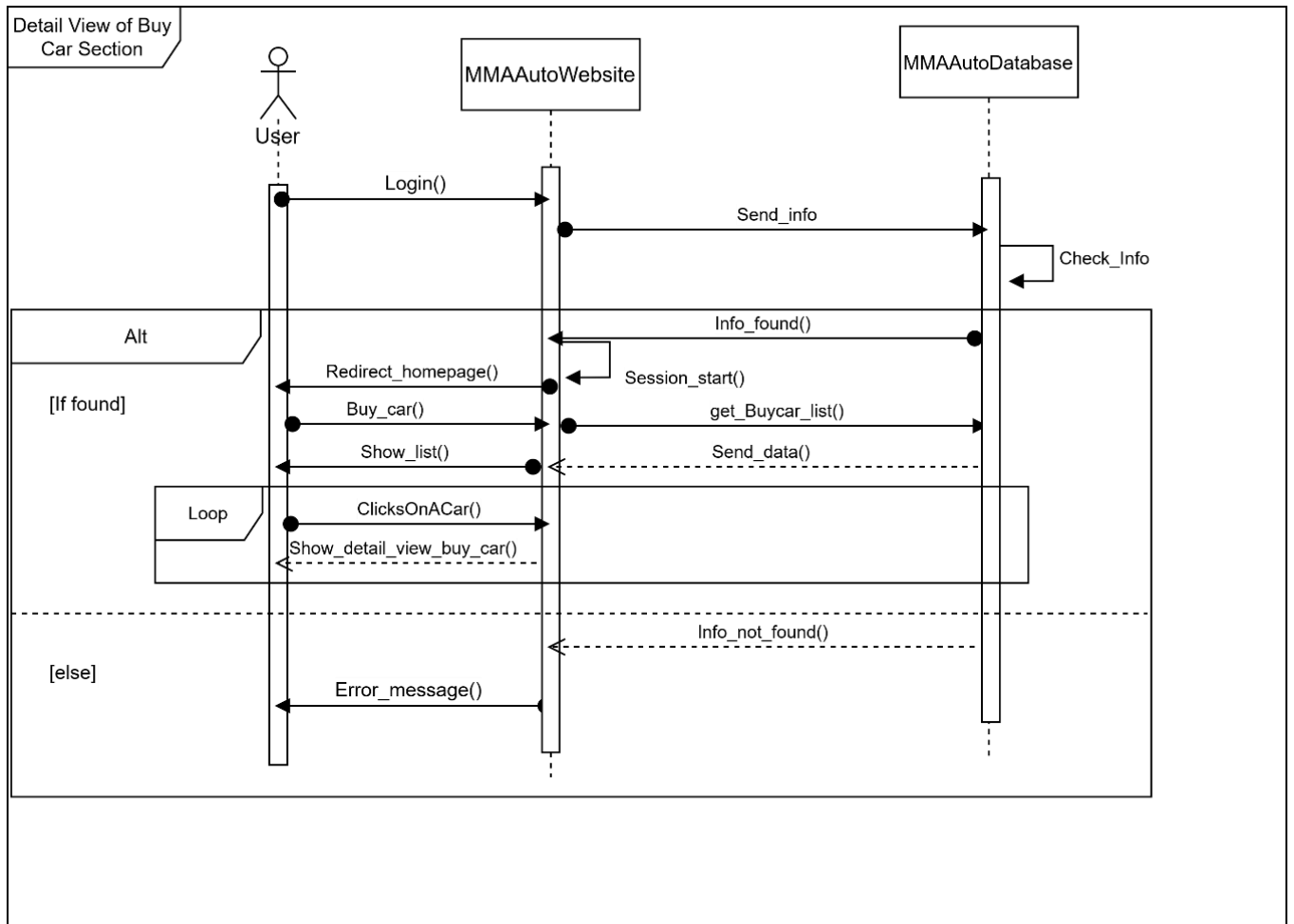


Brief View Of Rent
Car Section

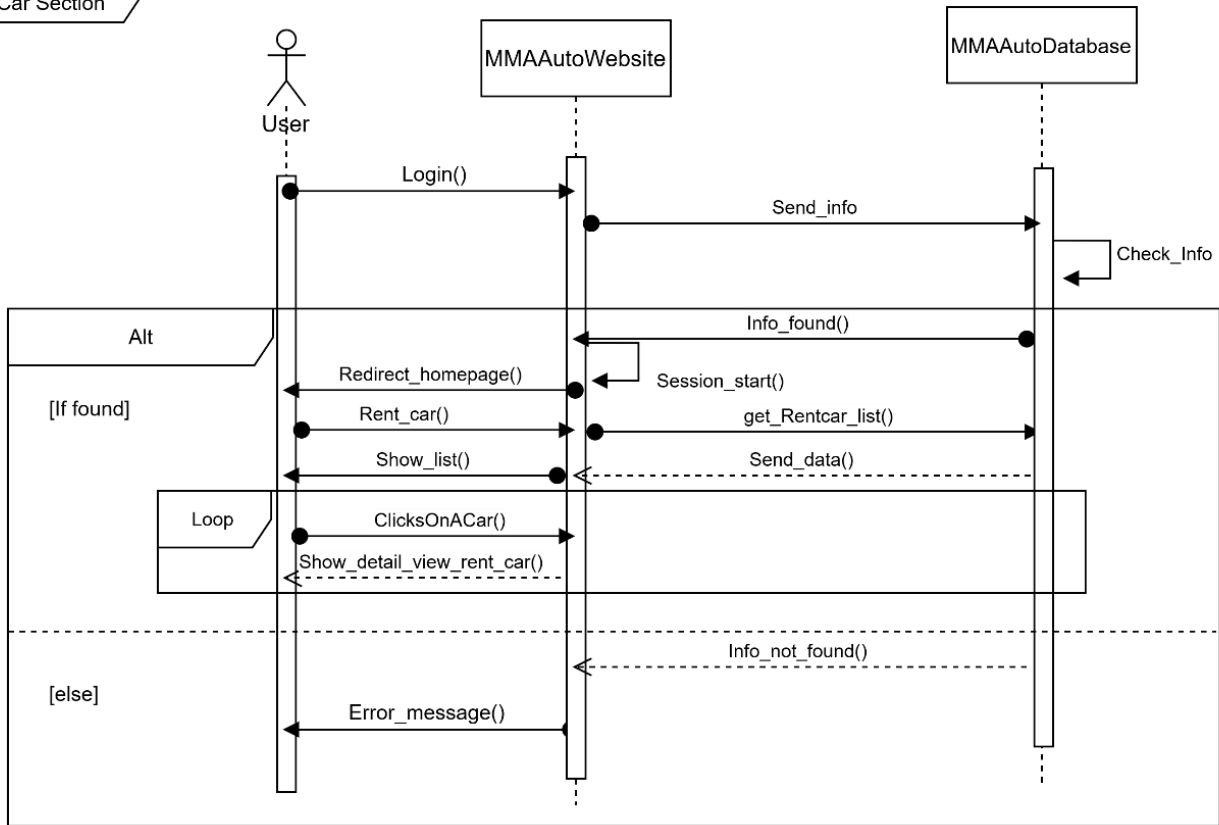


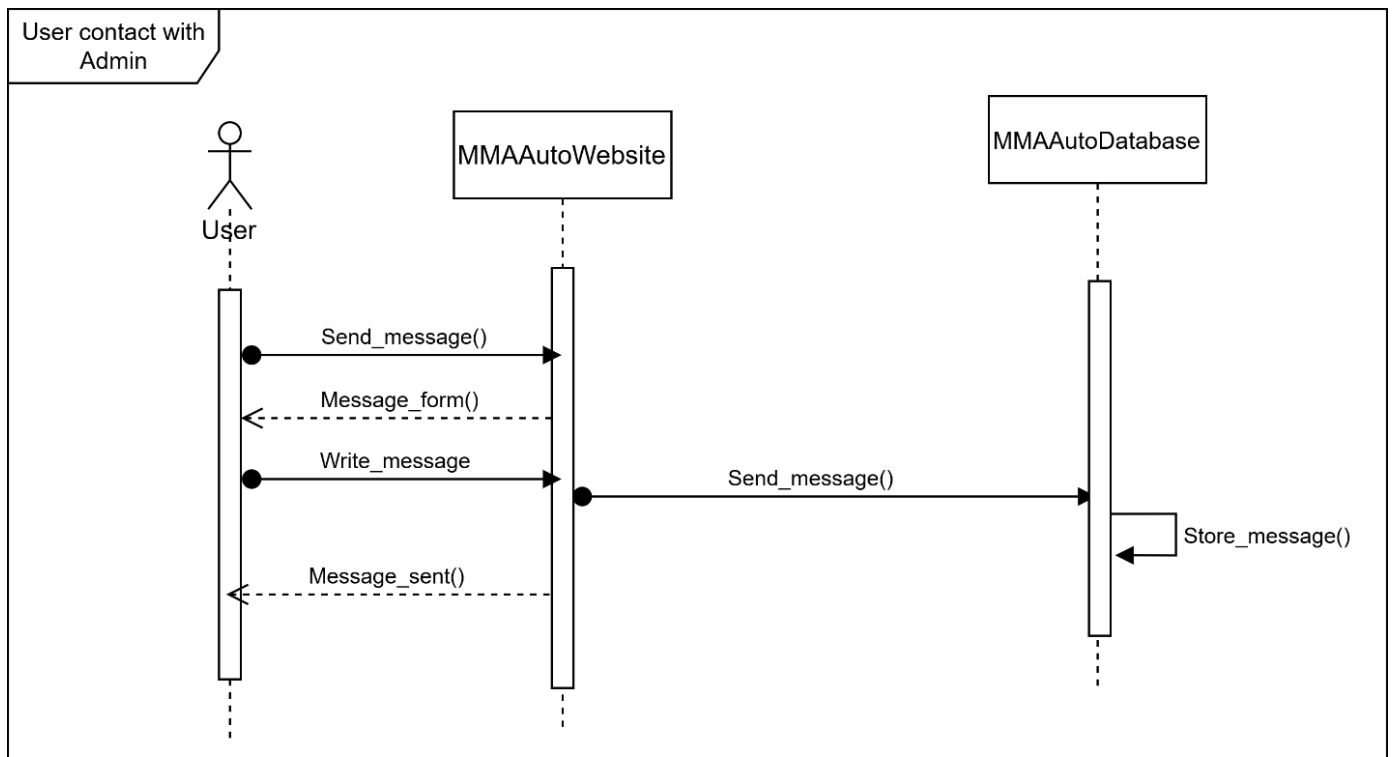
Brief View of Buy
Car Section

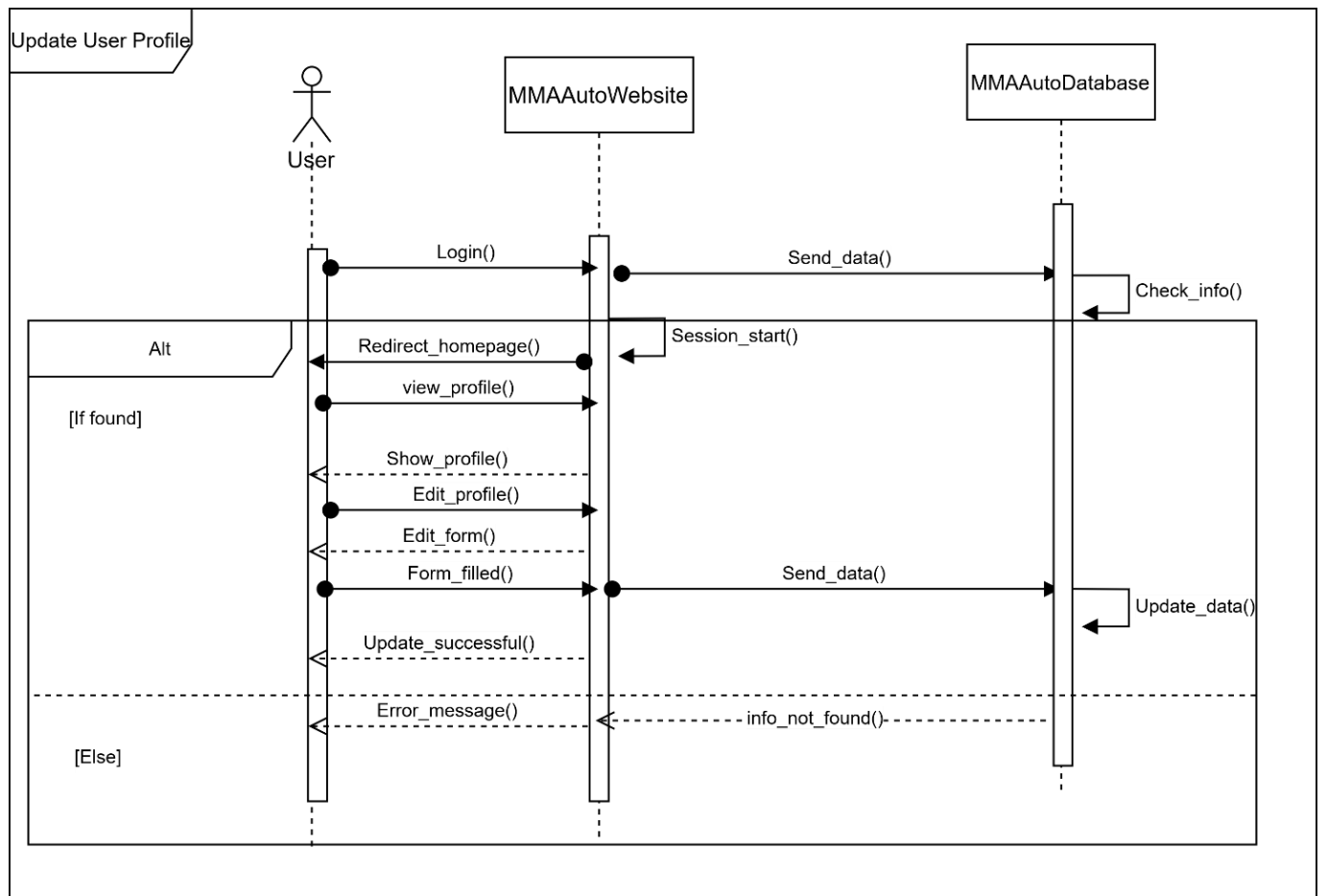


