

**National University of Computer and Emerging
Sciences - FAST Computer Science Department**



**Fundamentals of Software Engineering
CS-2004**

Submitted to:

Ma'am Maheen Arshad

Submitted by:

1. Musaab Imran (20i-1794)
2. Muhammad Ismail Ramzan (20i-1941)
3. Muhammad Usman Shahid (20i-1797)

Contents

Fast Bus Management System.....	2
User Stories:	3
02	3
03	4
04	6
05	8
Project Backlog	9
Product Backlog:	10
Sprint Backlog	11
 Sprint 02:	12
Tasks:	13
Trello:	13
Team Duties:	18

Fast Bus Management System

The module we will be covering in sprint 1 contains three user stories. All three user stories are interlinked. The user stories are related to login, registration, and log-out. We will be fully implementing all three of the user stories in sprint 1 which would include both the front and back end.

User Stories:

02

<p>ID: US-02 Title: Accounts (finance)</p> <ul style="list-style-type: none"> ➤ Created by: Group ➤ Date created: 01/03/2022 ➤ Last updated by: 20/05/2022 <p>As a customer, I want to have a proper accounts section that should show students his/her fee status. If the fee has been paid then he/she is allowed to book the seat in the vehicle. There should be a message on the 30th of the month telling all the drivers to check their accounts for salary and dropbox if someone has an issue with the finances.</p> <p><u>Descriptions:</u></p> <ul style="list-style-type: none"> o Students will check the accounts section to check the status of their fees. o Drivers should see the accounts section to check the status of their salary. o Dropbox to entertain any queries. <p><u>Acceptance criteria:</u></p> <ul style="list-style-type: none"> o If the fee is not paid by the student, he/she shouldn't be allowed to book the seat in the vehicle unless he/she pays it. 	<p><u>Priority:</u></p> <p>High</p> <p><u>Estimate hours:</u></p> <p>4-5</p>
---	---

ID: US-03 Title: Booking Details

- **Created by:** Group
- **Date created:** 01/03/2022
- **Last updated by:** 20/05/2022

As a customer, I want to have a proper seat booking system where the student from 12 pm-midnight could select a particular free seat in the vehicle and after that, a boarding number would be generated which would be required for the identification of the student by the bus driver.

Sub-User Stories:**SUS:01 Seat Management****Description:**

Showing the list of seats available and unavailable. Full bus mapped seats.

Acceptance criteria:

Once the seat is selected no other seat would be selected.

Priority:

Medium

Estimate hours:

1- 1.5

Priority:

High

Estimate hours:

Description:

- o Students could select any free seat to sit in the bus.
- o The seats of the vehicles should be shown.
- o A boarding number should be generated and shown to the student.

4-5

Acceptance criteria:

- o Without the boarding number, students mustn't be allowed to sit in the vehicle.
- o A boarding list should be generated against the name of different students.
- o They booked seats that shouldn't be shown or allowed to be rebooked. They should be locked.

ID: US-04 Title: Travel Details**Priority:**

High

- **Created by:** Group
- **Date created:** 01/03/2022
- **Last updated by:** 20/05/2022

As a customer, I want to show the students all of the travel details of the trip. They should be able to see the bus number, driver details (number and name), and route with the time mentioned.

Sub-User Stories:**SUS:01 Maps****Priority:****Description:**

Showing the current location of the user and the location of the university.

Medium

Acceptance criteria:

The location gets updated when the user reopens the app.

Estimate**hours:**

1- 1.5

Estimate hours:**Description:**

Things to be displayed:

- Bus's number.
- Driver's number.
- Driver's name.
- Route with time.

<u>Acceptance criteria:</u>	4-5
<ul style="list-style-type: none">○ The travel details are extensively told.○ After 12 pm-midnight the status and travel details should be updated for the students to check it early in the morning.○ The route should be clear on the map.○ The map should be labeled with the arrival time of different stopping points.	

