

19CSE314-SOFTWARE ENGINEERING

PROJECT NAME : CAB HIRING APPLICATION



Team members:

Roll Number	Name	Role for Sprint 1
CB.EN.U4CSE19403	Aadith S	Scrum Master
CB.EN.U4CSE19407	Akshara S Nair	Developer
CB.EN.U4CSE19424	Gokuldas D	Tester
CB.EN.U4CSE19447	Pala Gnana Harshavardhan	Developer
CB.EN.U4CSE19452	Radhika Rajalal	Developer

INDEX / TABLE OF CONTENTS

<u>INDEX NO</u>	<u>TABLE OF CONTENT</u>	<u>Page no</u>
	<i>Abstract</i>	3
	<i>Introduction/need/motivation</i>	3
	<i>Product Backlog</i> ❖ Report of product backlog from JIRA ❖ Epics and stories	4-15
	<i>Use case/DFD diagram</i>	16-19
	<i>Scrum Activities</i> ❖ Sprint 1 backlog ❖ Sprint 1 standup notes ❖ Sprint 1 retrospective notes ❖ Sprint 2 backlog ❖ Sprint 2 backlog	20-26
	<i>Implementation</i>	27
	<i>Sample Code</i>	27-30
	<i>Unit Testing</i> ❖ Tool used and its setup details ❖ Test cases statistics (#Units in the code and #unit tests written for each unit) ❖ Test case code	31-47

	<i>Screenshots of project</i>	48-53
	<i>Conclusion</i>	53

PREVIEW OF THE PROJECT

ABSTRACT

An online car booking management system is created to help manage hiring cabs with an online portal. The system is intended to help the cab agencies monitor, run and grow their companies. The user interface will contain all necessary customer information including personal details as well as details regarding needs of the customer. The agencies can use the system to efficiently provide cabs by easily assigning an available cab at a nearby location, to the customer. The cab booking system will be able to compile this information and make improvements to their quality of service. The system will help categorize locations, timings, frequent customers and more. This will inevitably organize and consolidate the data collected by the agencies and help manage their customer base and employees more effectively. Initially when the user installs the app, they will be asked to create an account to book the cab. Once the account is created the user can then continue to book a cab. All details of the available cabs will be provided to the customer so they can choose according to their necessities. They will also then be able to track and analyze how much time it will take the cab to reach their destination, how much distance there is from the departure point, the on-time satellite direction tracking and so on. Once the cab drops the customer at the desired destination, payment can be handled in both online and offline methods.

INTRODUCTION , MOTIVE AND NEED FOR PROJECT

Running a cab hiring service can be a hectic job to begin with. It must be properly organized from not only an administration point of view, but also from a driver and user point of view. For this, there must be an application which supports the different necessary credentials for each party. The application will also require a database for storing data. So, having an online cab management system is the only solution to address all these requirements. A software that is specific to cab hiring will make things more personalized and effective for the service providers. The online application creates a solid interface that can be accessed by anyone according to their position. There will be multiple drivers, hundreds of users and a whole team running the system who will all be needing to access the application accordingly. Our software system will organize and execute all such necessities.

Tools and Technologies USED:

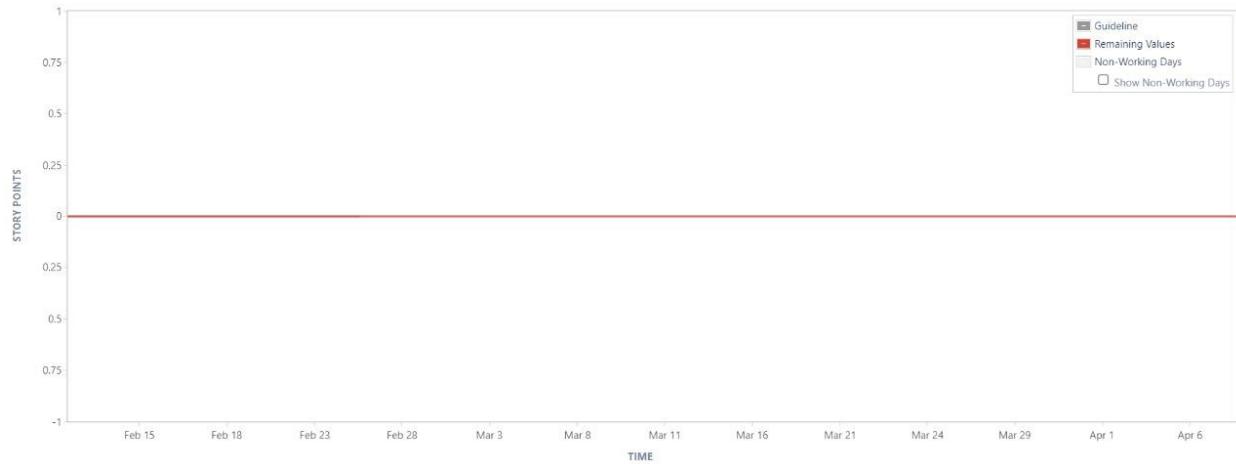
- Jira
- Visual Studio Code
- Xampp
- Squish - UI testing
- SQL workbench

Product Backlog:

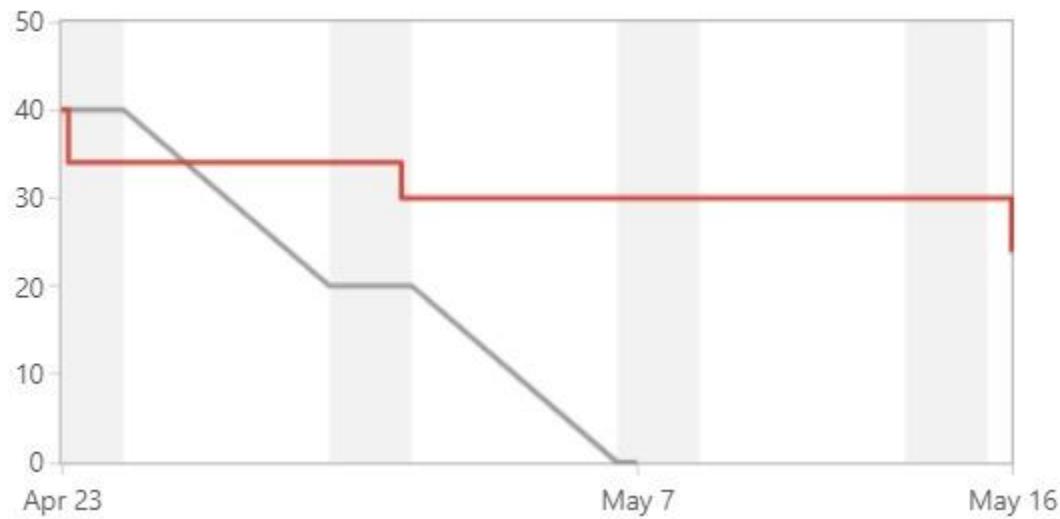
CHA Sprint 1 13 issues ACTIVE		...
	CHA-7 I want to see all the available cabs near me	unlabelled-CHA-1
	CHA-8 I want to choose what type of car I want to ride in.	unlabelled-CHA-1
	CHA-9 I want to see fares of all types of cars.	unlabelled-CHA-1
	CHA-10 I want to choose the cab according to my requirements.	unlabelled-CHA-1
	CHA-21 I want to select the reason for my cancellation	unlabelled-CHA-2
	CHA-12 As a driver, I should be able to create an account for confirming the ride	unlabelled-CHA-3
	CHA-13 As a driver, I need to edit my account or profile	unlabelled-CHA-3
	CHA-14 As a driver, I should be able to delete my account	unlabelled-CHA-3
	CHA-23 I should be able to create an account for booking cabs	unlabelled-CHA-4
	CHA-24 I need to edit my account or profile	unlabelled-CHA-4
	CHA-28 I should know what my cab's starting location is	unlabelled-CHA-5
	CHA-39 I should be able to see the route of my cab	unlabelled-CHA-5
	CHA-37 I should be able to update my availability status	unlabelled-CHA-6

REPORT OF PRODUCT BACKLOG FROM JIRA

Burndown chart Sprint 1



Burndown Chart Sprint 2



EPICS :

1.Hire a Cab

1.I have to make payment for that particular journey

- 1)Get total distance for the journey .
- 2)Calculate the total fare for the ride.
- 3>Show the different payments modes to the user.
- 4)If the user chooses online payment using UPI, take him to the online payment using UPI site.
- 5)If the user chooses online payment using debit/credit card , take the user to the particular payment gateway.
- 6)If the user chooses cash after ride, sent message to driver regarding this and also the total fare for the journey.

2.I want to see all the available cabs near me

- 1)Get the user's location
- 2)Get locations of all the cabs near the user.
- 3)show location of all the cabs to the user.
- 4)show min time it will take for a cab to reach the user.
- 5)show details of all the cabs to the user.

3.I want to choose what type of car I want to ride in

- 1) Get all types of cars available near the user.
- 2)show all the types cars available for the user.
- 3)show min distance from the user to each type of car.
- 4)Enquire to the user, how many passengers are there.
- 5)Suggest the best type of car with respect to the number of passengers.

4.I want to see fares of all types of cars

- 1)Get the user's location.
- 2) Get all types of cars available to the user.
- 3)Calculate the fare estimate with respect to each type of car.
- 4)Show all the fares of each types of car to the user.
- 5)Suggest the best type of car with respect to the fare.

5.I want to choose the cab according to my requirements

- 1)Get user's location
- 2)Get destination location
- 3)Get number of passengers

- 4)Verify what car type the user requires
- 5>Show the total ride fare ride

2.User Cancellation

1.I want to select the reason for my cancellation

2.As a non premium member , I want to cancel the cab till 6 min before it reaches my location without any fine.

- 1)Verify whether the user is a non-premium member.
- 2)Verify that a non-premium can cancel the cab till 6 min before it reaches his location without any fine.
- 3)Verify that if the non-premium member has canceled the cab after the given time, then notify him regarding the fine.
- 4)Verify that the user has not canceled the cab more than 3 times the same day .
- 5)Verify that if the user has canceled cab for more than 3 times the same day then send a notification regarding this ,that a fine would be imposed if he cancels 1 more time the same day.
- 6)verify that a confirmation is send to the cab driver regarding the cancellation .

3.I want to get a message or Email of the fine receipt and also the last date till which can be paid

- 1)Verify the email Id of the user.
- 2)Verify the phone number of the user.
- 3)Verify the date and the amount of the fine
- 4)Verify that the fine receipt is send to the user through mail and message.
- 5)Store a copy of the fine receipt in the DB and also send it to the payments section for further process.

4.I want to get notified about the cancellation fee and its terms and conditions before I finally cancel it

- 1)Verify passenger and cab details
- 2)Verify time of cab confirmation
- 3)Verify time of cancellation
- 4)Assign cancellation fee according to distance and time of cancellation
- 5)Send message to passenger regarding split up of cancellation fee

5.As a premium member , I want to cancel the cab till the last minute without any fine.

- 1)Verify whether the user is a premium member.
- 2)Verify that a premium can cancel the cab till the last minute without any fine.

- 3)Verify that the user has not canceled the cab more than 3 times the same day .
- 4)Verify that if the user has canceled cabs for more than 3 times the same day , then send a notification regarding this to the user that a fine would be imposed if he cancels the cab 3 more times the same day.
- 5)verify that a confirmation is sent to the cab driver regarding the cancellation .