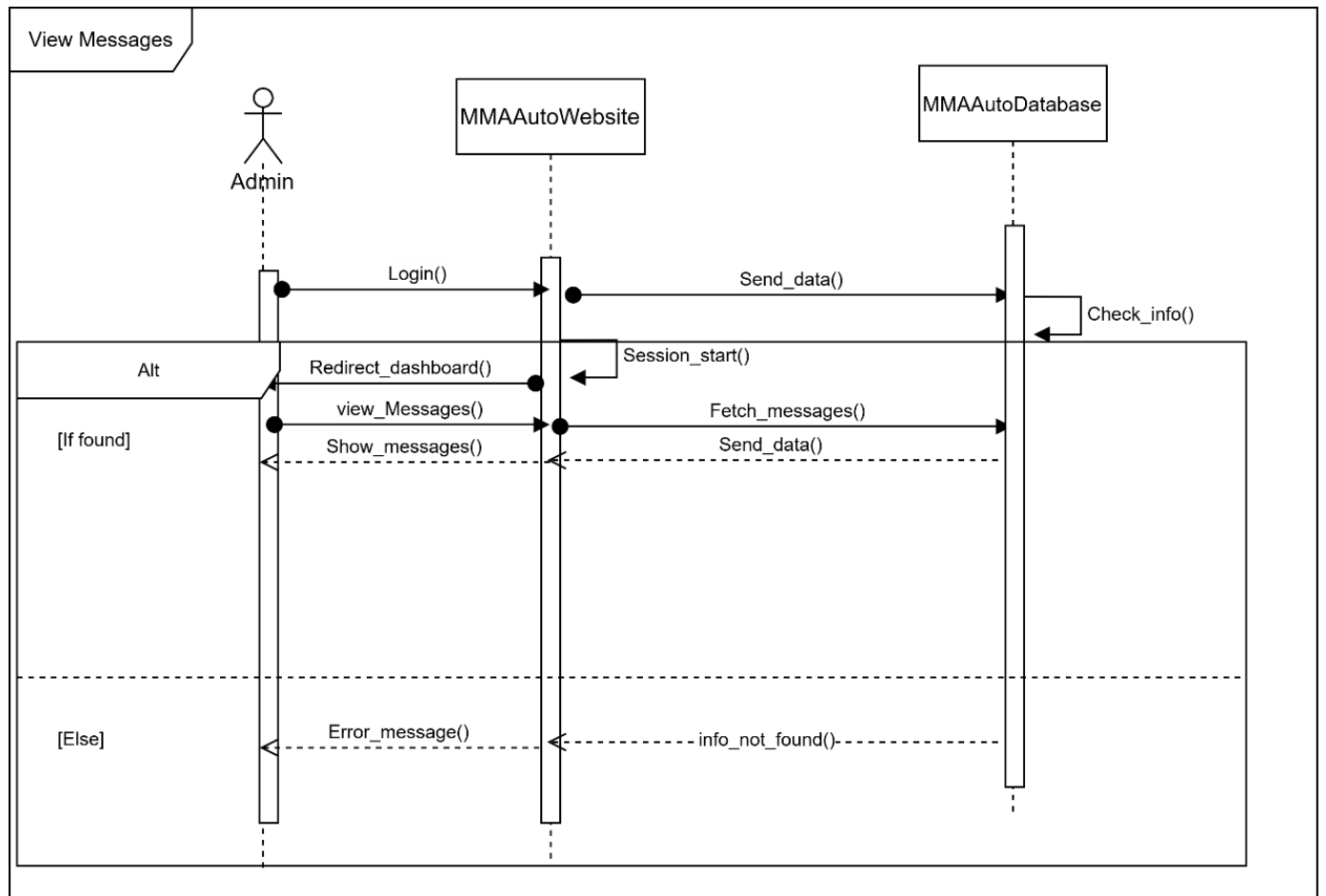


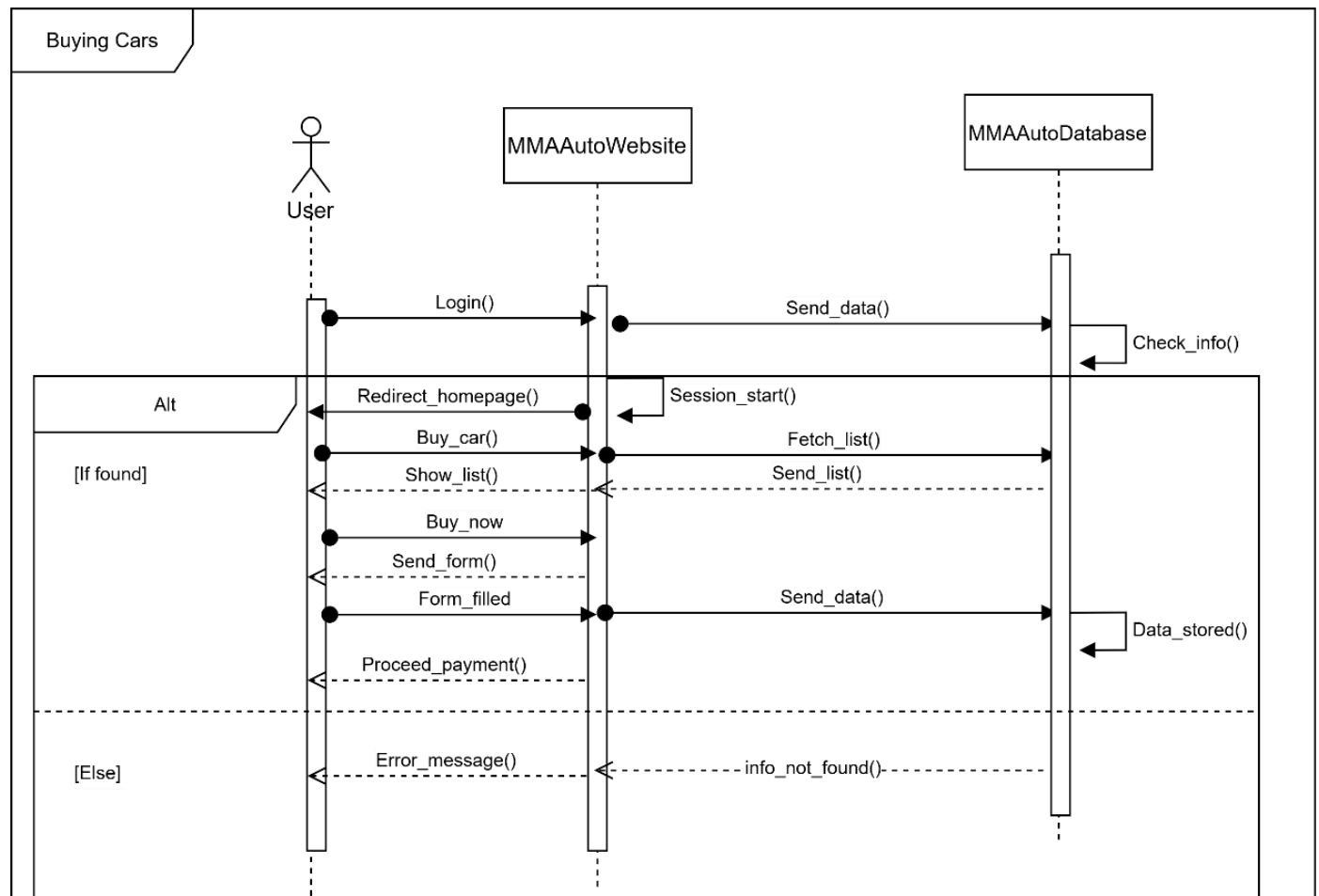
Detail View of Rent
Car Section

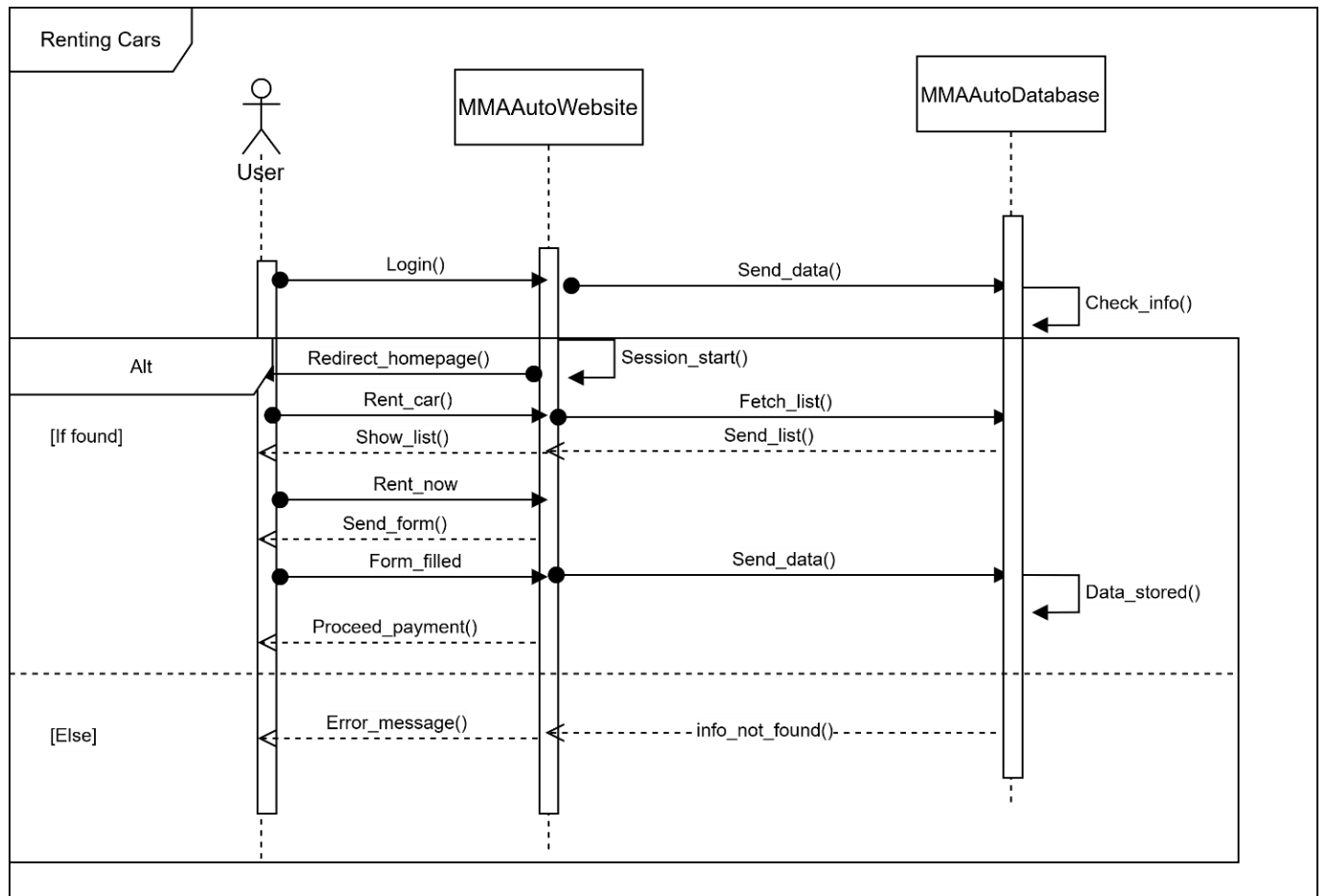












Payment for buying
Cars

Buyer

MMAAutoWebsite

MMAAutoDatabase

Proceed_payment

Get_info()