ID: US-05 **Title: Student Info Log**

Priority:

High

- **Created by:** Group
- **Date created:** 01/03/2022
- **Last updated by:** 20/05/2022

As a customer, I want to that when a student boards the bus the driver should have a list of names of students against their boarding number generated in the previous module. This way he can cross-check whether the student is of FAST university or an outsider. The picking points of the same route students should be shown to the driver.

Estimate hours:

Description:

Things to be displayed to the driver:

- o Boarding number against the names of the students.
- o Picking points of the same route students.

4-5

Acceptance criteria:

- The boarding number should be the same in the list of drivers and shown to students. (important)
- The picking points must be near the predefined routes.

Project Backlog

USER STORIES	DESCRIPTION
US01:	Login system for students, bus drivers, and admin.
US02:	Chalan status for students and salary confirmation for the bus drivers.
US03:	Seat booking and boarding number generation for students.
US04:	Travel information, details like bus and driver numbers, and time mentioned routes on maps.
US05:	Passenger information log, showing names, tokens, and picking points to the driver.
US06:	Students and drivers register who are traveling for the first time against different information.
US07:	Daily bus and route allotment by the admin.
US08:	Account deletion of students and drivers who have left.
US09:	And adding new students and drivers, also creating their accounts.
US10:	Maintaining bus data, also adding and removing the buses.
US11:	Logout of the account after the user enters the logout button.

Product Backlog:

PRODUCT BACKLOG



User story	Estimate	Priority
USER LOGIN	7-8	HIGH
USER REGISTRATION	7-8	HIGH
BOOKING DETAILS	6-7	HIGH
TRAVEL DETAILS	6-7	HIGH
STUDNET INFO LOG	6-7	MEDIUM
ACCOUNTS	7-8	MEDIUM
ACCOUNT MANAGEMENT 01+ 02	9-12	MEDIUM
BUS ALLOTMENT	5-6	MEDIUM
BUS MANAGEMENT	4-5	LOW
LOG OUT	4-5	LOW

Sprint Backlog

Sprint 1 (1.5 weeks):

US01- User Login

US06- User Registration

US11- Logout

Sprint 2 (1weeks):

US03- Booking Details

US04- Travel Details

Sprint 3 (1 week):

US08- Account Management 01

US09- Account Management 02

Sprint 4 (1 week):

US07- Bus Allotment

US10- Bus Management

Sprint 5 (1 weeks):

US05- Student Info Log

US02- Accounts(finance)

- The main goal is to complete the project in approximately 2.5 months.
- Each sprint contains interlinked modules for fast work.
- The average maximum time for each sprint is 7-8 days.

Sprint 02:

Requirement#	Description
2	Chalan status for students and salary confirmation for the bus drivers.
3	Seat booking and boarding number generation for students.
4	Travel information, details like bus and driver numbers, and time mentioned routes on maps.
5	Passenger information log, showing names, tokens, and picking points to the driver.

SPRINT 02

US:02
Accounts
(Finance)