3.Driver maintenance :

1.As a driver,I should be able to create an account for confirming the ride

- TASKS:
- 1)verify all the details are filled
- 2)verify all the details are filled correctly
- 3)verify that the phone number entered by the driver is correct by sending an OTP for confirmation
- 4)verify if the driver already exists or not
- 5)verify that the driver get a confirmation mail when the account has been set up

2.As a driver,I need to edit my account or profile

- TASKS:
- 1)can change the profile pic
- 2)can view personal information
- 3)can edit email, will receive confirmation to new email
- 4)edit phone number, should receive an otp
- 5)logout from my account

3.As a driver,I should be able to delete my account

- TASKS:
- 1)should ask to verify identity using OTP/email before deleting the account
- 2)verify that the driver does not have payment/fine backlog
- 3)should be able to deactivate the account
- 4)could be re-activated within 30 days
- 5)permanently delete the account from database after 30 days

4.As a driver,I need a forgot password

- TASKS:
- 1)prompt the driver to enter username
- 2)prompt the driver to enter mobile number
- 3)verify that the driver account already exists

- 4)can reset the password by sending a "reset password "mail
- to the driver
- 5)forgot username, they can use email ID/mobile number to get in

5.As a driver , I want to see my previous ride details

- 1)acquire previous details from the database
- 2)show the last location
- 3)show users details
- 4) show the trip details like fare, pickup and drop-off destinations
- 5)show total distance covered by the driver till this day

6.)As a driver, I want to see my transaction details

- 1)display the invoice of last 10 payments
- 2)display how the fare is calculated
- 3)should be able to download/print the receipt
- 4) display the total amount of money spent in the app
- 5)show the mode of payment used

4.User Maintenance :

1.I should be able to create an account for booking cabs

- 1)verify all the details are filled
- 2)verify all the details are filled correctly
- 3)verify that the phone number entered by the user is correct by sending an OTP for confirmation
- 4)verify if the user already exists or not
- 5)get a confirmation mail when the account has been set up

2.I need to edit my account or profile

- 1)can change the profile pic
- 2)can view personal information
- 3)can edit email,will receive confirmation to new email
- 4)edit phone number,should receive an otp
- 5)logout from my account

3.,I want to see my payment related details

- 1)display the invoice of last 10 payments
- 2)display how the fare is calculated

- 3)should be able to download/print the receipt
- 4) display the total amount of money spent in the app
- 5)show the mode of payment used

4.I need a forgot password /username option in case I forgot my credentials.

- 1)prompt the user to enter username
- 2)prompt the user to enter mobile number
- 3)verify that the user already exists
- 4)can reset the password by answering security questions setup by user
- 5)forgotten username,use email ID/mobile number to get in

5. I want to see my previous ride details

- 1)acquire previous details from the database
- 2)show the last location
- 3)show driver details.
- 4)show car details
- 5)show the trip details like fare,pickup,dropoff destination

6. As a user, I should be able to delete my account

- 1)should ask to verify identity using otp before deleting the account
- 2)should notify about outstanding payments before account deletion
- 3)should be able to deactivate the account
- 4)could be reactivated with in 30 days
- 5)permanently delete the account from database after 30 days

5.Tracking :

1.I should know what my cab's starting location is

- 1a)Verify the cab driver has switched on live location
- 1b)Verify the user has switched on live location
- 1c)Verify the starting location is closest to me
- 1d>Show the path from cab driver to user.
- 1e)show the distance between the cab driver and the user.

2.I should be able to see the route of my cab

- 3a)Verify the current cab location
- 3b)Verify the distance from the current cab location to starting location
- 3c)Verify the time taken to reach current location from starting location
- 3d)Check current time
- 3e)Compare current time to time taken to reach current location to get starting time

3.I should be able to check at what time the cab will reach my destination

- 2a)Verify the time at which the driver started from the starting location
- 2b)verify the current location of the cab
- 2c)Find the total duration from cab location to user location
- 2d)Verify the driver has taken the shortest path
- 2e) add that calculated time with current time and show the final estimated time for the cab driver to reach the user.

4.I should be able to see the total duration it will take for my cab to reach me

- 4a)Verify the current location of the cab
- 4b)Verify the total distance from the current location to the user location
- 4c)Verify the route taken by the cab is the shortest path
- 4d)Check for traffic on the route
- 4e)Verify the time the cab started and current time

5. I should be able to see my cab moving

- 5a)Verify the cab driver has live location on
- 5b)Verify the live location is updating on the app
- 5c)Verify the map is updating on the driver's side
- 5d)Verify the map is updating on the user's side

6.Driver cancellation

1. I should be able to update my availability status

- Contact admin regarding availability status
- Update status on application
- Availability status should be visible to users and admin
- If unavailable, driver should not appear on user's interface

2. I should be able to see user's pickup location

- Contact the driver to inquire about cancellation
- Verify if there are any issues with the passengers current location
- Verify if there are any issues with the user's chosen destination
- Verify if there are more passengers than initially mentioned
- Send message to user about cancellation

3. As a driver , I may be busy with some other work

- Verify why driver wants to cancel drive
- User can able to get another ride
- Giving discount to user
- Verify the driver previous records
- Verify driver status

4.User not present at the pickup location

- 1) verify pickup address
- 2) verify user location
- 3) verify phone number
- 4) verify otp given by user
- 5) verify destination point

5. I am unable to contact user

- 1) verify the user location
- 2) get the user's contact number
- 3) get the user's mail id
- 4) contact user by calling through mobile call or sending message
- 5) send a message to the user specifying why the driver canceled

Epic 1 (As a user I have to maintain my account)

Story 1 (As a user, I should be able to create an account for booking cabs)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Akshara S Nair	High	2
Create Backend with Python		High	3
Create Database in MySQL Workbench		Medium	1
Link UI with database		High	1
Verify user credentials		Medium	1

Story 2 (As a user, I need to edit my account or profile.)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Akshara S Nair	High	2
Create Backend with Python		High	2
Create Database in MySQL Workbench		Medium	1
Link UI with database		Medium	1

Epic 2 (As a user I want to hire a cab)

Story 1 (As a user ,I want to see fares of all types of cars..)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Aadith S	High	3
Create Backend with Python		High	3
Create Database in MySQL Workbench		High	1
Link UI with database		Medium	1
Task 5			

Story 2 (As a user ,I want to choose what type of car I want to ride in.

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Aadith S	High	2
Create Backend with Python		Medium	3
Create Database in MySQL Workbench		High	1.5
Link UI with database		Medium	1
Task 5			

Story 3(As a user ,I want to choose the cab according to my requirements.

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Aadith S	High	2
Create Backend with Python		High	2
Create Database in MySQL Workbench		Medium	2
Link UI with database		High	1
Task 5			

Epic 3 (As a user I have to track my cab)

Story 1 (As a user , I should be able to see the route of my cab)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Radhika Rajalal	High	2
Create Backend with Python		High	2
Create Database in MySQL Workbench		Medium	1
Link UI with database		High	1

Epic 4(As a user, I want to cancel the hired cab.)

Story 1 (As a user , I should be able to select a reason for cab cancelling)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Radhika Rajalal	Medium	2
Create Backend with Python		High	1
Create Database in MySQL Workbench		Medium	1
Link UI with database		High	1

Epic 5 (As a cab driver , I want to maintain my account.)

Story 1 (As a cab driver, I should be able to create an account for booking cabs)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Gokuldas D	High	3
Create Backend with Python		High	2
Create Database in MySQL Workbench		Medium	1
Link UI with database		Medium	1
Verify user credentials		High	2

Story 2 (As a driver, I need to edit my account or profile.)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Gokuldas D	High	2
Create Backend with Python		High	2
Create Database in MySQL Workbench		High	1
Link UI with database		Medium	1

Epic 6 (AS A DRIVER I WANT CANCEL RIDE)

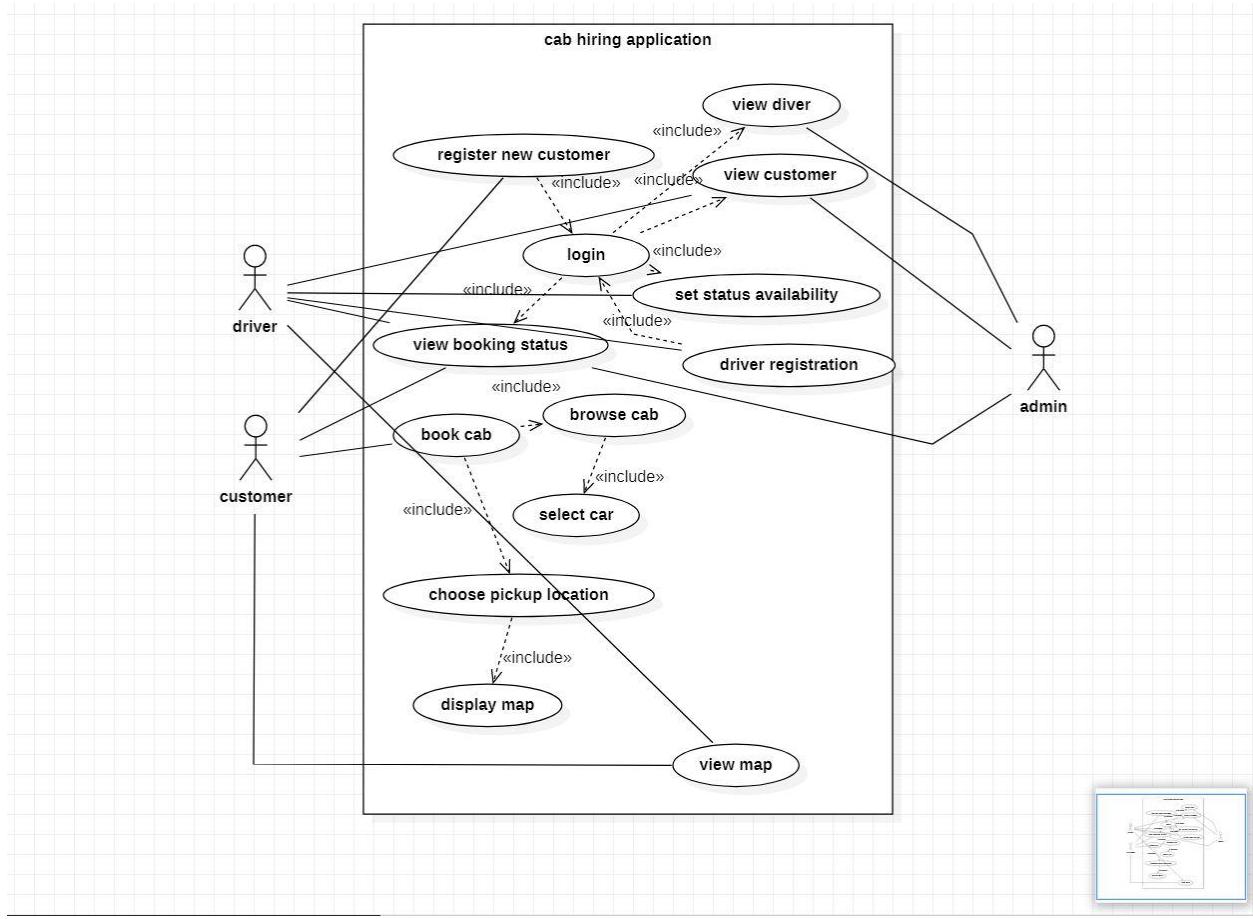
Story 1 (As a driver,I should be able to update my availability status)

Task	Member	Priority	Estimate (Hours)
Create UI with Python	Harshvardhan	High	3
Create Backend with Python		Medium	2
Create Database in MySQL Workbench		Medium	1
Link UI with database		High	1

Story 2(As a driver,I should be able to give reason for cancellation)