Send_form(buy_amount,nid)

Send_info(buy_amount,nid)

Form_filled(amount_input)

send_info(amount_input)

Check_amount()

Alt

Success()

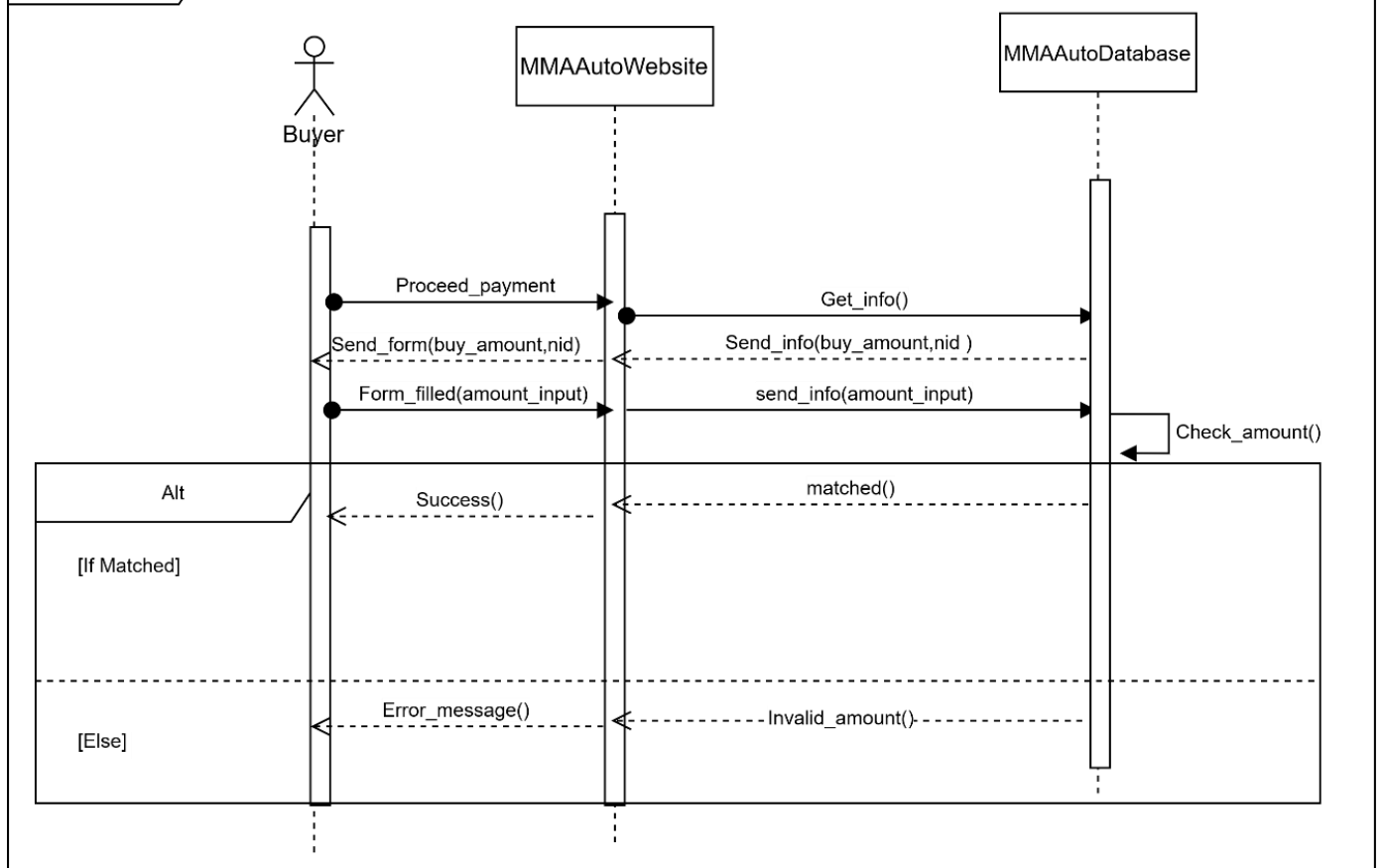
matched()

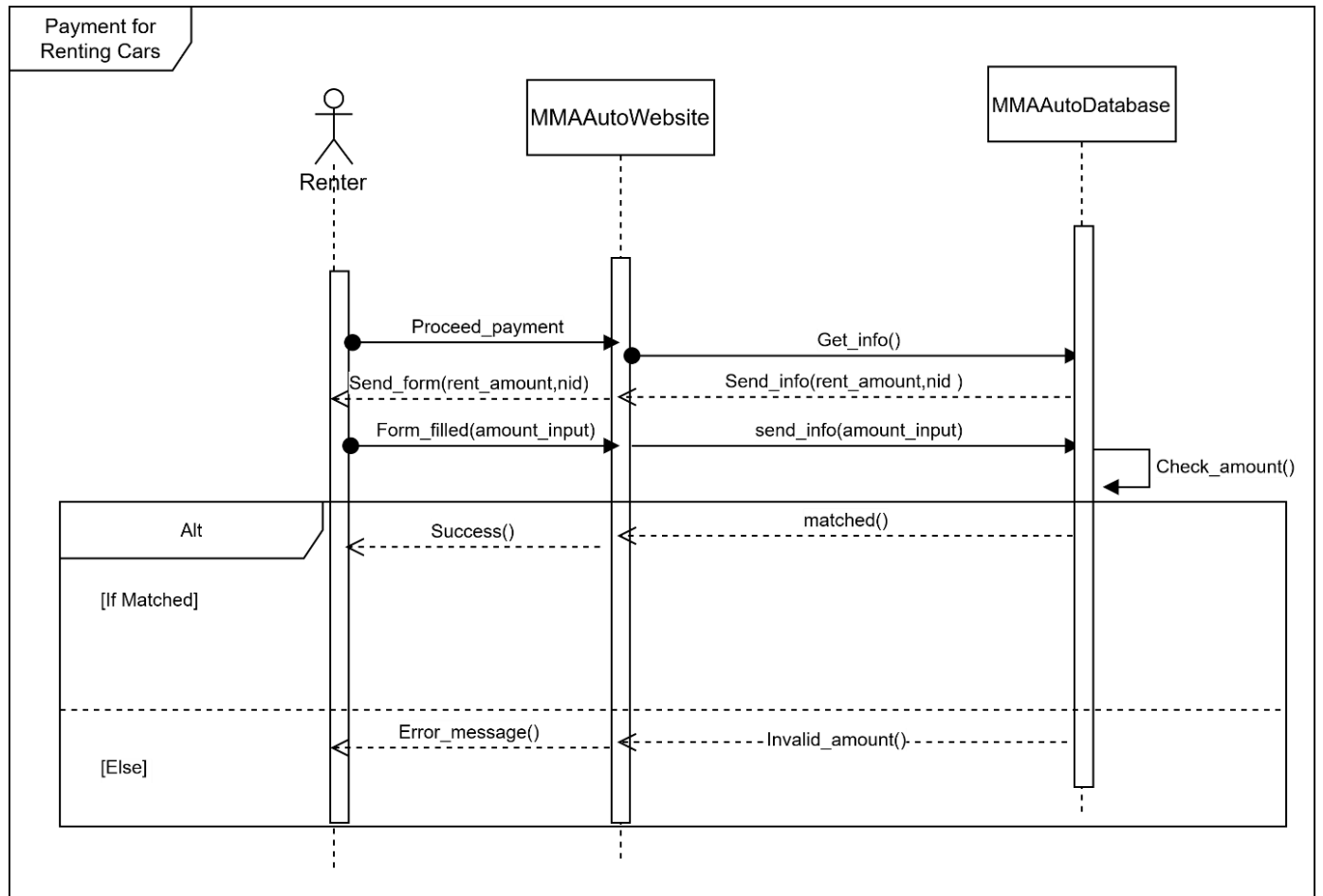
[If Matched]