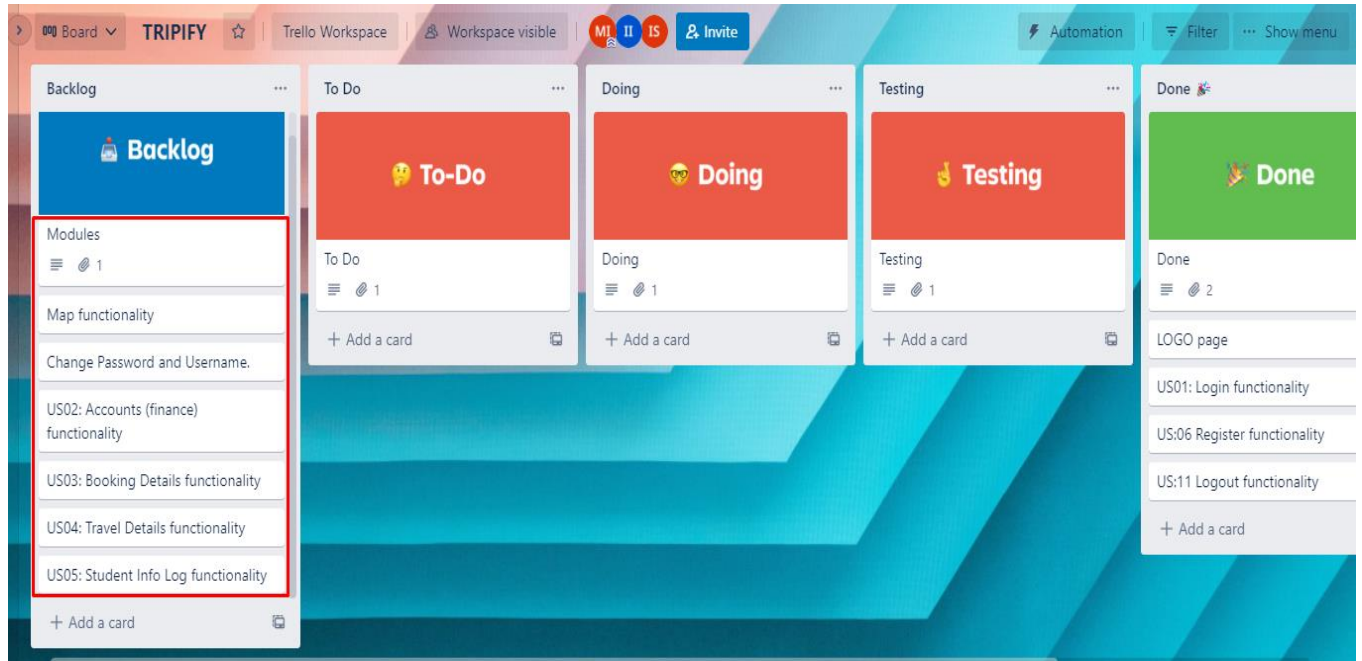
US: 03
Booking
Details

US:04
Travel
Details

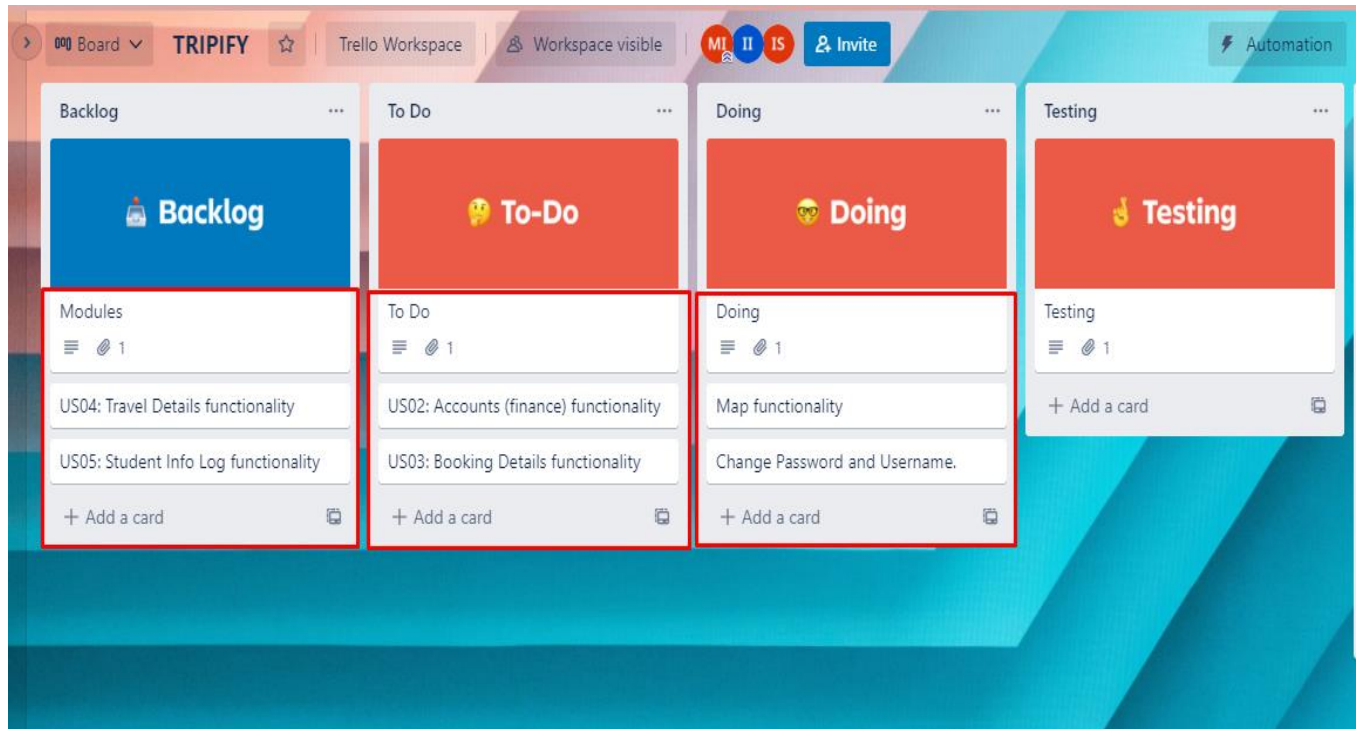