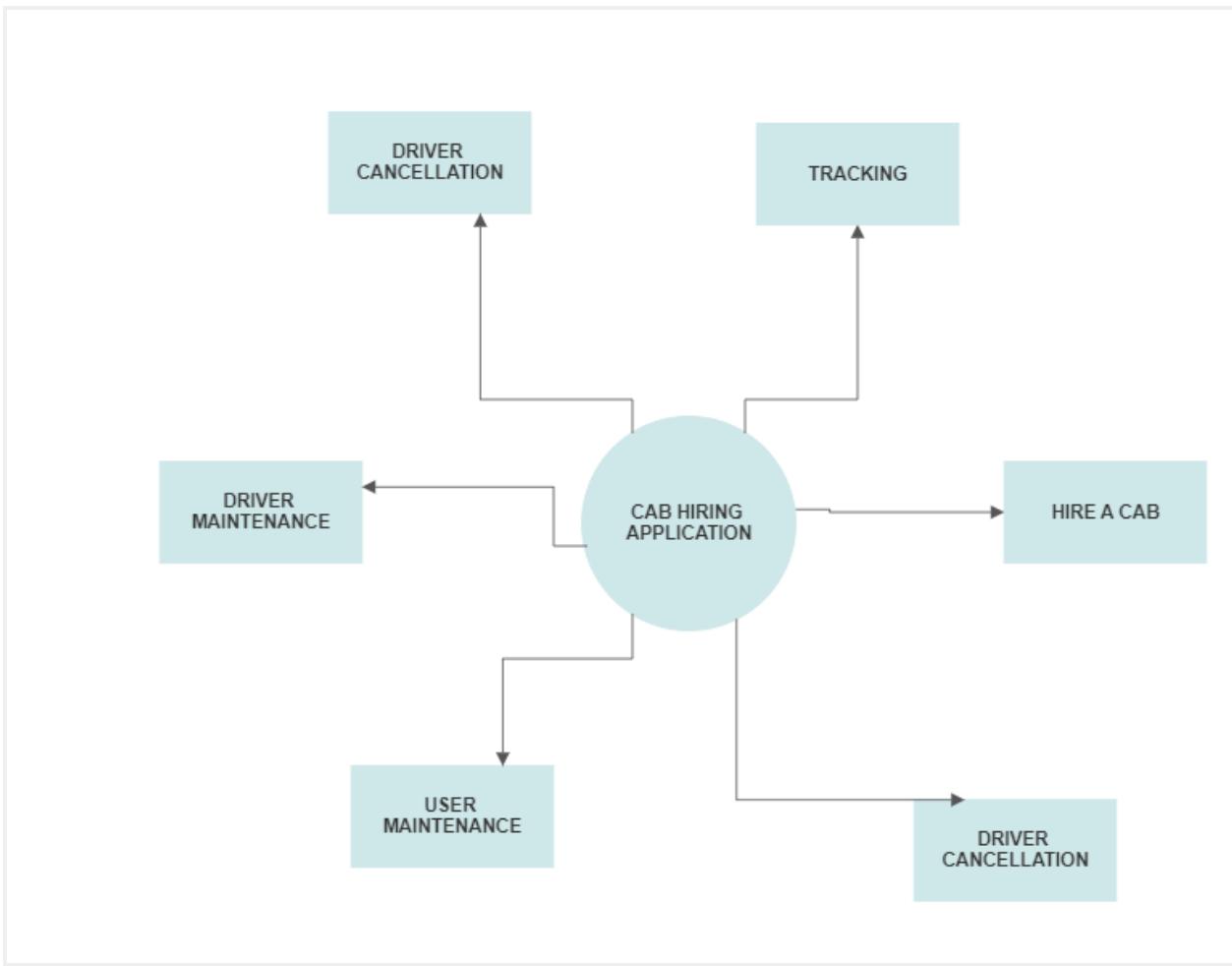
Task	Member	Priority	Estimate (Hours)
Create UI with Python	Harshvardhan	High	2
Create Backend with Python		High	3
Create Database in MySQL Workbench		Medium	1
Link UI with database		High	2

UseCase diagram:

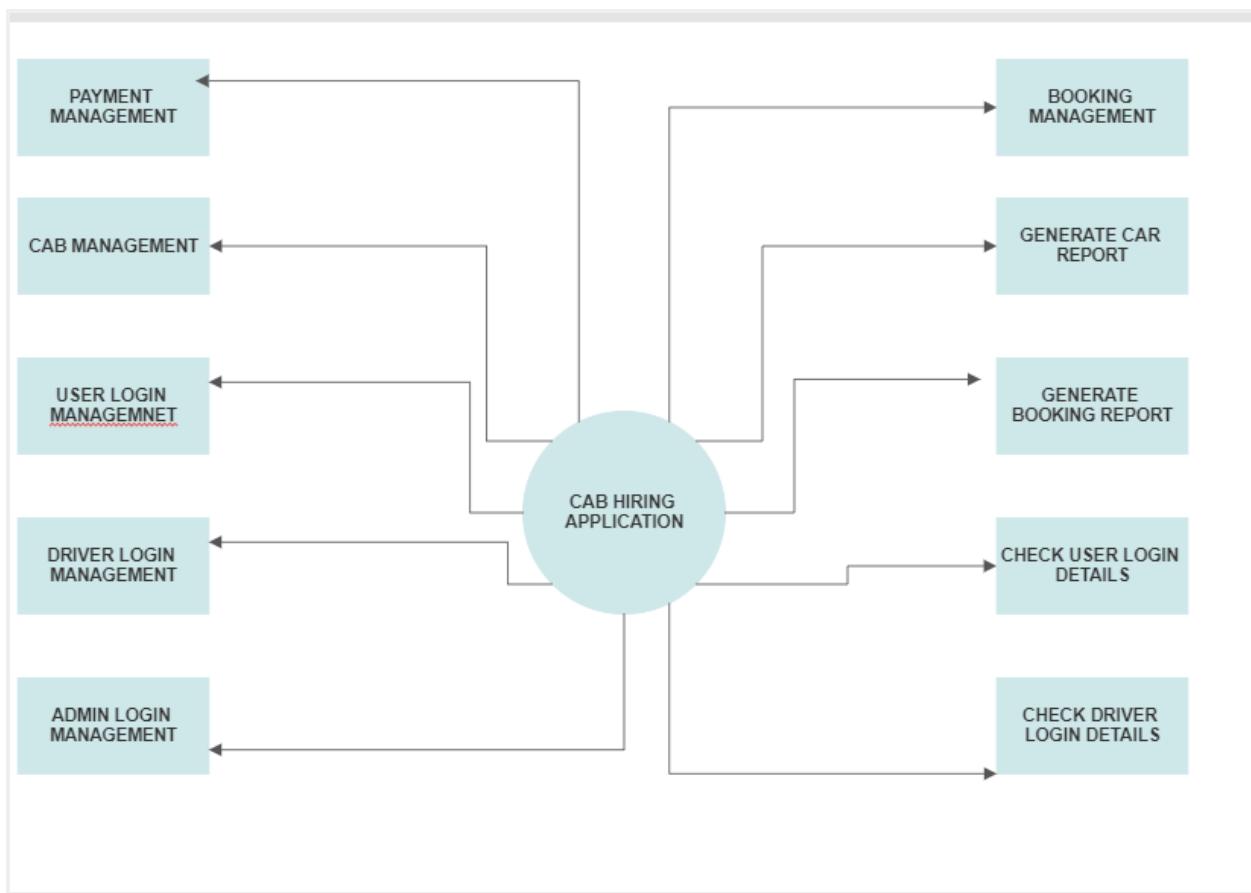


DFD:

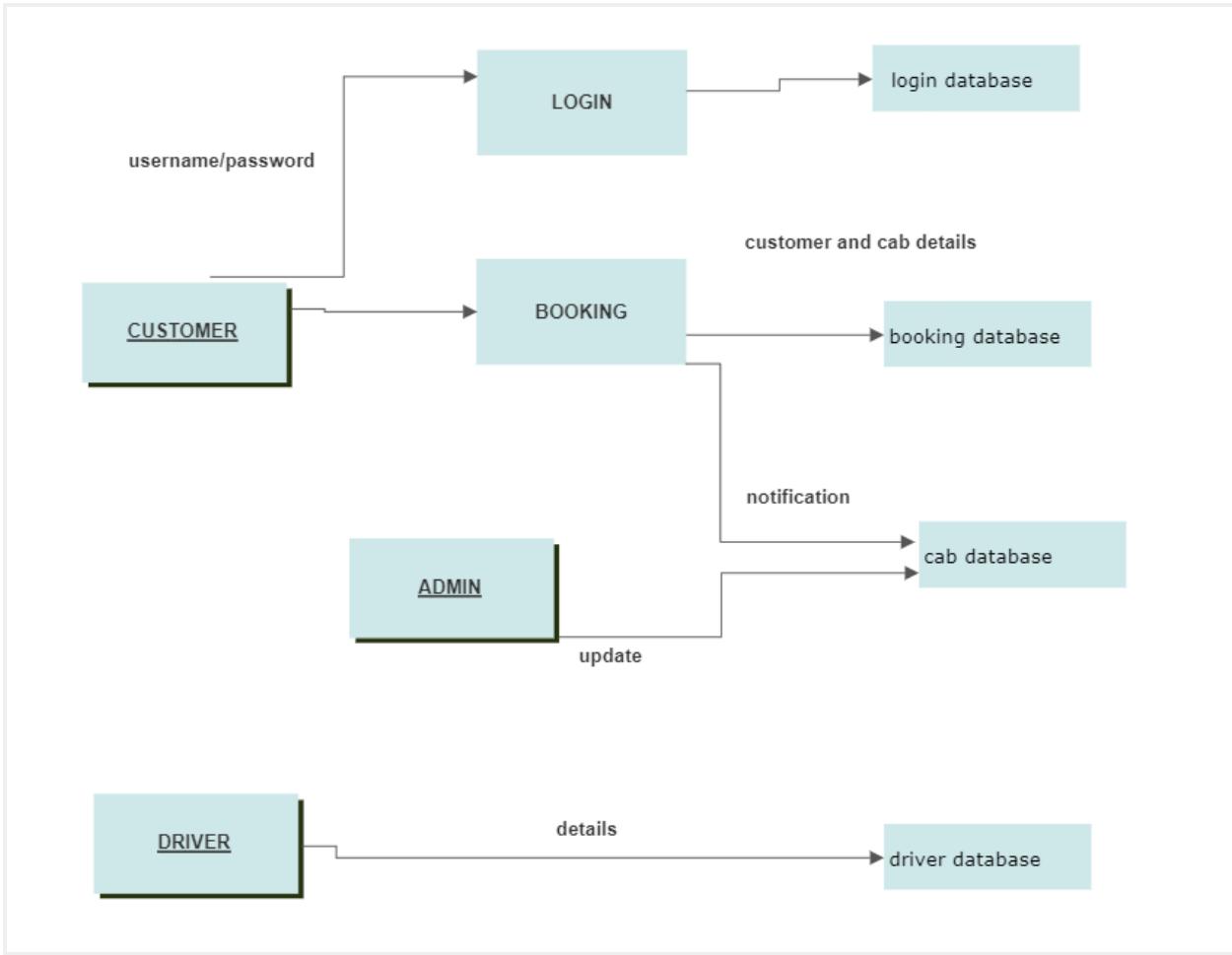
LEVEL 0



LEVEL 1



LEVEL 3



Online tool used : <https://cloud.smartdraw.com/?nsu=1>

SCRUM ACTIVITIES

SPRINT 1 BACKLOG

CHA Sprint 1 13 issues ACTIVE		...
	CHA-7 I want to see all the available cabs near me	unlabelled-CHA-1
	CHA-8 I want to choose what type of car I want to ride in.	unlabelled-CHA-1
	CHA-9 I want to see fares of all types of cars.	unlabelled-CHA-1
	CHA-10 I want to choose the cab according to my requirements.	unlabelled-CHA-1
	CHA-21 I want to select the reason for my cancellation	unlabelled-CHA-2
	CHA-12 As a driver, I should be able to create an account for confirming the ride	unlabelled-CHA-3
	CHA-13 As a driver, I need to edit my account or profile	unlabelled-CHA-3
	CHA-14 As a driver, I should be able to delete my account	unlabelled-CHA-3
	CHA-23 I should be able to create an account for booking cabs	unlabelled-CHA-4
	CHA-24 I need to edit my account or profile.	unlabelled-CHA-4
	CHA-26 I should know what my cab's starting location is	unlabelled-CHA-5
	CHA-30 I should be able to see the route of my cab	unlabelled-CHA-5
	CHA-37 I should be able to update my availability status	unlabelled-CHA-6