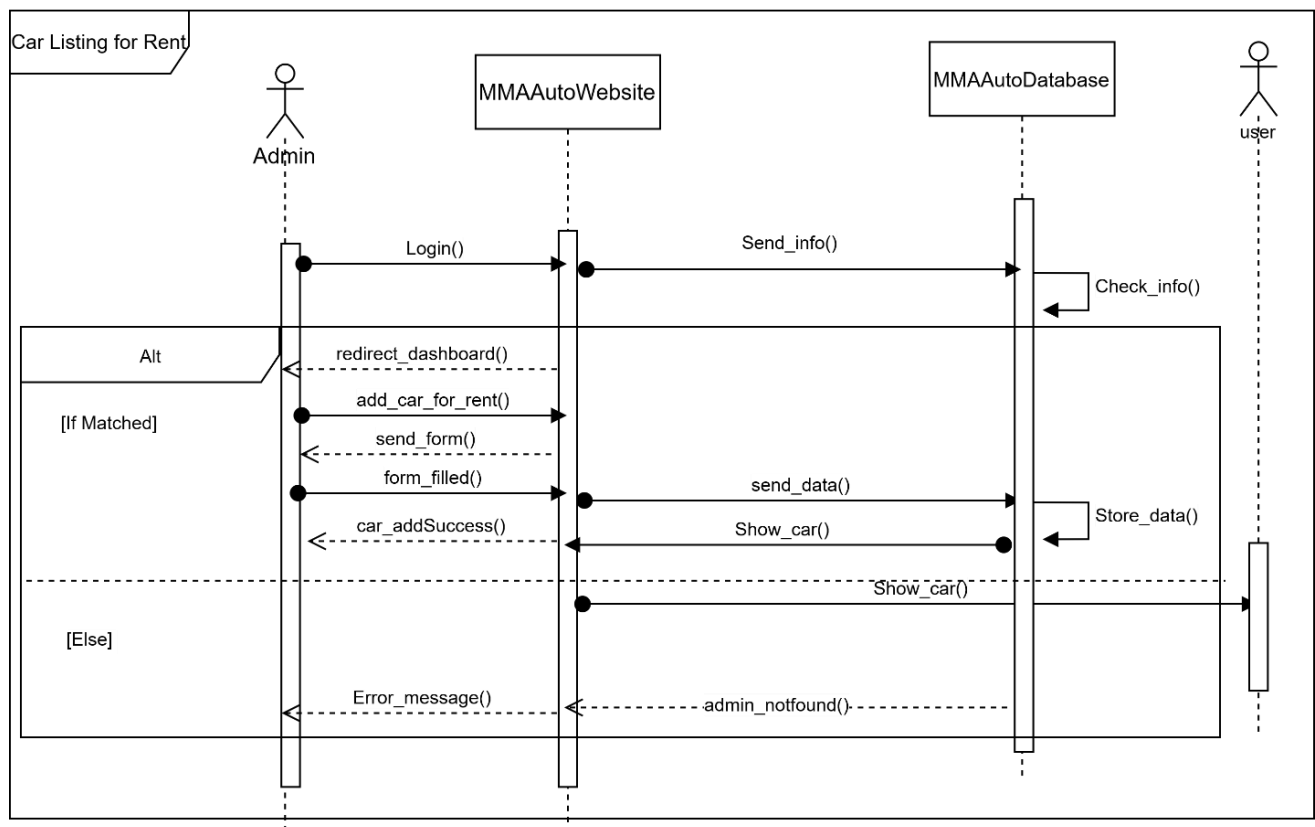
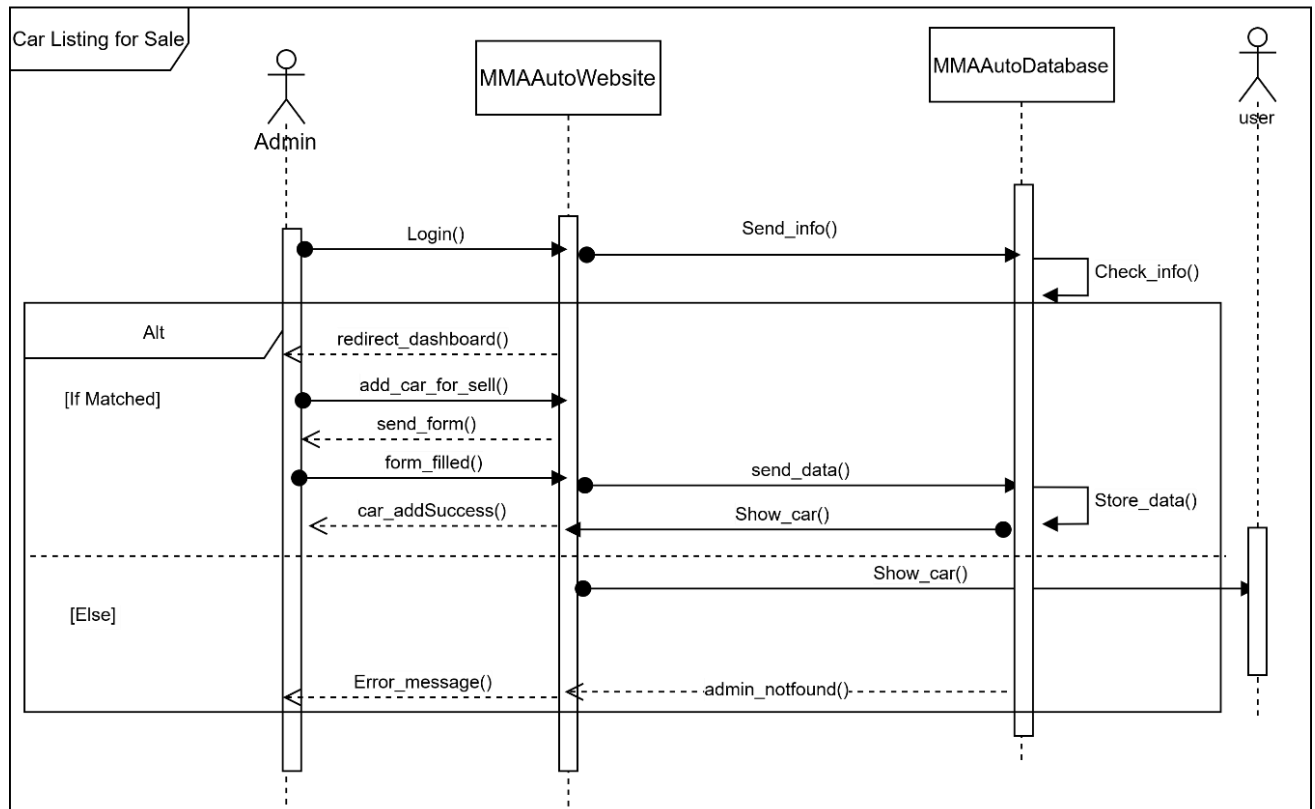
Error_message()

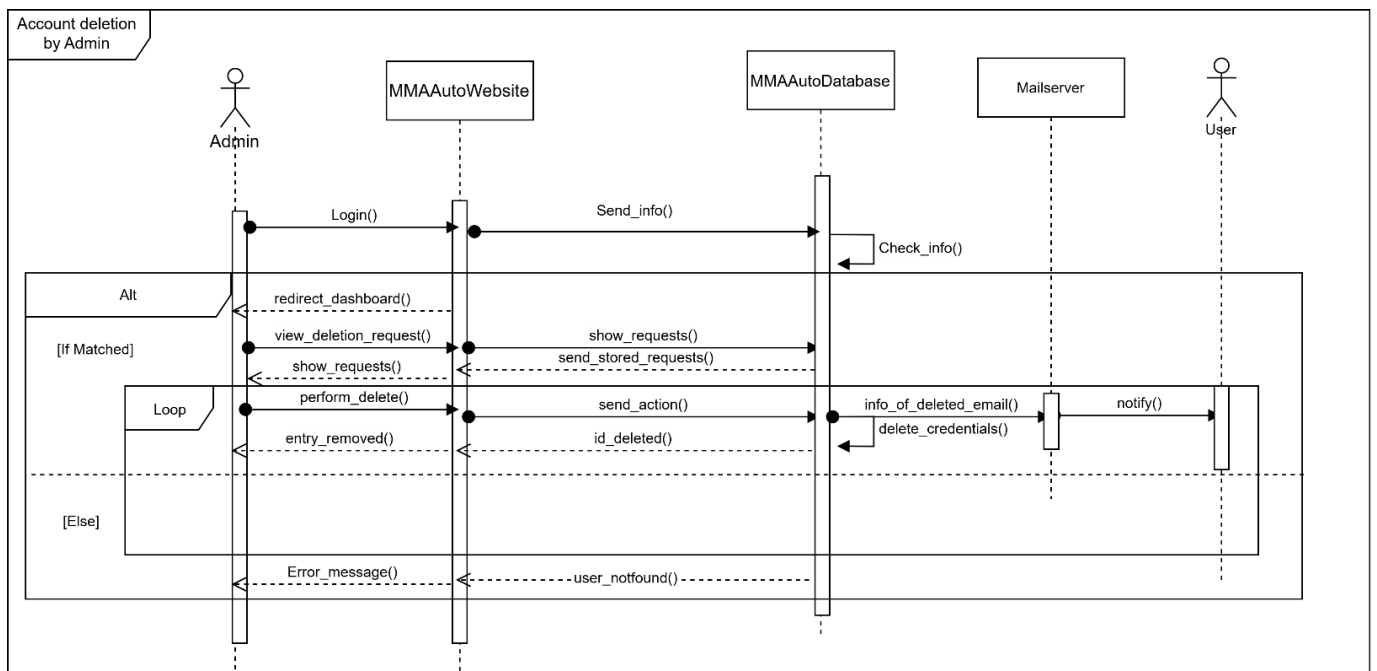
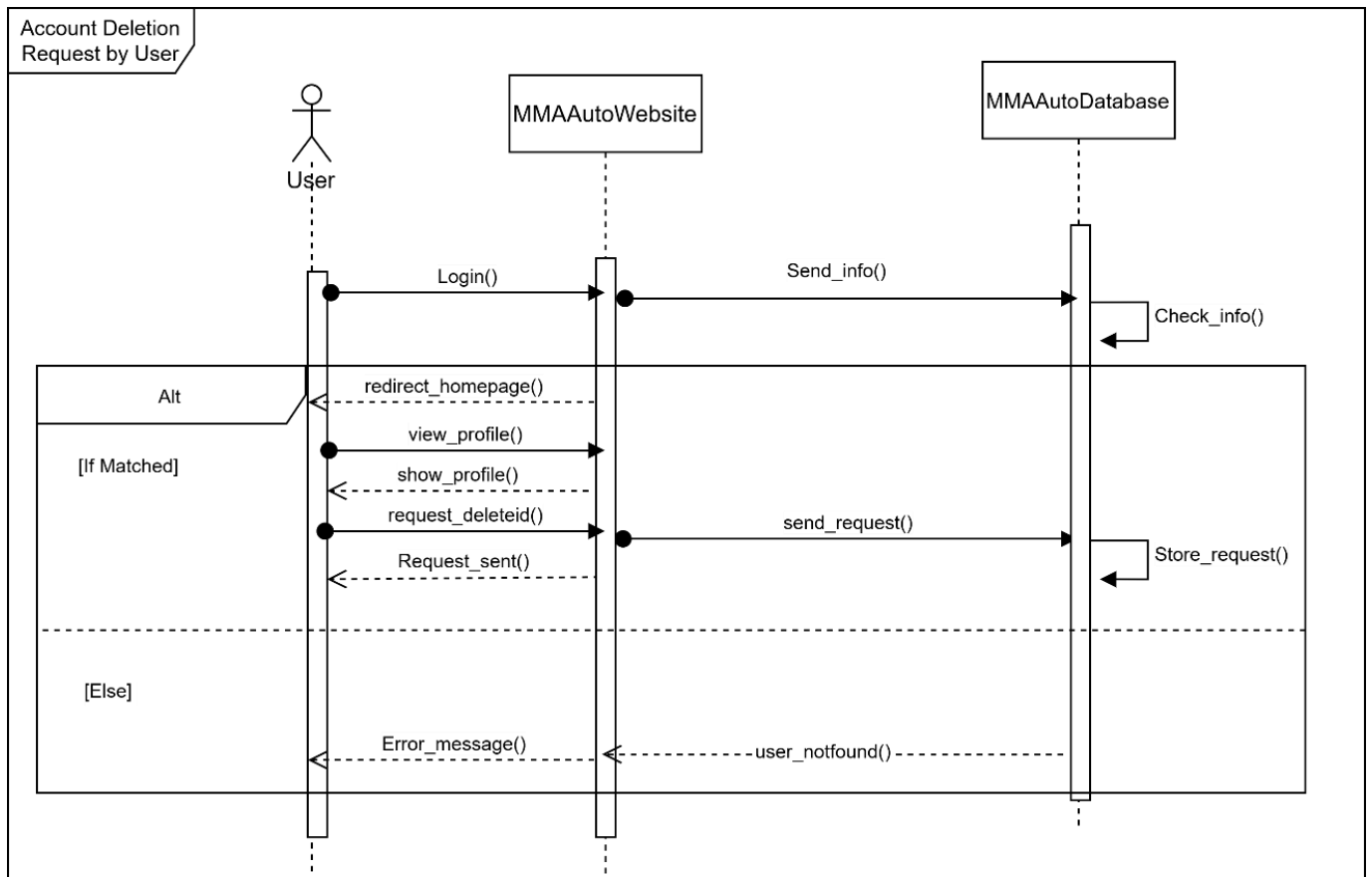
Invalid_amount()