SPRINT 1 STANDUP NOTES

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	N/A	
	What will you do today?	Discussed and created epics, user stories and tasks	
	Are there any impediments in your way?	Took some time to understand how to assign tasks	
Akshara S Nair	What did you do yesterday?	N/A	
	What will you do today?	Discussed and created epics, user stories and tasks	
	Are there any impediments in your way?	Took some time to understand how to assign tasks	
Gokuldas D	What did you do yesterday?	N/A	
	What will you do today?	Discussed and created epics, user stories and tasks	
	Are there any impediments in your way?	Took some time to understand how to assign tasks	
P Harshavardhan	What did you do yesterday?	N/A	
	What will you do today?	Discussed and created epics, user stories and tasks	
	Are there any impediments in your way?	Took some time to understand how to assign tasks	
Radhika Rajalal	What did you do yesterday?	N/A	
	What will you do today?	Discussed and created epics, user stories and tasks	
	Are there any impediments in your way?	Took some time to understand how to assign tasks	

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	Planned how to create UI for cab hiring	
	What will you do today?	Finish half of the UI for hiring a cab	
	Are there any impediments in your way?	–	
Akshara S Nair	What did you do yesterday?	Planned UI for user account	
	What will you do today?	Start creating UI for user account	
	Are there any impediments in your way?	–	
Gokuldas D	What did you do yesterday?	Planned UI for driver account	
	What will you do today?	Start creating UI for user account	
	Are there any impediments in your way?	–	
P Harshavardhan	What did you do yesterday?	Planned how to implement driver cancellation program	
	What will you do today?	Finish part of driver ride cancelation program	
	Are there any impediments in your way?	–	
Radhika Rajalal	What did you do yesterday?	Planned how to implement user cab tracking program	
	What will you do today?	Work on cab tracking program using latitude and longitude	
	Are there any impediments in your way?	Getting the destination to show on the map	

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	Finished UI for cab hiring	
	What will you do today?	Start backend python programming for UI of cab hiring	
	Are there any impediments in your way?	–	
Akshara S Nair	What did you do yesterday?	Finished UI for user account maintenance	
	What will you do today?	Start backend python programming	
	Are there any impediments in your way?	Connecting frontend and backend	
Gokuldas D	What did you do yesterday?	Finished UI for user account maintenance	
	What will you do today?	Start backend programming	
	Are there any impediments in your way?	Connecting frontend and backend	
P Harshavardhan	What did you do yesterday?	Worked on driver cancellation program	
	What will you do today?	Worked on driver cancellation program	
	Are there any impediments in your way?	–	

Radhika Rajalal	What did you do yesterday?	Planned user ride cancellation program	
	What will you do today?	Start user ride cancellation program	
	Are there any impediments in your way?	–	

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	Finished half of the backend python programming	
	What will you do today?	Start database implementation for cab hiring	
	Are there any impediments in your way?	Few issues with linking frontend and backend	
Akshara S Nair	What did you do yesterday?	Finished half of the backend python programming	
	What will you do today?	Start database implementation for user account	
	Are there any impediments in your way?	Separating 2 sets of data coming from the same UI	
Gokuldas D	What did you do yesterday?	Finished half of the backend python programming	
	What will you do today?	Start database implementation for driver account	
	Are there any impediments in your way?	Separating 2 sets of data coming from the same UI	
P Harshavardhan	What did you do yesterday?	Worked on driver cancellation program	
	What will you do today?	Worked on driver cancellation program	
	Are there any impediments in your way?	Connecting UI and backend	
Radhika Rajalal	What did you do yesterday?	Worked on user cancellation program	
	What will you do today?	Worked on cab tracking program	
	Are there any impediments in your way?	Getting cab location of map	

SPRINT 1 RETROSPECTIVE NOTES :



SPRINT 2 BACKLOG

CHA-18 As a non-premium member, I want to cancel the cab till 6 min before it reaches my location without any fine.	unlabelled-CHA-2	4
CHA-20 I want to get notified about the cancellation fee and its terms and conditions before I finally cancel it.	unlabelled-CHA-2	2
CHA-29 I should be able to check at what time the cab will reach my destination	unlabelled-CHA-5	2
CHA-31 I should be able to see the total duration it will take for my cab to reach me	unlabelled-CHA-5	2
CHA-22 As a premium member, I want to cancel the cab till the last minute without any fine.	unlabelled-CHA-2	2
CHA-15 As a driver , I need a forgot password incase I forgot the password	unlabelled-CHA-3	2

SPRINT 2 STANDUP NOTES

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	Worked on backend python programming	
	What will you do today?	Work on backend python programming	
	Are there any impediments in your way?	N/A	
Akshara S Nair	What did you do yesterday?	Worked on backend python programming	
	What will you do today?	Work on backend python programming	
	Are there any impediments in your way?	Separating 2 sets of data coming from the same UI	
Gokuldas D	What did you do yesterday?	Worked on backend python programming	
	What will you do today?	Work on backend python programming	
	Are there any impediments in your way?	Separating 2 sets of data coming from the same UI	
P Harshavardhan	What did you do yesterday?	Worked on driver cancellation program	
	What will you do today?	Work on driver cancellation program	
	Are there any impediments in your way?	Connecting UI and backend	
Radhika Rajalal	What did you do yesterday?	Worked on user cancellation program	
	What will you do today?	Worked on cab tracking program	
	Are there any impediments in your way?	Getting cab location of map	

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	Worked on backend python programming	
	What will you do today?	Work on sql connection	
	Are there any impediments in your way?	N/A	
Akshara S Nair	What did you do yesterday?	Worked on backend python programming	
	What will you do today?	Work on sql connection	
	Are there any impediments in your way?	N/A	

Gokuldas D	What did you do yesterday?	Worked on backend python programming	
	What will you do today?	Work on sql connection	
	Are there any impediments in your way?	N/A	
P Harshavardhan	What did you do yesterday?	Worked on driver cancellation program	
	What will you do today?	Work on sql connection	
	Are there any impediments in your way?	N/A	
Radhika Rajalal	What did you do yesterday?	Worked on cab tracking program	
	What will you do today?	Work of Sql for user cancellation	
	Are there any impediments in your way?	Getting cab route	

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	Worked on validation for data	
	What will you do today?	Work on cloud connection	
	Are there any impediments in your way?	N/A	
Akshara S Nair	What did you do yesterday?	Worked on documentation	
	What will you do today?	Worked on Unit Testing	
	Are there any impediments in your way?	N/A	
Gokuldas D	What did you do yesterday?	Researched how to do testing	
	What will you do today?	Worked with squish testing	
	Are there any impediments in your way?	Learning squish	
P Harshavardhan	What did you do yesterday?	Worked on driver cancellation program	
	What will you do today?	Worked on unit testing	
	Are there any impediments in your way?	N/A	
Radhika Rajalal	What did you do yesterday?	Worked on connecting map and cab tracking	
	What will you do today?	Help with testing	
	Are there any impediments in your way?	Learning squish	

Name	Question	Answer	Sign
Aadith S	What did you do yesterday?	Worked on cloud connection and final code changes	
	What will you do today?	Make final code changes	
	Are there any impediments in your way?	N/A	

Akshara S Nair	What did you do yesterday?	Worked on documentation	
	What will you do today?	Make final code changes	
	Are there any impediments in your way?	N/A	
Gokuldas D	What did you do yesterday?	Worked on testing on squish	
	What will you do today?	Work on testing on squish	
	Are there any impediments in your way?	N/A	
P Harshavardhan	What did you do yesterday?	Worked on unit testing	
	What will you do today?	Work on documentation	
	Are there any impediments in your way?	N/A	
Radhika Rajalal	What did you do yesterday?	Helped with testing	
	What will you do today?	Make final code changes	
	Are there any impediments in your way?	N/A	

IMPLEMENTATION

We have primarily used Pycharm for Python for coding. The entire frontend and backend programming of our software is done using python. All of our UI and its functionalities have been done using python. In addition we've used Visual Studio Code in implementing various aspects of our code. In order to incorporate the database of our software, we've used mySql through Sql Workbench. Workbench updates any data additions, changes etc. that occur through the UI of the actual software. For testing our software, we have used Squish, a testing software in which we created test cases to run our program. Squish allows us to see where there may be issues and which test cases have passed or failed.

SAMPLE CODE :

```
Users > Akshara S Nair > OneDrive > Desktop > userloginandsignup.py > ...
1  from time import time
2  from tkinter import *
3  import tkinter as tk
4  from tkinter import ttk, messagebox
5  import mysql
6  import mysql.connector
7  from PIL import ImageTk, Image
8  import re
9
10 from tkintertmapview import TkinterMapView
11
12
13 def change_frame():
14     regisframe.forget()
15     loginframe.pack(expand=True, padx=30, pady=30)
16
17
18
19 def Add():
20     first_name= firstnameEntry.get()
21     last_name = lastnameEntry.get()
22     age = ageEntry.get()
23     city = cityEntry.get()
24     addre = addeEntry.get()
25     phone = phoneEntry.get()
26     mail = mailEntry.get()
27     user_name = usernameEntry.get()
28     password = passwordEntry.get()
```

```

if(driverfirstnameEntry.get() == "" or driverlastnameEntry.get() == "" or drivenameEntry.get() == "" or drivercityEntry.get() == ""):
    messagebox.showinfo("Information","Please enter all the details!")
elif(len(first_name) < 2 or len(first_name) > 15 or is_namematch == False):
    messagebox.showinfo("Information","Please enter a proper first name")
elif(len(last_name)<1 or len(last_name)>15 or is_lastnamematch == False):
    messagebox.showinfo("Information","Please enter a proper last name")
elif(len(driverphoneEntry.get()) != 10 ):
    messagebox.showinfo("Information","Please enter a valid phone number")
elif(ageint < 18):
    messagebox.showinfo("Information","Please enter a proper age")
elif(carmatch == False):
    messagebox.showinfo("Information","Please enter a valid car type [Micro,Sedan,SUV]")
elif(len(driverusernameEntry.get())<6):
    messagebox.showinfo("Information","Username must atleast 6 characters")
elif(len(driverpasswordEntry.get()) < 6):
    messagebox.showinfo("Information","Password must be atleast 6 characters long!")

```

```

mysqldb = mysql.connector.connect(host="localhost", user="root", password="",database="cabapp",auth_plugin='mysql_native_password')
mycursor = mysqldb.cursor()

try:
    sql = "INSERT INTO driver_info (driver_first_name,driver_last_name,driver_age,driver_city,driver_adde,driver_phone,driver_car
    val = (first_name,last_name,age,city,adde,driver_phone,car_type,user_name,password)
    sql1 = "INSERT INTO driver_status (driver_user_name,driver_car_type,driver_status_now) VALUES (%s,%s,%s)"
    val1 = (user_name,car_type,status)
    mycursor.execute(sql, val)
    mycursor.execute(sql1,val1)
    mysqldb.commit()
    lastid = mycursor.lastrowid
    messagebox.showinfo("information", "Driver inserted successfully...")
    driverfirstnameEntry.delete(0, END)
    driverlastnameEntry.delete(0, END)
    drivenameEntry.delete(0, END)
    drivercityEntry.delete(0, END)
    driveraddeEntry.delete(0 , END)
    driverphoneEntry.delete(0, END)
    drivertypeEntry.delete(0,END)
    driverusernameEntry.delete(0,END)
    driverpasswordEntry.delete(0,END)
    firstnameEntry.focus_set()
except Exception as e:
    print(e)
    mysqldb.rollback()

```

```

if (len(row_set2) == 3):
    userlastridesbookingid.set("Booking ID")
    print(userlastridesbookingid)
    userlastridesbookingid.set("Booking ID")
    print(userlastridesbookingid)
    userlastridesdriverusername.set("User username")
    userlastridespickuploc.set("Pickup")
    userlastridesdroploc.set("Drop")
    userlastridesamount.set("Amount")
    userlastridesdriverusername1.set(row_set2[0][1])
    userlastridesdriverusername2.set(row_set2[1][1])
    userlastridesdriverusername3.set(row_set2[2][1])
    userlastridespickuploc1.set(row_set2[0][3])
    userlastridespickuploc2.set(row_set2[1][3])
    userlastridespickuploc3.set(row_set2[2][3])
    userlastridesdroploc1.set(row_set2[0][4])
    userlastridesdroploc2.set(row_set2[1][4])
    userlastridesdroploc3.set(row_set2[2][4])
    userlastridesamount1.set(row_set2[0][7])
    userlastridesamount2.set(row_set2[1][7])
    userlastridesamount3.set(row_set2[2][7])
    userlastridesbookingid1.set(row_set2[0][6])
    userlastridesbookingid2.set(row_set2[1][6])
    userlastridesbookingid3.set(row_set2[2][6])