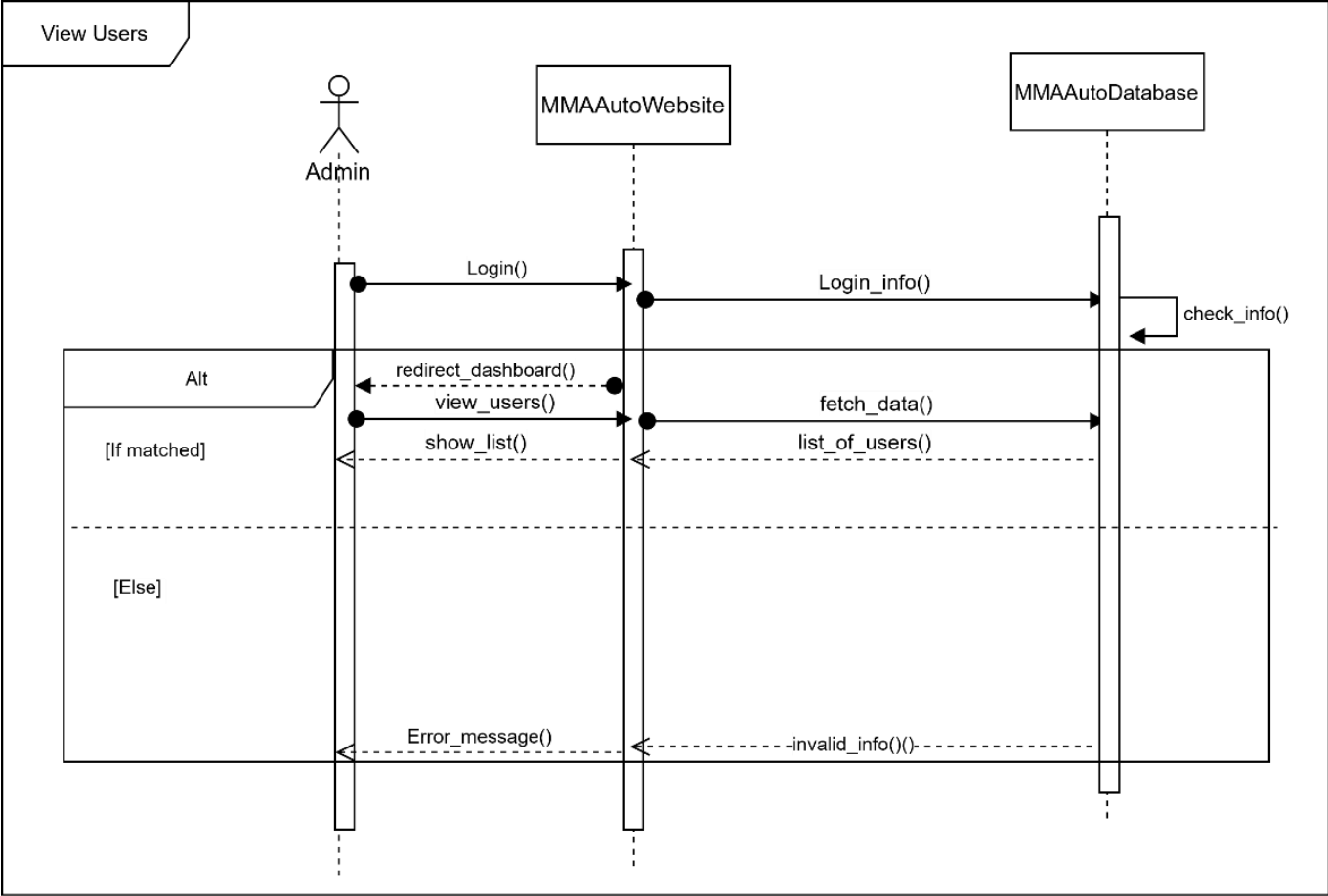
[Else]