```

```

driveageEntry.grid(row=3,column=1,pady=10,padx=20)
drivercitylabel = Label(driverregisframe,text="City")
drivercitylabel.grid(row=4,column=0,pady=10,padx=20)
drivercityEntry = Entry(driverregisframe,width=30)
drivercityEntry.grid(row=4,column=1,pady=10,padx=20)
driveraddelabel = Label(driverregisframe,text="Address")
driveraddelabel.grid(row=5,column=0,pady=10,padx=20)
driveraddeEntry = Entry(driverregisframe,width=30)
driveraddeEntry.grid(row=5,column=1,pady=10,padx=20)
driverphonelabel = Label(driverregisframe,text="Mobile")
driverphonelabel.grid(row=6,column=0,pady=10,padx=20)
driverphoneEntry = Entry(driverregisframe,width=30)
driverphoneEntry.grid(row=6,column=1,pady=10,padx=20)
drivercartype = Label(driverregisframe,text="Car Type")
drivercartype.grid(row=7,column=0,pady=10,padx=20)
drivercartypeentry = Entry(driverregisframe,width=30)
drivercartypeentry.grid(row=7,column=1,pady=10,padx=20)
driverusernamelabel = Label(driverregisframe,text="Username")
driverusernamelabel.grid(row=8,column=0,pady=10,padx=20)
driverusernameEntry = Entry(driverregisframe,width=30)
driverusernameEntry.grid(row=8,column=1,pady=10,padx=20)
driverpasswordlabel = Label(driverregisframe,text="Password")
driverpasswordlabel.grid(row=9,column=0,pady=10,padx=20)
driverpasswordEntry = Entry(driverregisframe,width=30)
driverpasswordEntry.grid(row=9,column=1,pady=10,padx=20)
driversignupbutton = Button(driverregisframe,text="Sign Up",command=driver_Add)
driversignupbutton.grid(row=10,column=0,pady=10,padx=20)

```

```
frame = LabelFrame(root,text="Cab Booking",padx=10,pady=20)
frame1 = LabelFrame(frame,text="Customer Info",padx=10,pady=20)
frame2 = LabelFrame(frame,text="Route Info",padx=10,pady=20)
frame3 = LabelFrame(frame,text="Car Selection",padx=10,pady=20)
frame4 = LabelFrame(frame,text="Receipt",padx=10,pady=20)

frame3.pack(side=BOTTOM , expand = True, fill = BOTH)
frame1.pack(side=LEFT, expand = True, fill = BOTH)
frame2.pack(side=LEFT, expand = True, fill = BOTH)
frame4.pack(side=LEFT, expand = True, fill = BOTH)

mylabel00 = Label(frame1,text="First Name",pady=10,padx=10)
mylabel01 = Label(frame1,text="Last Name",pady=10,padx=10)
mylabel02 = Label(frame1,text="Address",pady=10,padx=10)
mylabel03 = Label(frame1,text="Mobile",pady=10,padx=10)
mylabel04 = Label(frame1,text="Email",pady=10,padx=10)
mylabel05 = Label(frame1,text="Emergency Num",pady=10,padx=10)
e00 = Entry(frame1,width=30)
e00.insert(0,'Optional')
e01 = Entry(frame1,width=30)
e01.insert(0,'Optional')
```

TESTING

Unit Testing :

FORM 1					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
USER option	go to User Signup page	can only choose one	Aadith	Akshara	
Driver option	go to driver signup page	can only choose one			

USER signup FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
First Name	maximum 15 characters No special characters minimum 2 characters	cannot be left blank	Akshara	Aadith	done done done
Last Name	minimum 1 character maximum 15 characters no special characters permitted	cannot be left blank			done done done
Age	must be a number	between 18 & 100			fail
City	must not be a special character/number				fail
Address	can include numbers can include special characters				done done
Email	must include domain name	shouldn't exist in database			done
UserName	must be minimum of 6 characters	must be unique(should not match any from the database)			done
Password	must be minimum of 6 characters must include one special character must include one number				done fail fail
SIGN UP button	must redirect to Dashboard form				done
Back button	must redirect to form 1				done
Already have an account button	redirects to login form				done

USER LOGIN FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
UserName	must be minimum of 6 characters	must match the username from the database	Akshara	Aadith	done
Password	must be minimum of 6 characters must include one special character must include one number must match the password in the database				done fail fail done
Login Button	must got to booking page				done
Back Button	must go to first form -> selection of user or driver				done

USER CAB CANCELLATION FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
Username	checks if user exists in database		Radhika	Akshara	done
Reason	must select reason from the given option	can only choose one			done
Cancel ride button	check if the booking exists checks payment details redirects to user dashboard	confirm if refund is necessary			done fail done

DRIVER signup FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
First Name	maximum 15 characters No special characters	cannot be left blank	Gokuldas	Radhika	done
	minimum 2 characters				done
Last Name	minimum 1 character maximum 15 characters	cannot be left blank			done
Age	must be a number	should be between 18 & 100			done
City	must not be a special character	should be in India			done
Address	must not include special characters	can include numbers			done
CarType	must select one from the given options	can only choose one (per booking)			done
UserName	must be minimum of 6 characters	must be unique(should not match any from the database)			done
Password	must be minimum of 6 characters must include one special character must include one number				done
SIGN UP button	must redirect to Dashboard form				fail
Back button	must redirect to form 1				done
Already have an account button	redirects to login form				done

DRIVER LOGIN FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
UserName	must be minimum of 6 characters	must match the username from the database	Gokuldas	Harsha Vardhan	done
Password	must be minimum of 6 characters must include one special character must include one number	must match the password in the database			done fail fail
Login Button	must go to booking page				done
Back Button	must go to first form -> selection of user or driver				done

DRIVER DASHBOARD FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
Hire a cab button	Redirects to booking form	can only choose one	Radhika	Gokuldas	done
Cancel a ride button	Redirects to cancellation form	can only choose one			done
Log out	redirects to form 1	can only choose one			done

DRIVER CAB CANCELLATION FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
Username	checks if user exists in database		Harshavardhan	Radhika	done
Reason	must select reason from the given option	can only choose one option			done
Cancel ride button	check if the booking exists				done
	checks payment details				done
	redirects to DRIVER dashboard				done

BOOKING FORM					
Control Name	Validation	Added Details	Developer name	Tester Name	Status
Select Start Location	Only select from the options on the options list	Starting location cannot be the same as dropping location	Aadith	Gokul	done
Select Dropping Location	Only select from the options on the options list	Dropping location cannot be the same as starting location			done
Select type of car	Only select 1 type of car from 3 options as radio buttons	1)Base Price must change when that particular type of car is selected 2)Charge per km must change when the particular type of car is selected.			done
Get Bill Button					done
Book button	if car available prints message booked redirects to user dashboard opens tracking page				done
Home button	redirects to user dashboard				done
Log Out	redirects to form 1				done

UI TESTING

Tool used and its setup details : Squish

The screenshot shows the Squish IDE interface with two windows side-by-side.

Left Window (Test Case Editor):

- File Path: C:\Users\Akshara S Nair\suite_test_suite4\tst_test1313\test.py - Squish IDE 7.0.0
- Menu: File, Edit, Refactoring, Source, Navigate, Search, Run, Window, Help
- Toolbars: Standard, Squish, Test Management, Spy, Test Debugging, Global Scripts
- Test Suites: suite_test_suite4
- Test Cases: tst_test1313
- Code View:

```
def main():
    startApplication("userLoginAndSignup")
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2)["class"], "Button")
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).enabled, True)
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).type, "Button")
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).text, "I am a User")
    test.compare(waitForObjectExists(names.o_Button).enabled, True)
    test.compare(waitForObjectExists(names.o_Button)[ "class"], "Button")
    test.compare(waitForObjectExists(names.o_Button).text, "I am a Driver")
```
- Test Case Resources: Scripts, Test Data, VPs, Gestures, test.py
- Test Suite Resources: Scripts, Test Data, VPs, Gestures, __pycache__, names.py

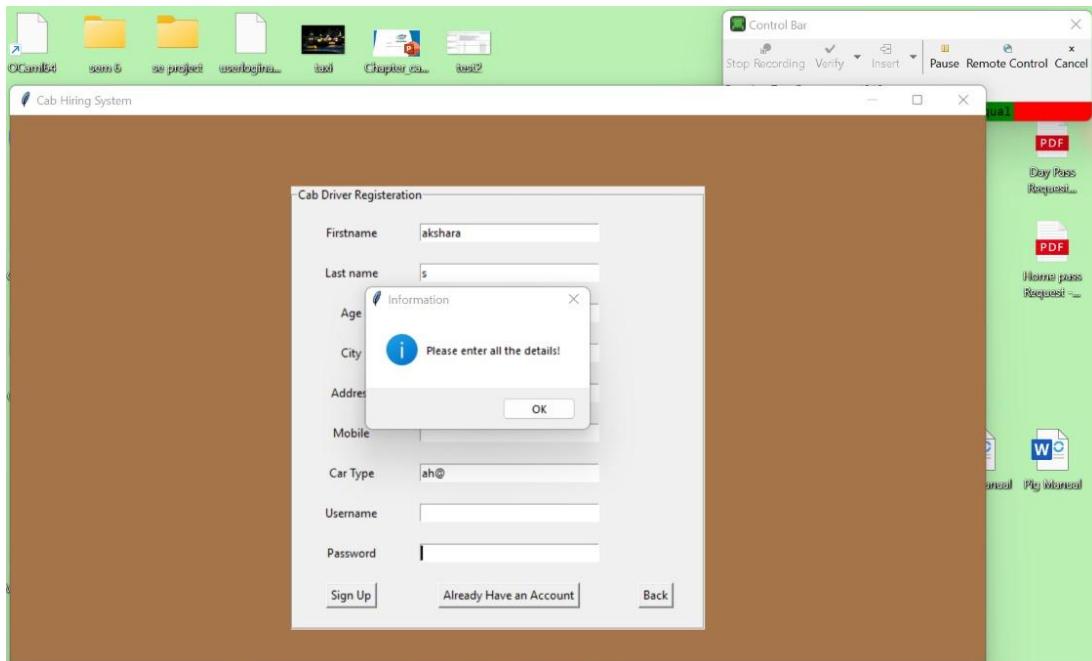
Right Window (Test Results):

- Test Results: Test Summary, Runner/Server Log, Test Description
- Recent Test Results
- Table:

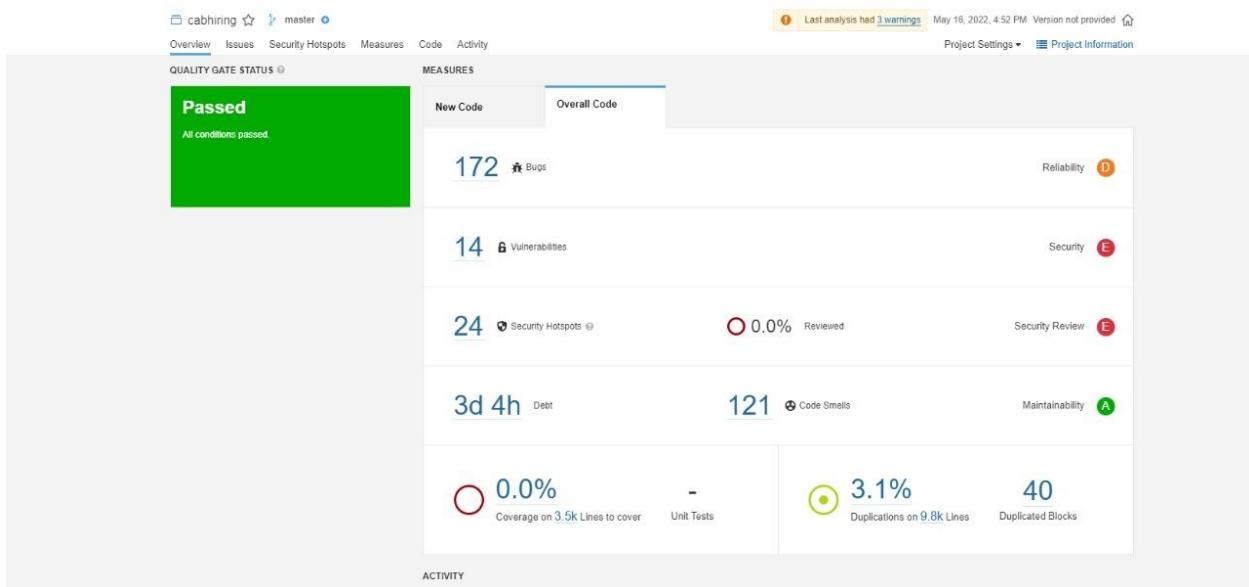
Result	Message	Time
> TestCase	tst_test1313	16 May 2022 at 2:12:35 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:45 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:45 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:45 PM +05:30
> Fail	Comparison	16 May 2022 at 2:12:45 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:45 PM +05:30
> Fail	Comparison	16 May 2022 at 2:12:45 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:46 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:46 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:46 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:46 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:46 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:46 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:46 PM +05:30
> Fail	Comparison	16 May 2022 at 2:12:47 PM +05:30
> Pass	Comparison	16 May 2022 at 2:12:48 PM +05:30