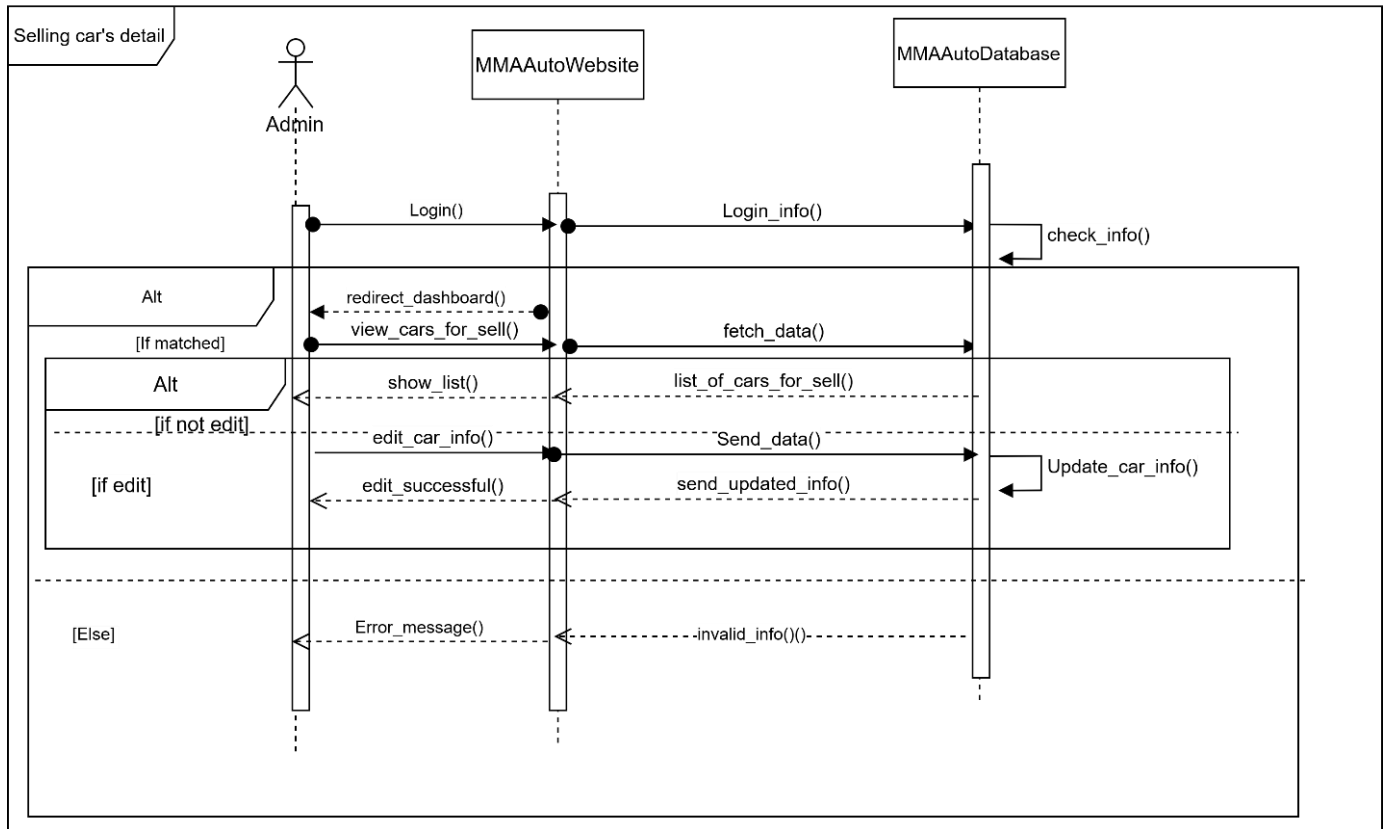


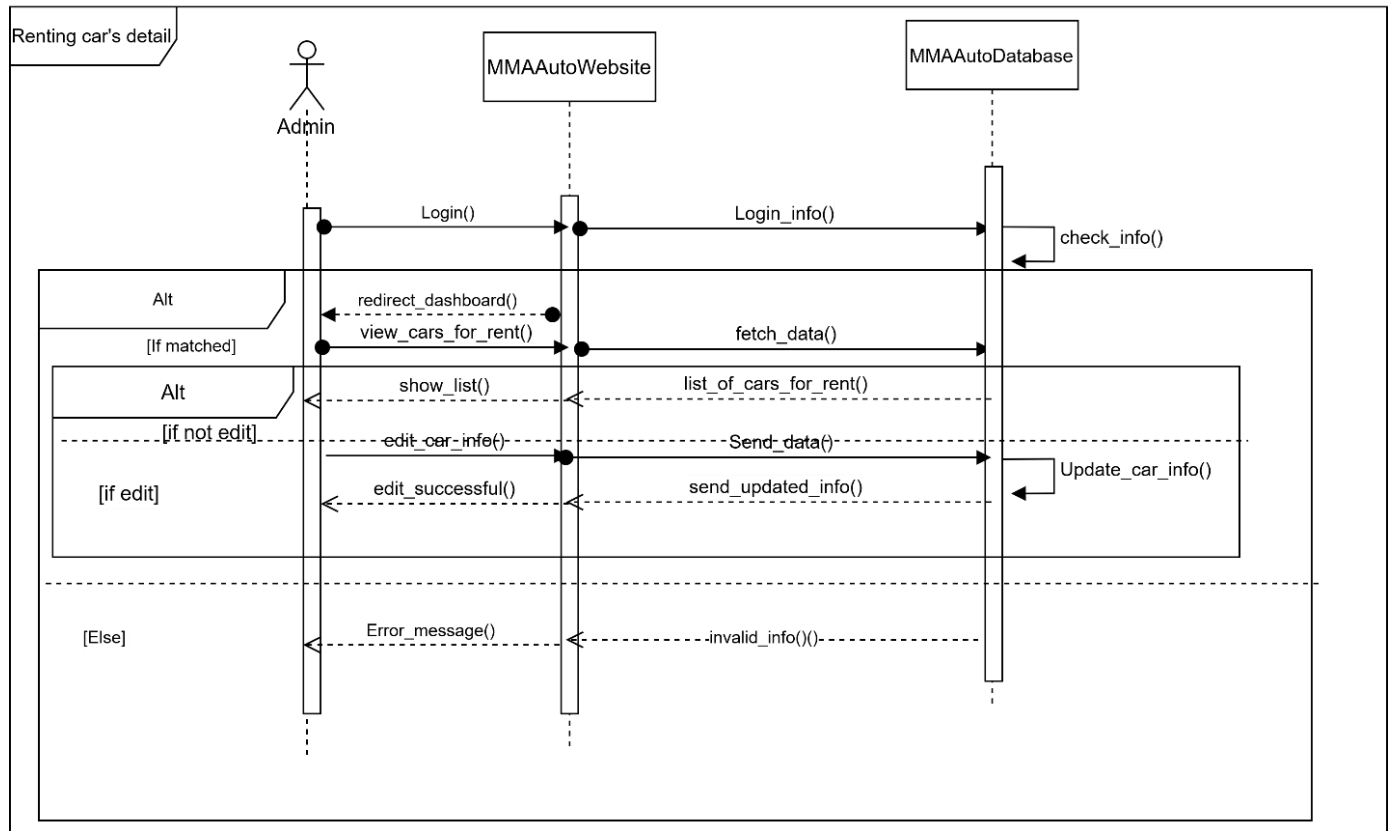


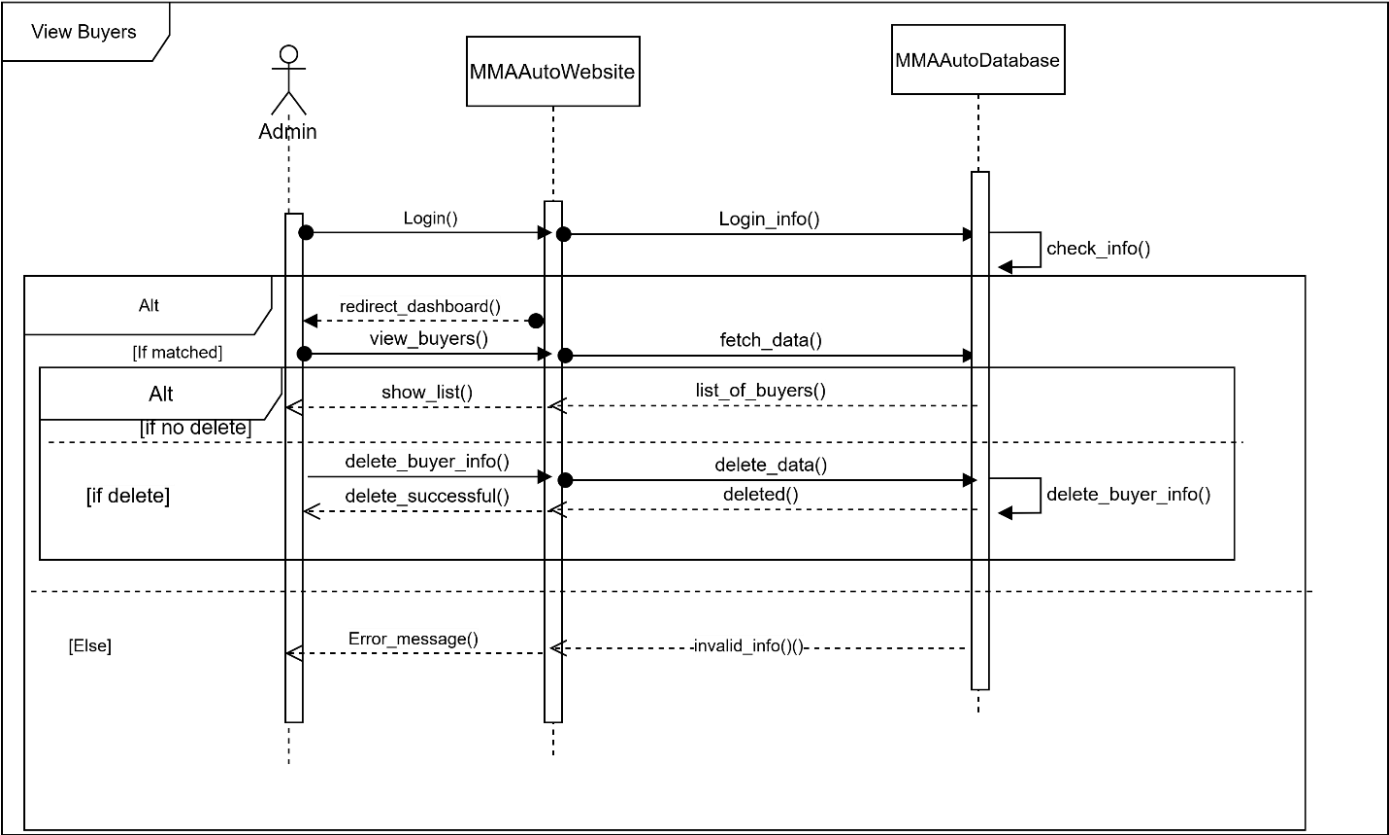


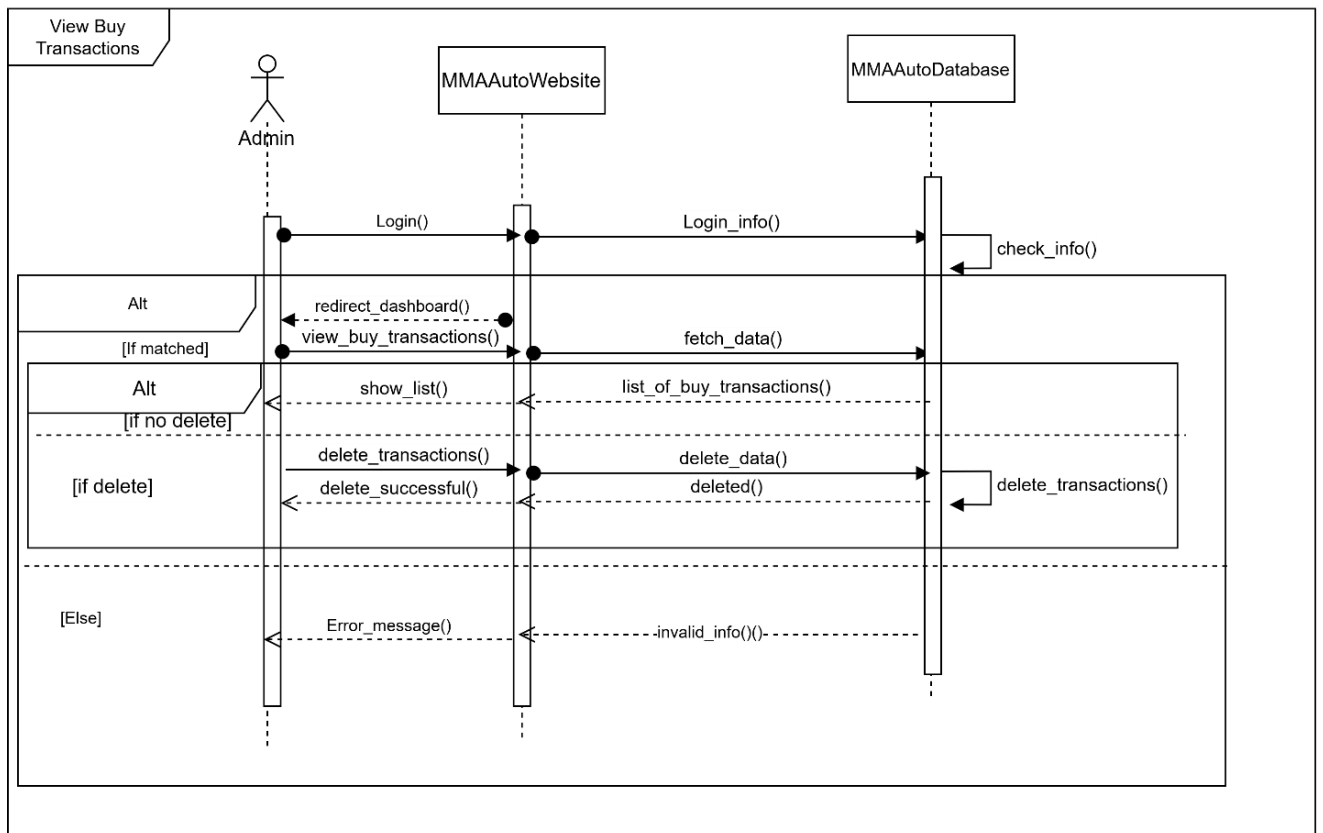
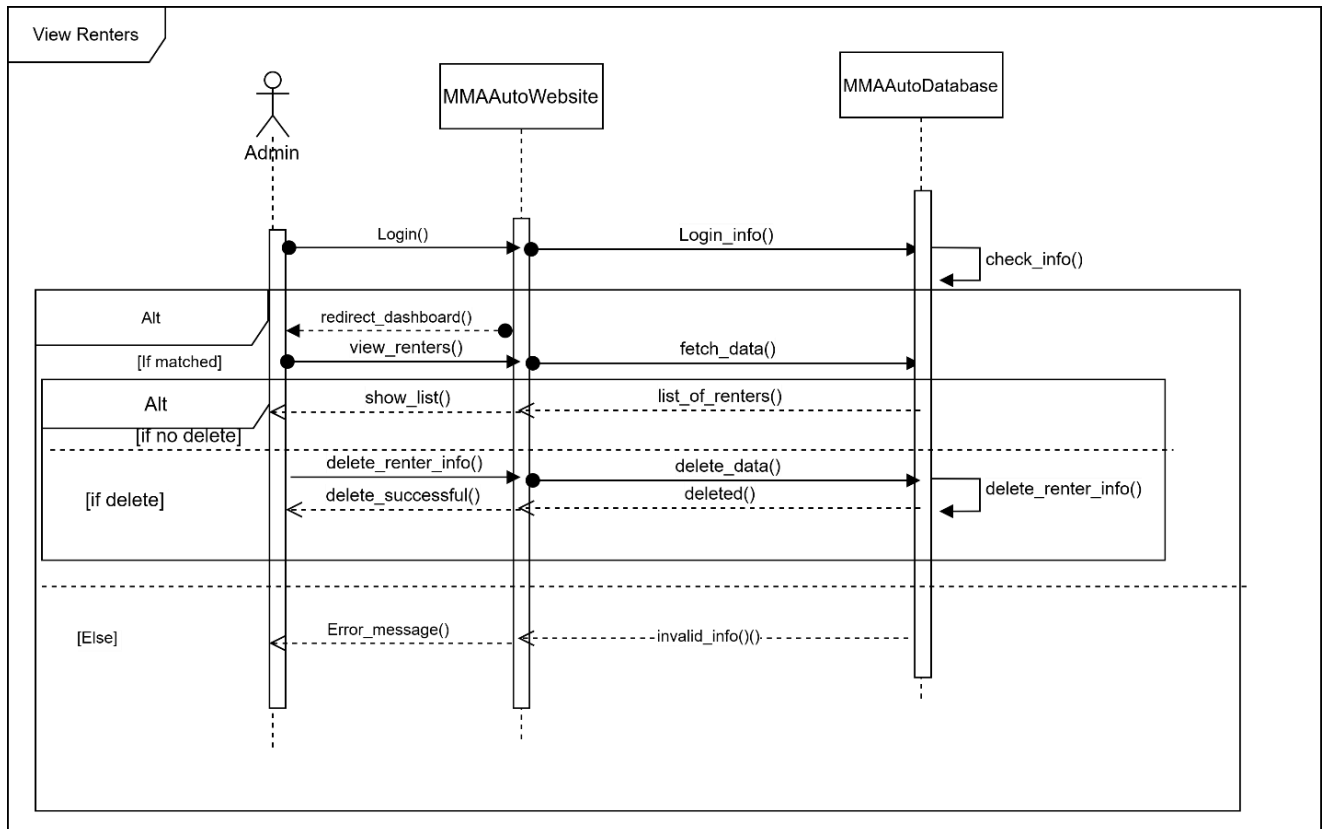


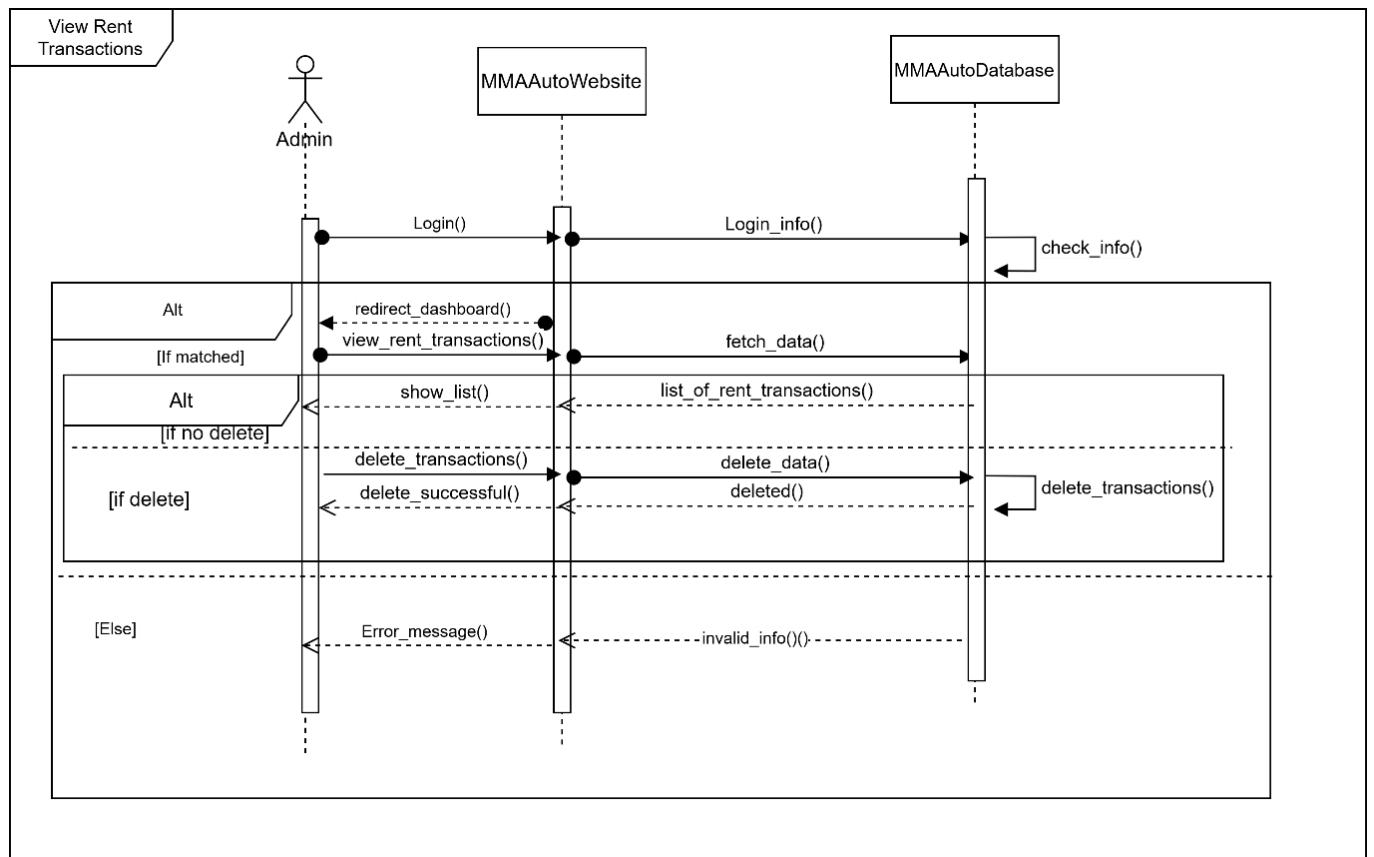




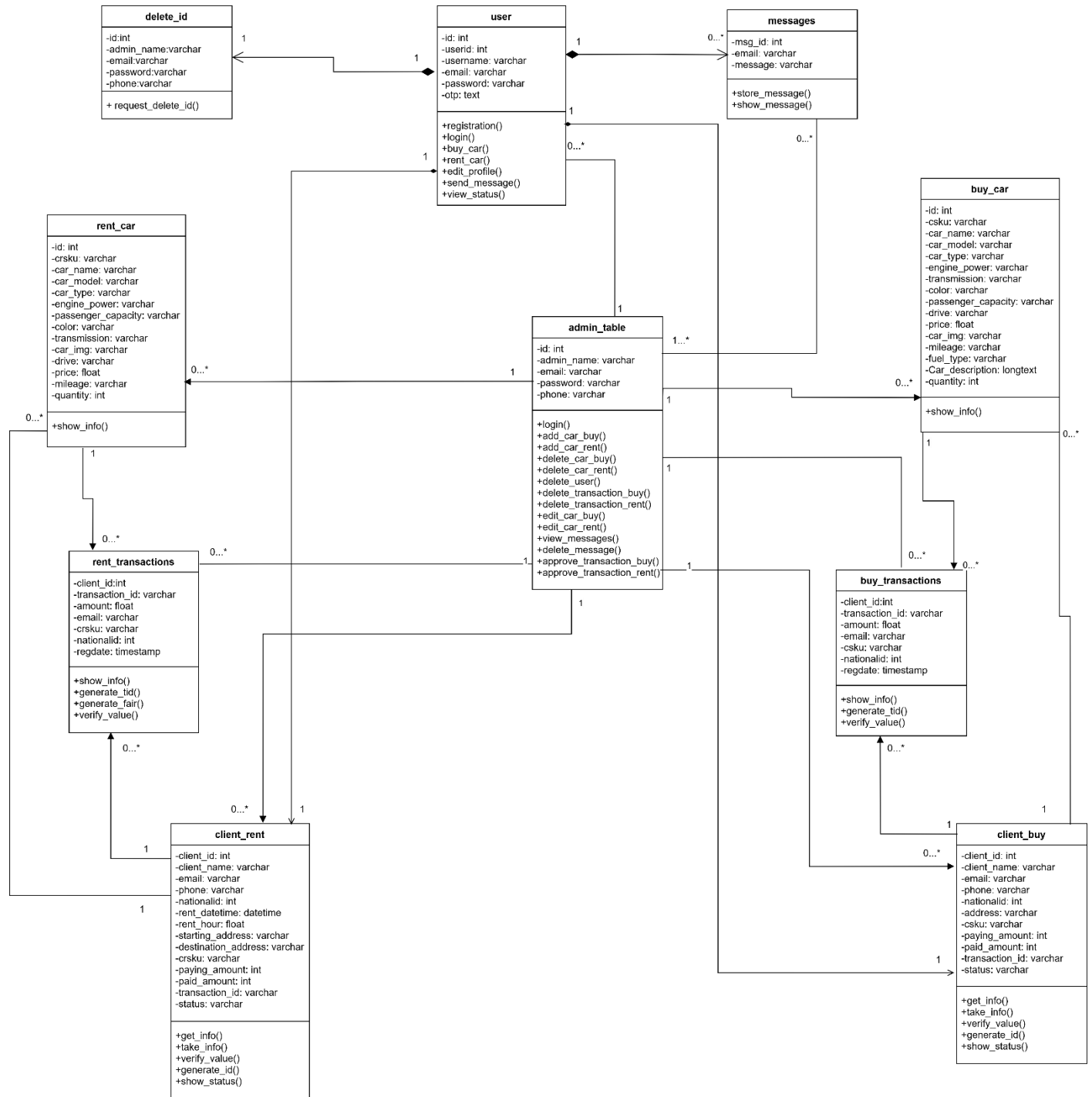




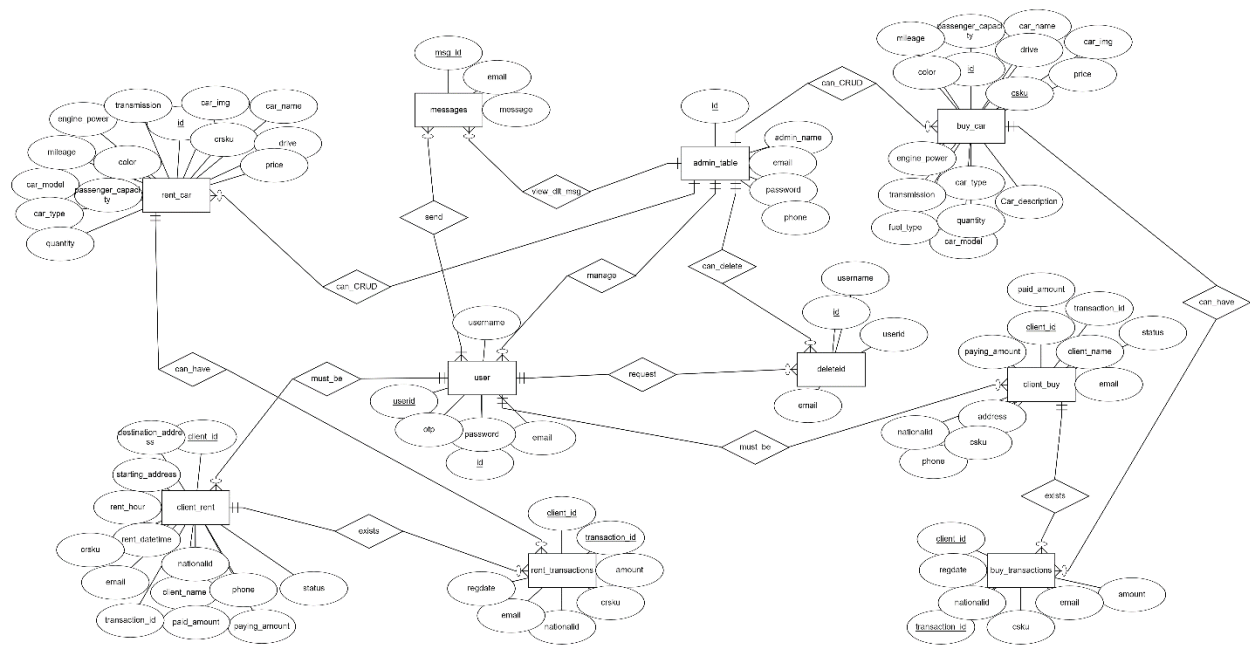




UML Class Diagram:



ER-Diagram:

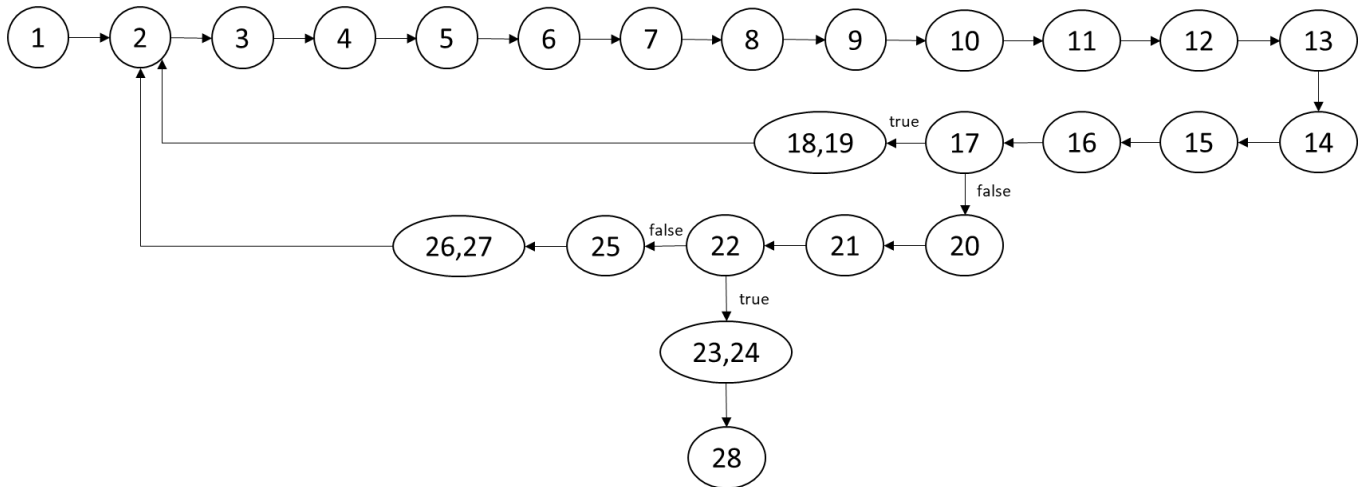


White Box Testing 1: (Add Car)

1. begin
2. img = input()
3. csku =input()
4. carname= input()
5. carmodel = input()
6. cartype = input()
7. engine = input()
8. transmission= input()
9. color= input()
10. passcap= input()
11. drivewheels= input()
12. price= input()
13. mileage= input()
14. fuel= input()
15. description= input()
16. quantity= input()
17. if(mysql_num_rows(query(select * from buy_car where csku=csku or car_name=carname and car_model=carmodel)>0)
18. print: car already exists
19. goto line: 2
20. else
21. sql=query(insert into buy_car(csku,car_name,car_model,car_type,engine_power, transmission, color, passenger_capacity,drive, price,car_img,mileage, fuel_type,Car-description,quantity) values (csku,carname.carmodel,cartype,engine,transmission, color,passcap,drivewheels, price,img,mileage,fuel,description,quantity))
22. if(sql)
23. print: car added
24. goto line: 28
25. else

26. print: error message
27. goto line: 2
28. End

a. Control Flow Graph:

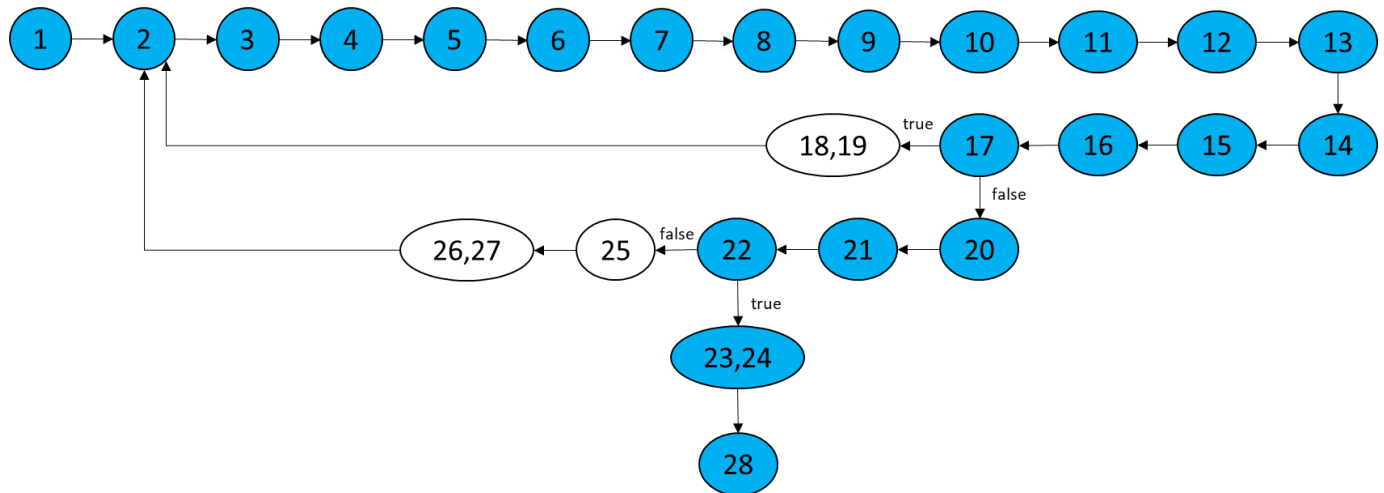


b. Test Case

img = 'audi.jpg'
csku = 'audi.1'
carname = 'audi a8l'
carmodel = '2021'
cartype = 'sedan'
engine = 'Inline 4, 1496 cm3 or 91.3 cu-intransmission'
color = 'black'
passcap = '4'
drivewheels = 'AWD'
price = '36400000'
mileage = '14.88 km/l'
fuel = petrol
description = 'Inline 4, 1496 cm3 or 91.3 cu-in...'
quantity = '4'

select query unsuccessful at line 17

insert query successful at line 21



White Box Testing 2: (Make Payment)

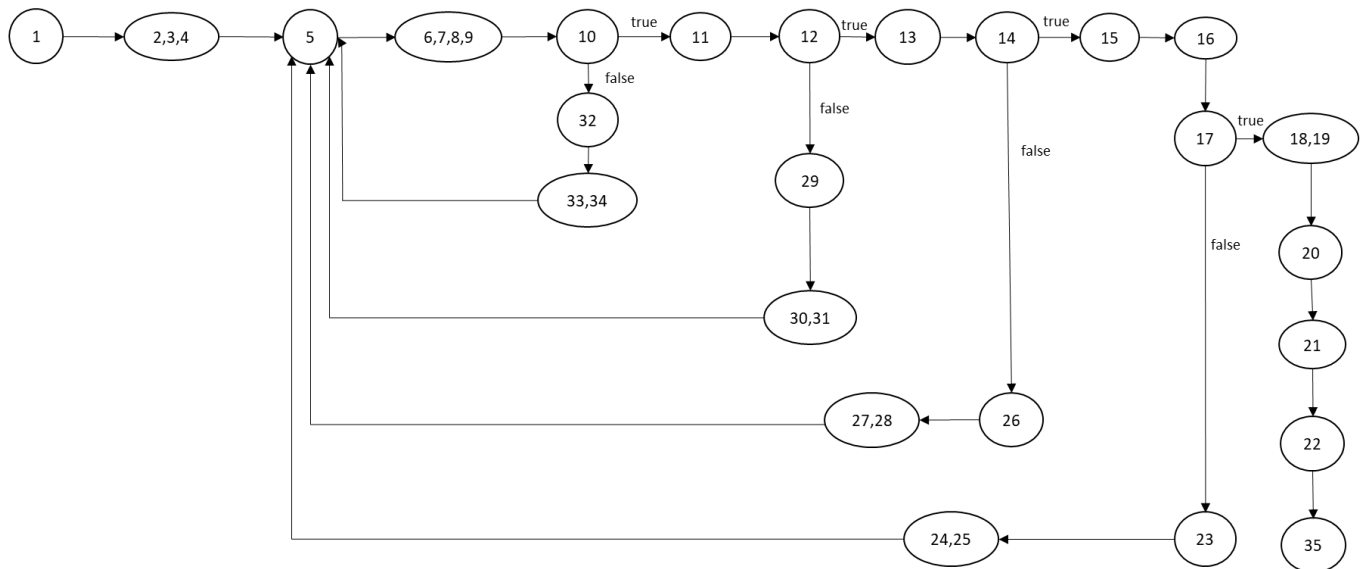
1. Begin
2. keylength=8
3. str= string
4. randstr = random_string
5. amount = input()
6. nationalid = get(from database)
7. csku = get(from database)
8. paying_amount = get(from database)
9. client_id = get(from previous page)
10. if(amount === paying_amount)
11. result=query(update client_buy set transaction_id = randstr, paid_amount=amount, status='approved' where csku = csku and client_id = client_id)
12. if(result)
13. newquery= query(insert into buy_transactions (client_id,transaction_id, amount,email, csku,nationalid) values (client_id, randstr, amount, session_email, csku,nationalid)
14. if(newquery)
15. print: Infomation Added Succesfully
16. sql= query(select * from buy_car where csku=csku)
17. if(mysql_fetch_assoc(sql))
18. csku=get(from database)
19. quantity= get(from database)
20. quantity1 = quantity - 1
21. sql2=query(update buy_car set quantity = quantity1 where csku=csku)
22. goto line 35

```

23.             else
24.             print: error message
25.             goto line 5
26.         else
27.         print: error message
28.         goto line 5
29.     else
30.     print: error message
31.     goto line 5
32. else
33. print: enter correct amount
34. goto line 5
35. end

```

a. Control Flow Graph:



b. Test Case

```

Keylength=8
str= '1234567890abcdefghijklmnopqrstuvwxyz'
randstr = nkvt rp6d
amount = 125000
nationalid = 1325434
csku = audi.1
paying_amount = 125000
client_id = 12

```

update query successful at line 11

insert query successful at line 13

select query successful at line 16

update query successful at line 21