Second Screenshot (Test Results):

- File Path: C:\Users\Akshara S Nair\suite_test_suite4\tst_test1313\test.py - Squish IDE 7.0.0
- Test Suite Name: suite_test_suite4
- Duration: 53 seconds
- Number of Test Cases: 1
- Number of Features: 0
- Number of Scenarios: 0
- Number of Skipped Steps: 0
- Number of Results: 138
- Number of Passes: 114
- Number of Fails: 24
- Number of Errors: 0
- Number of Fatal: 0
- Number of Warnings: 0



SonarQube



Test Case Code

```
import names

def main():
    startApplication("userloginandsignup")
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2)["class"], "Button")
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).enabled, True)
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).type, "Button")
    test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).text, "I am a User")
    test.compare(waitForObjectExists(names.o_Button).enabled, True)
    test.compare(waitForObjectExists(names.o_Button)["class"], "Button")
    test.compare(waitForObjectExists(names.o_Button).text, "I am a Driver")
    test.compare(waitForObjectExists(names.o_Button).type, "Button")
    clickButton(waitForObject(names.o_Button))
    test.compare(waitForObjectExists(names.o_Pane).type, "Pane")
    test.compare(waitForObjectExists(names.o_Pane).text, "")
    test.compare(waitForObjectExists(names.o_Pane).enabled, True)

    test.compare(waitForObjectExists(names.o_Pane).nativeObject.IsTextPatternAvailable, False)
    test.compare(waitForObjectExists(names.o_Pane).nativeObject.SizeTypeOfSet, 0)
    mouseClick(waitForObject(names.o_Pane), 204, 15, MouseButton.PrimaryButton)
    type(waitForObject(names.o_Pane_3), "<Backspace>")
    type(waitForObject(names.o_Pane_3), "akshara")
```

```

mouseClick(waitForObject(names.cab_Hiring_System_Window), 1239, 327,
MouseButton.PrimaryButton)

test.compare(waitForObjectExists(names.o_Pane).text, "akshara")
test.compare(waitForObjectExists(names.o_Pane).visible, True)
test.compare(waitForObjectExists(names.o_Pane_2).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_2).text, "")

test.compare(waitForObjectExists(names.o_Pane_2).nativeObject.IsTextPatternAvailable
, False)

test.compare(waitForObjectExists(names.o_Pane_2).nativeObject.SizeOfSet, 0)
test.compare(waitForObjectExists(names.o_Pane_2).visible, True)
mouseClick(waitForObject(names.o_Pane_2), 149, 14, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "s")
test.compare(waitForObjectExists(names.o_Pane_2).text, "s")
test.compare(waitForObjectExists(names.o_Pane_8).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_8).text, "")
test.compare(waitForObjectExists(names.o_Pane_8).type, "Pane")
test.compare(waitForObjectExists(names.o_Pane_8).visible, True)
test.compare(waitForObjectExists(names.o_Pane_8).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_8), 251, 25, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "2")
test.compare(waitForObjectExists(names.o_Pane_8).text, "27")
test.compare(waitForObjectExists(names.o_Pane_9).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_9).text, "")

mouseClick(waitForObject(names.cab_Hiring_System_Window), 1339, 285,
MouseButton.PrimaryButton)

type(waitForObject(names.o_Pane_3), "ozhikode")

```

```

test.compare(waitForObjectExists(names.o_Pane_9).visible, True)
test.compare(waitForObjectExists(names.o_Pane_9).text, "kozhikode")
test.compare(waitForObjectExists(names.o_Pane_10).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_10).text, "")
test.compare(waitForObjectExists(names.o_Pane_10).visible, True)
test.compare(waitForObjectExists(names.o_Pane_10).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_10), 81, 10, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "kozhikode kerala")
test.compare(waitForObjectExists(names.o_Pane_10).text, "kozhikode kerala")
test.compare(waitForObjectExists(names.o_Pane_4).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_4).text, "")
test.compare(waitForObjectExists(names.o_Pane_4).nativeObject.IsContentElement,
True)

type(waitForObject(names.o_Pane_3), "7306228009")
test.compare(waitForObjectExists(names.o_Pane_4).text, "7306228009")
test.compare(waitForObjectExists(names.cab_Hiring_System_Window).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_11).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_11).text, "")
test.compare(waitForObjectExists(names.o_Pane_11).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_11), 179, 12, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "ah@")
test.compare(waitForObjectExists(names.o_Pane_11).text, "aksh@gmail.com")
test.compare(waitForObjectExists(names.o_Pane_5).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_5).text, "")
test.compare(waitForObjectExists(names.o_Pane_5).nativeObject.IsContentElement,
True)

```

```
test.compare(waitForObjectExists(names.o_Pane_5).text, "aksh1")
test.compare(waitForObjectExists(names.o_Pane_7).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_7).text, "")
test.compare(waitForObjectExists(names.o_Pane_7).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_7), 71, 6, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "1")
test.compare(waitForObjectExists(names.o_Pane_7).text, "12345")
clickButton(waitForObject(names.cab_Hiring_System_Button_3))
mouseClick(waitForObject(names.o_Pane), 87, 6, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "<Backspace>")
```

```
type(waitForObject(names.o_Pane_3), "<Backspace>")
clickButton(waitForObject(names.cab_Hiring_System_Button_3))
type(waitForObject(names.o_Pane_3, 33000), "ksharasnair")
type(waitForObject(names.o_Pane_3), "<Left>")
type(waitForObject(names.o_Pane_3), "<Left>")
type(waitForObject(names.o_Pane_3), "<Left>")
type(waitForObject(names.o_Pane_3), "<Right>")
type(waitForObject(names.o_Pane_3), "<Right>")
type(waitForObject(names.o_Pane_3), "<Right>")
type(waitForObject(names.o_Pane_3), "<Right>")
type(waitForObject(names.o_Pane_3), "<Right>")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4)["class"], "Button")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).enabled, True)
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).type, "Button")
test.compare(waitForObjectExists(names.o_Button)["class"], "Button")
test.compare(waitForObjectExists(names.o_Button).enabled, True)
test.compare(waitForObjectExists(names.o_Button).type, "Button")
```

```

clickButton(waitForObject(names.cab_Hiring_System_Button_4))
clickButton(waitForObject(names.cab_Hiring_System_Button_2))
test.compare(waitForObjectExists(names.o_Button)["class"], "Button")
test.compare(waitForObjectExists(names.o_Button).enabled, True)
test.compare(waitForObjectExists(names.o_Button).type, "Button")
clickButton(waitForObject(names.cab_Hiring_System_Button_2))
type(waitForObject(names.o_Pane_3), "ksh1")
type(waitForObject(names.o_Pane_3), "<Backspace>")
test.compare(waitForObjectExists(names.o_Pane_5).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_5).text, "")
test.compare(waitForObjectExists(names.o_Pane_5).nativeObject.IsContentElement,
True)

type(waitForObject(names.o_Pane_3), "aksh1")
test.compare(waitForObjectExists(names.o_Pane_5).text, "aksh1")
test.compare(waitForObjectExists(names.o_Pane_7).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_7).text, "")
test.compare(waitForObjectExists(names.o_Pane_7).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_7), 81, 15, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "1345")
test.compare(waitForObjectExists(names.o_Pane_7).text, "12345")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2)["class"],
"Button")

test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).enabled,
True)

test.compare(waitForObjectExists(names.cab_Hiring_System_Button_2).type,
"Button")

dragAndDrop(waitForObject(names.o_Pane_5), 64, 7, names.o_Image, 13, 14)

```

```
type(waitForObject(names.o_Pane_3), "<Backspace>")
clickButton(waitForObject(names.cab_Hiring_System_Button_2))
type(waitForObject(names.error_OK_Button), "a")
type(waitForObject(names.o_Pane_3, 36000), "<Backspace>")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4)["class"],
```

```

"Button")

test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).enabled,
True)

test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).type,
"Button")

clickButton(waitForObject(names.cab_Hiring_System_Button_4))

clickButton(waitForObject(names.cab_Hiring_System_Button_4))

clickButton(waitForObject(names.cab_Hiring_System_Button_4))

test.compare(waitForObjectExists(names.o_Button)["class"], "Button")

test.compare(waitForObjectExists(names.o_Button).enabled, True)

test.compare(waitForObjectExists(names.o_Button).type, "Button")

clickButton(waitForObject(names.cab_Hiring_System_Button_4))

test.compare(waitForObjectExists(names.o_Pane).enabled, True)

test.compare(waitForObjectExists(names.o_Pane).text, "")

test.compare(waitForObjectExists(names.o_Pane).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane), 123, 16, MouseButton.PrimaryButton)

type(waitForObject(names.o_Pane_3), "gokul")

test.compare(waitForObjectExists(names.o_Pane).text, "gokul")

test.compare(waitForObjectExists(names.o_Pane_2).enabled, True)

test.compare(waitForObjectExists(names.o_Pane_2).text, "")

test.compare(waitForObjectExists(names.o_Pane_2).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_2), 119, 15, MouseButton.PrimaryButton)

type(waitForObject(names.o_Pane_3), "d")

test.compare(waitForObjectExists(names.o_Pane_2).text, "d")

test.compare(waitForObjectExists(names.o_Pane_8).enabled, True)

test.compare(waitForObjectExists(names.o_Pane_8).text, "")

```

```
test.compare(waitForObjectExists(names.o_Pane_8).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_8), 48, 19, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "21")

test.compare(waitForObjectExists(names.o_Pane_8).text, "21")
test.compare(waitForObjectExists(names.o_Pane_9).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_9).text, "")

test.compare(waitForObjectExists(names.o_Pane_9).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_9), 141, 14, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "t")

test.compare(waitForObjectExists(names.o_Pane_9).text, "trissur")
test.compare(waitForObjectExists(names.o_Pane_10).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_10).text, "")

test.compare(waitForObjectExists(names.o_Pane_10).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_10), 115, 14, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "trissur kerala")

test.compare(waitForObjectExists(names.o_Pane_10).text, "trissur kerala")
test.compare(waitForObjectExists(names.o_Pane_4).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_4).text, "")

test.compare(waitForObjectExists(names.o_Pane_4).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_4), 193, 10, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "7309876543")

test.compare(waitForObjectExists(names.o_Pane_4).text, "7309876543")
test.compare(waitForObjectExists(names.o_Pane_11).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_11).text, "")
```

```
test.compare(waitForObjectExists(names.o_Pane_11).nativeObject.IsContentElement,
True)

type(waitForObject(names.o_Pane_3), "micro")

test.compare(waitForObjectExists(names.o_Pane_11).text, "micro")

test.compare(waitForObjectExists(names.o_Pane_5).enabled, True)

test.compare(waitForObjectExists(names.o_Pane_5).text, "")

test.compare(waitForObjectExists(names.o_Pane_5).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_5), 169, 17, MouseButton.PrimaryButton)

type(waitForObject(names.o_Pane_3), "goku")

test.compare(waitForObjectExists(names.o_Pane_5).text, "gokull")

test.compare(waitForObjectExists(names.o_Pane_7).enabled, True)

test.compare(waitForObjectExists(names.o_Pane_7).text, "")

test.compare(waitForObjectExists(names.o_Pane_7).nativeObject.IsContentElement,
True)

mouseClick(waitForObject(names.o_Pane_7), 56, 17, MouseButton.PrimaryButton)

type(waitForObject(names.o_Pane_3), "111222")

test.vp("VP1")

mouseClick(waitForObject(names.o_Pane), 259, 18, MouseButton.PrimaryButton)

type(waitForObject(names.o_Pane_3), "<Backspace>")

type(waitForObject(names.o_Pane_3), "<Backspace>")
```

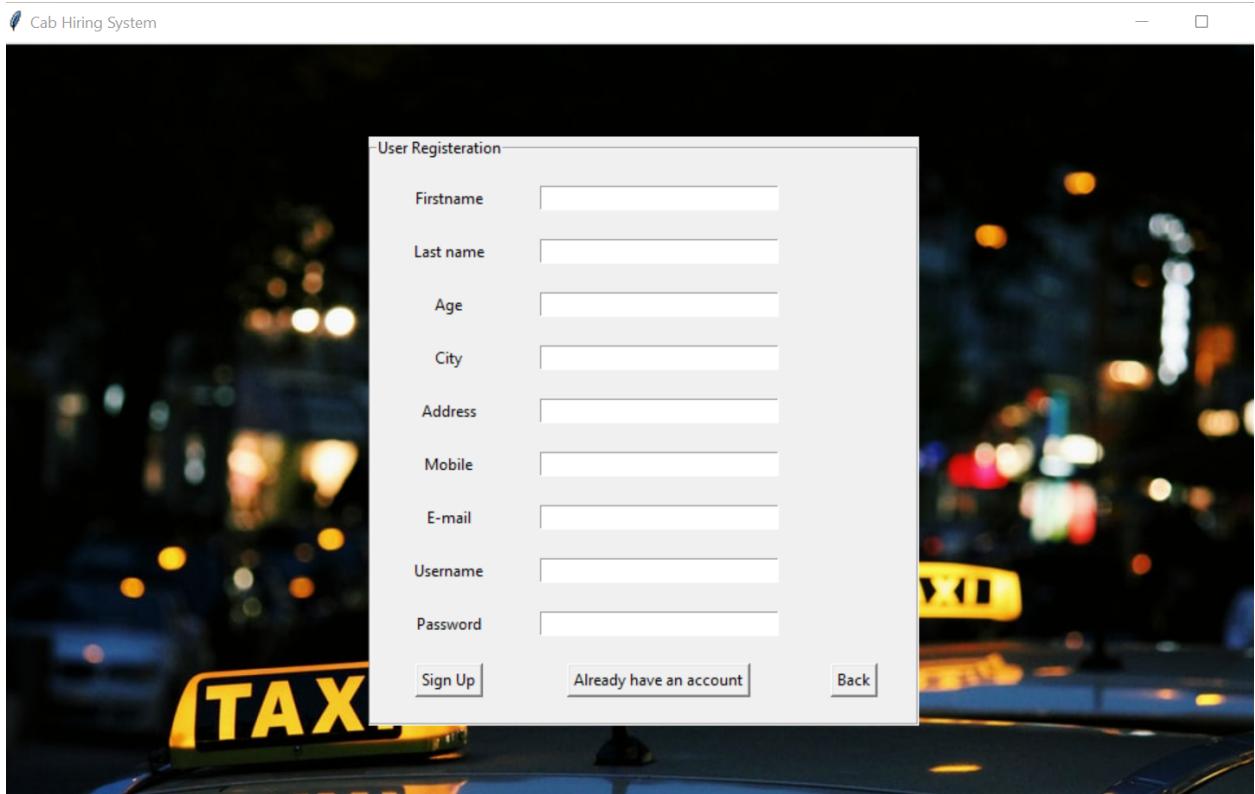
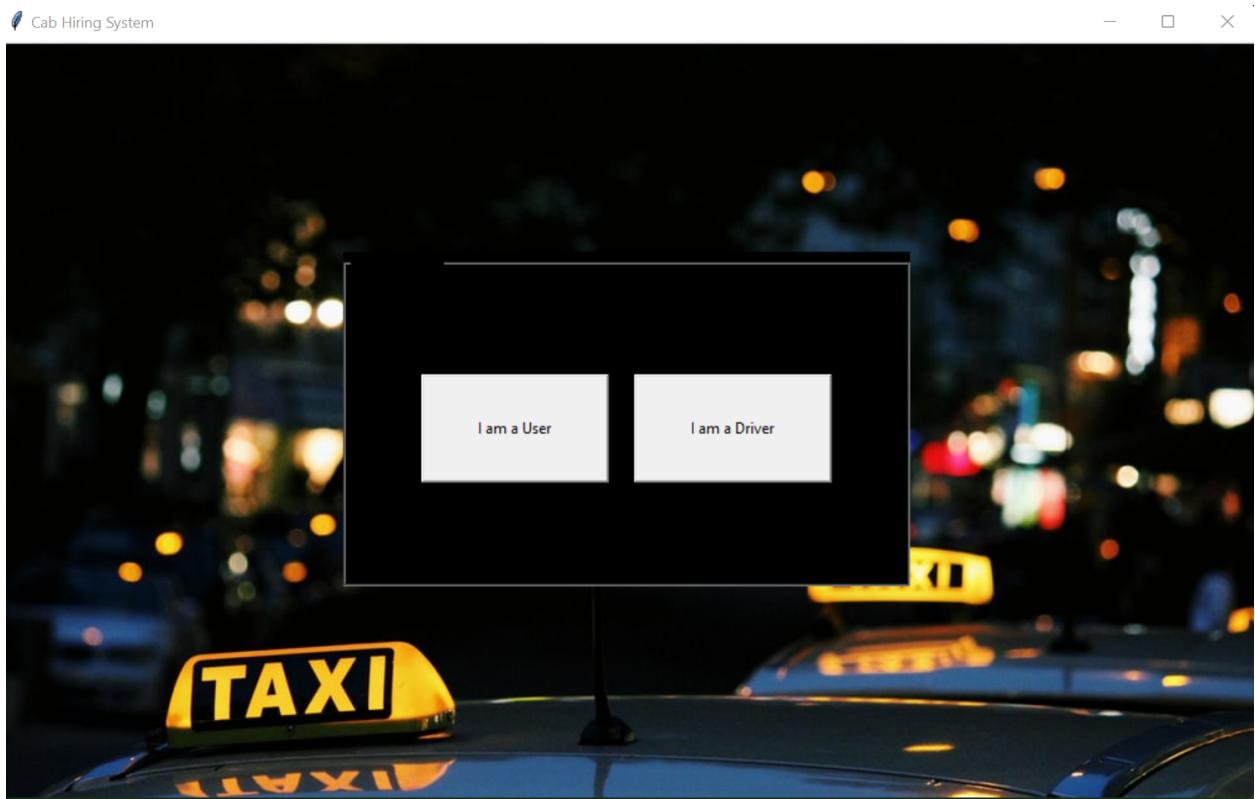
```
type(waitForObject(names.o_Pane_3), "<Backspace>")
clickButton(waitForObject(names.cab_Hiring_System_Button_3))
type(waitForObject(names.o_Pane_3), "g")
test.compare(waitForObjectExists(names.o_Button)["class"], "Button")
test.compare(waitForObjectExists(names.o_Button).enabled, True)
test.compare(waitForObjectExists(names.o_Button).type, "Button")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4)["class"],
"Button")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).enabled,
True)
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).type,
"Button")
```

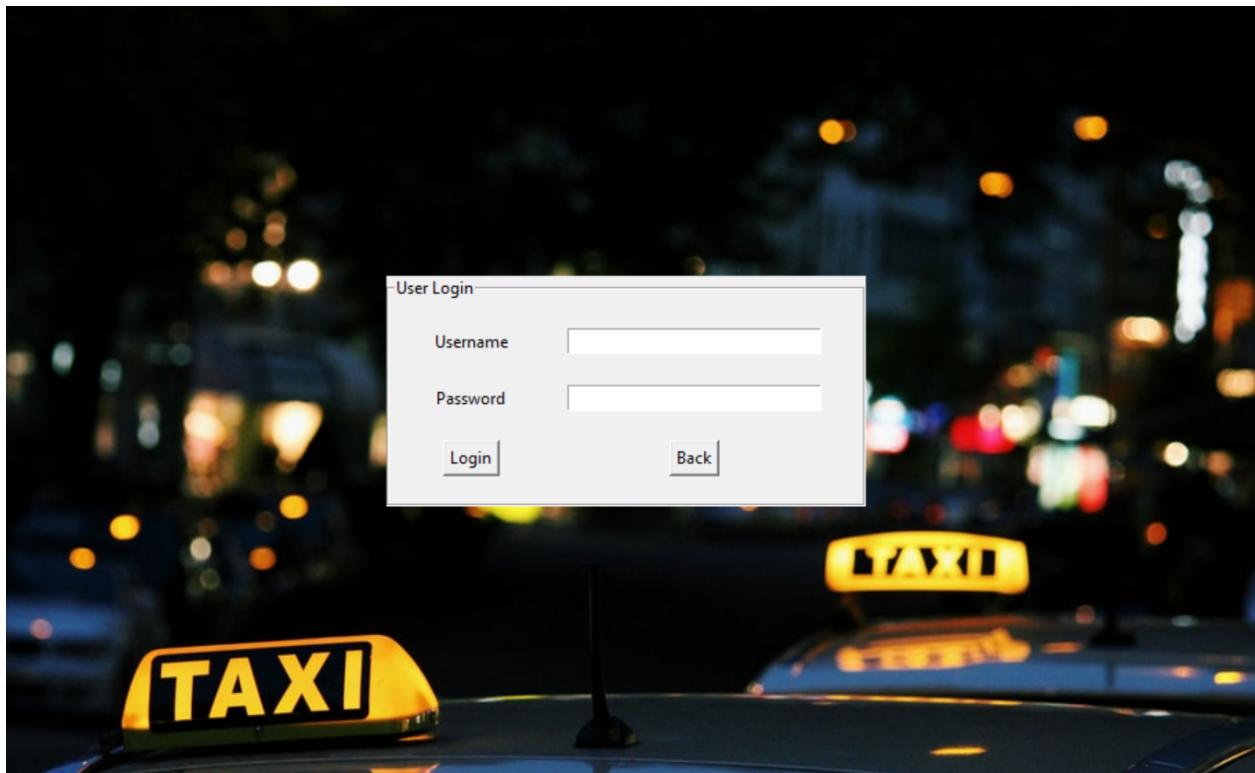
```

test.compare(waitForObjectExists(names.o_Button)["class"], "Button")
test.compare(waitForObjectExists(names.o_Button).enabled, True)
test.compare(waitForObjectExists(names.o_Button).type, "Button")
clickButton(waitForObject(names.cab_Hiring_System_Button_4))
clickButton(waitForObject(names.o_Button))
clickButton(waitForObject(names.cab_Hiring_System_Button_2))
test.compare(waitForObjectExists(names.o_Pane_5).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_5).text, "")
test.compare(waitForObjectExists(names.o_Pane_5).nativeObject.IsControlElement,
True)
test.compare(waitForObjectExists(names.o_Pane_5).text, "gokull")
test.compare(waitForObjectExists(names.o_Pane_7).enabled, True)
test.compare(waitForObjectExists(names.o_Pane_7).text, "")
test.compare(waitForObjectExists(names.o_Pane_7).nativeObject.IsContentElement,
True)
mouseClick(waitForObject(names.o_Pane_7), 140, 17, MouseButton.PrimaryButton)
type(waitForObject(names.o_Pane_3), "1122")
test.compare(waitForObjectExists(names.o_Pane_7).text, "111222")
test.compare(waitForObjectExists(names.o_Button)["class"], "Button")
test.compare(waitForObjectExists(names.o_Button).enabled, True)
test.compare(waitForObjectExists(names.o_Button).type, "Button")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4)["class"],
"Button")
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).enabled,
True)
test.compare(waitForObjectExists(names.cab_Hiring_System_Button_4).type,
"Button")
clickButton(waitForObject(names.cab_Hiring_System_Button_4))

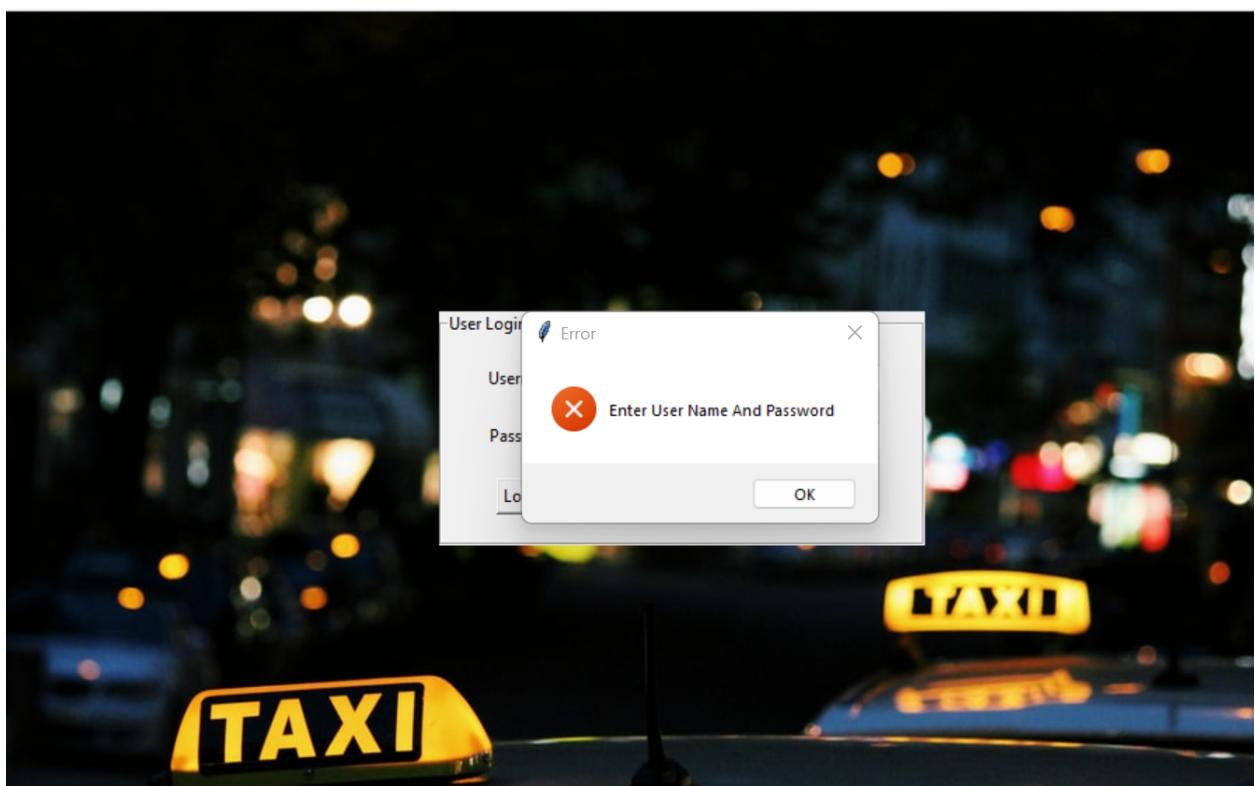
```

SCREENSHOTS OF THE PROJECT :





• Car hiring system





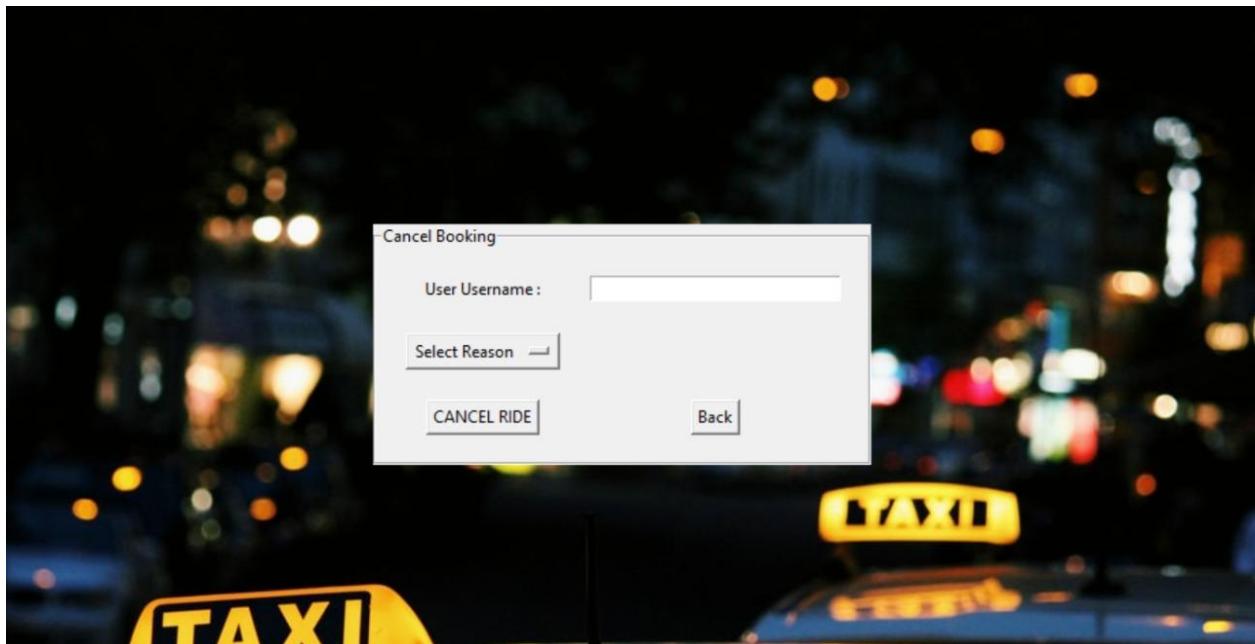
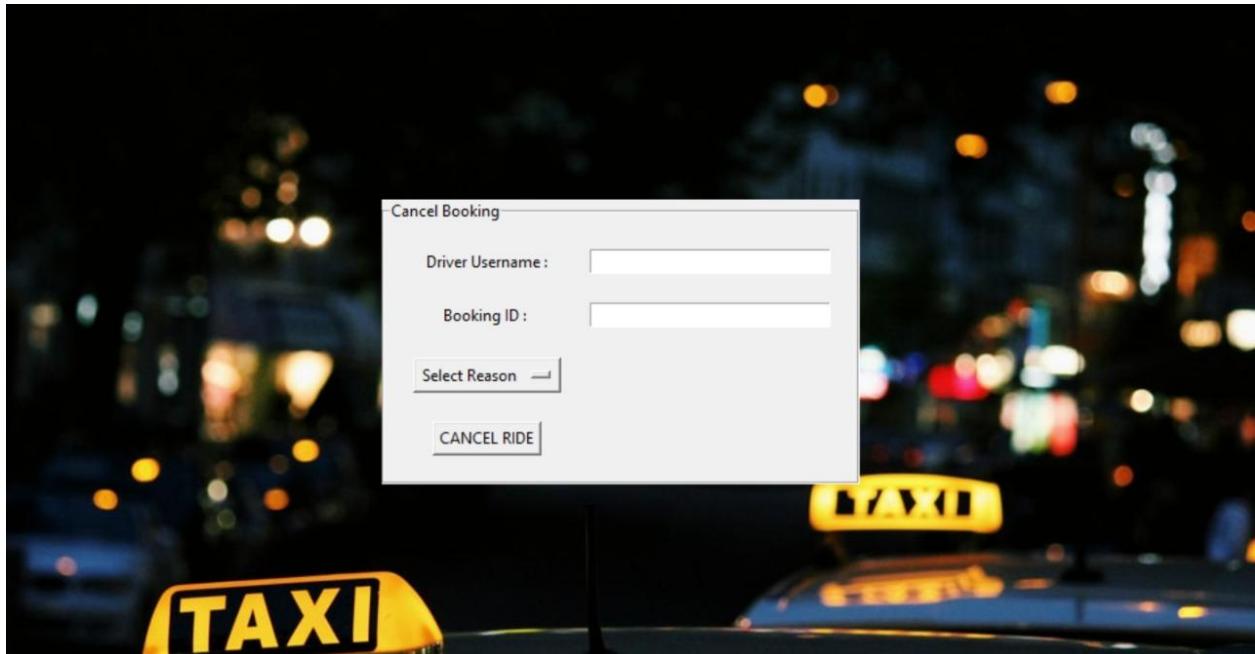
Cab Booking

Customer Info		Route Info		Receipt	
First Name	Optional	Pickup	Palakkad	#### BILL ####	Palakkad
Last Name	Optional	Drop	Kottayam	Pickup Location :	Kottayam
Address	Optional	Max Passengers :	3	Destination Location :	Micro
Mobile	Optional	Base Charge :	30	Cab Type :	2895
Email	Optional	Distance (Kms) :	191	Cab Charge :	Total : 2895
Emergency Num	Optional	<input type="checkbox"/> Travel Insurance	Rs.0		
		Get Bill			

Car Selection

Micro (Charge per Km) : Rs.15
 Sedan (Charge per Km) : NA
 SUV (Charge per Km) : NA

[Book](#) [Home](#) [Log Out](#)



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	driver_first_name	driver_last_name	driver_age	driver_city	driver_adde	driver_car_type	driver_username	driver_password
▶	Akshara	S	21	Kozhikode	Kozhikode	Micro	aksh	123
	Akshara	S	21	Kozhikode	Kozhikode	Micro	aksh123	123
	Gokul	D	22	Thrissur	Thrissur	SUV	goks	110
	Radhika	R	21	US	US	Sedan	radh	111
	Ram	S	40	Palakkad	Palakkad	Sedan	ram	101
	shyam	s	50	Palakkad	Palakkad	Sedan	shyam	102
*	HULL	HULL	NULL	HULL	HULL	HULL	HULL	HULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	driver_user_name	driver_car_type	driver_status_now	count
▶	aksh	Micro	Offline	0
	goks	SUV	Hired	0
	radh	Sedan	Hired	0
	ram	Sedan	Hired	3
	shyam	Sedan	Available	1
*	HULL	HULL	HULL	HULL

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	username	driver_user_name	cartype	pickuploc	droploc	status
	Aadith		SUV	Palakkad	Thrissur	Cancelled
	Aadith	aksh	Micro	Thiruvananthapuram	Palakkad	Cancelled
	Aadith	aksh	Micro	Kannur	Kollam	Cancelled
	Aadith		Micro	Kollam	Kozhikode	Cancelled
	Aadith		Micro	Thrissur	Palakkad	Cancelled
	Aadith		Micro	Kannur	Palakkad	Cancelled
	Aadith		Micro	Kollam	Palakkad	Cancelled
	Aadith	goks	SUV	Kollam	Palakkad	Hired
	Aadith	ram	Sedan	Palakkad	Thrissur	Cancelled
	Aadith	ram	Sedan	Palakkad	Kollam	Cancelled
	Aadith	ram	Sedan	Kollam	Palakkad	Cancelled
	Aadith	shyam	Sedan	Kollam	Palakkad	Done
	Aadith	ram	Sedan	Pathanamthitta	Kannur	Cancelled

	first_name	last_name	age	city	adde	user_name	password
▶	A	a	1	a	aa	a	1313
	Aadith	S	21	Palakkad	Suraksha, Puthur, Palakkad	Aadith	1313
*	Gokuldas	D	22	Thrissur	Thrissur	gokud11	gokud11
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

CONCLUSION

The aim of this project was to develop a downloadable software that helps a Cab company manage and run their system online. The administration unit, the drivers and the customers all have an organized system to make use of. This way, every party will get their own necessary resources and/or information. The administration unit of this cab hiring system will be able to work efficiently and productively with the help of this software. It is easy to use and has all needed features to successfully run the system.