US: 05
Student
Info Log

Trello:

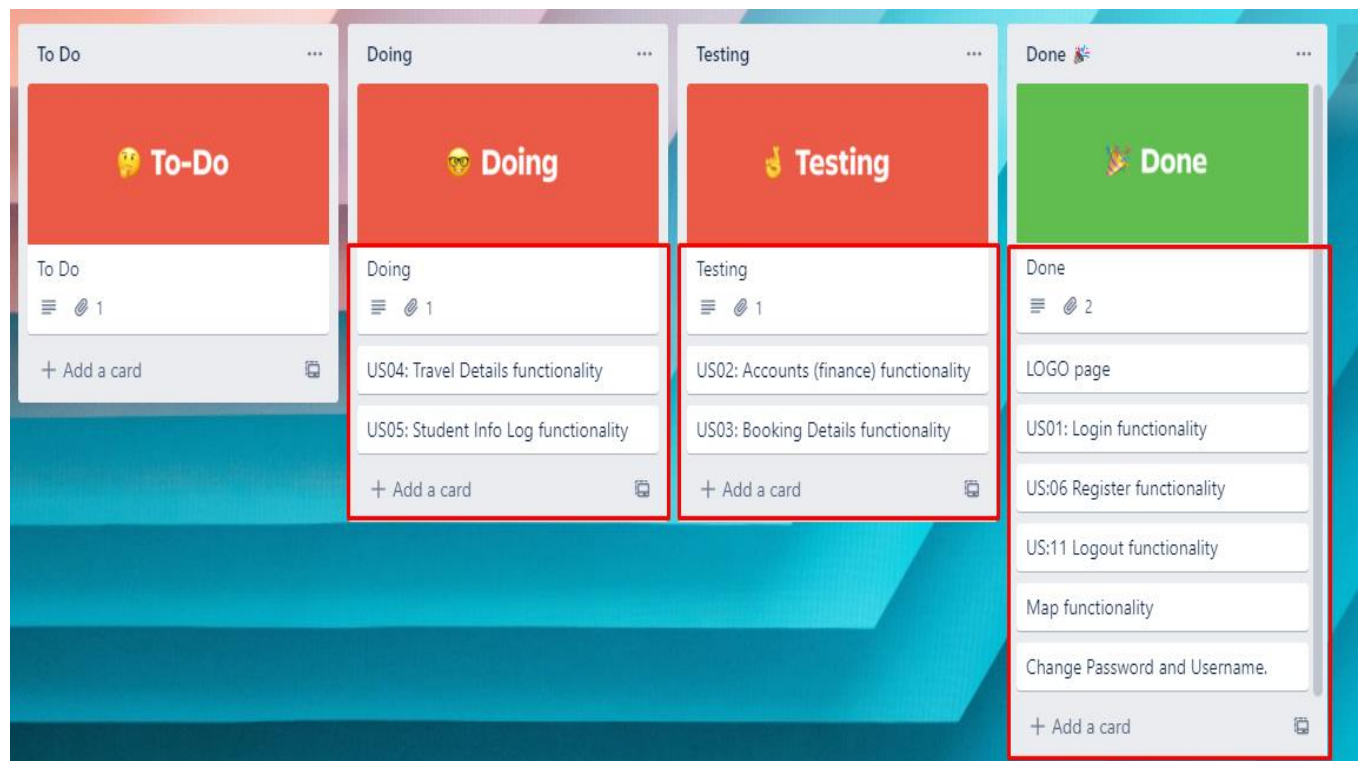
- At the start of the project:



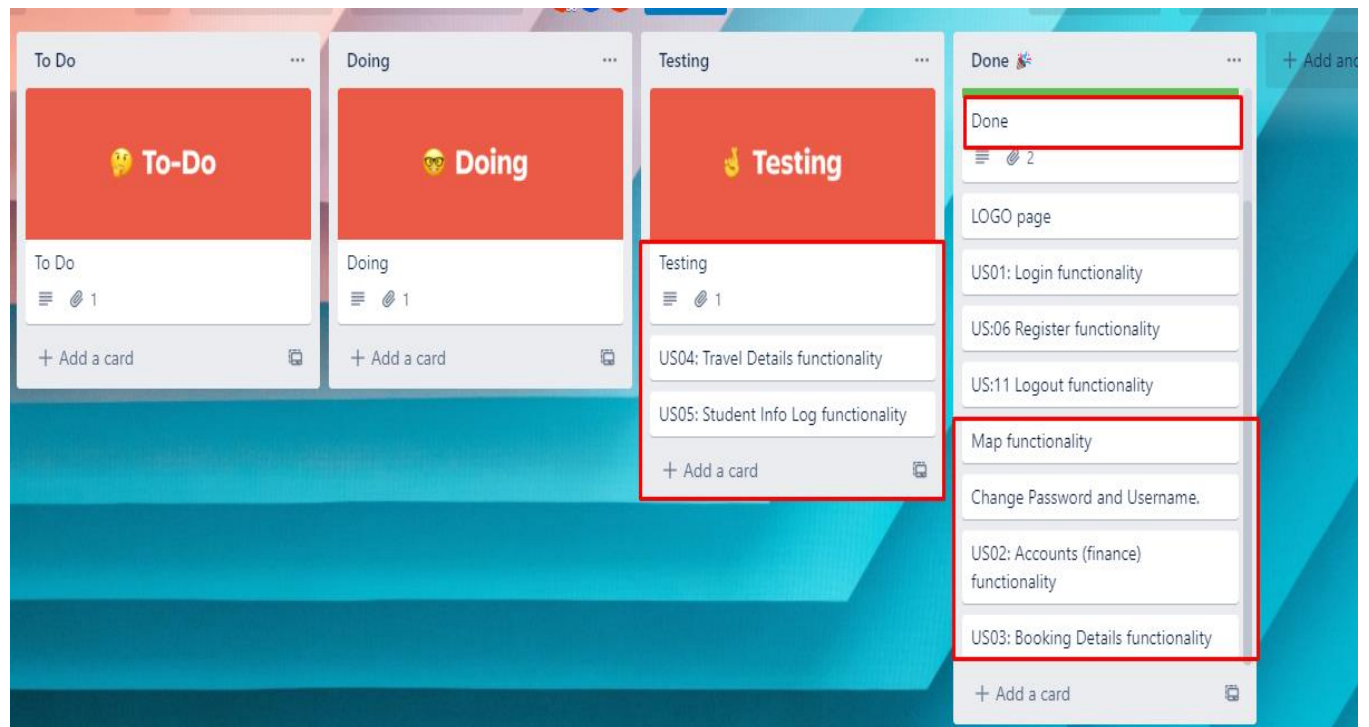
- At the middle of the project:



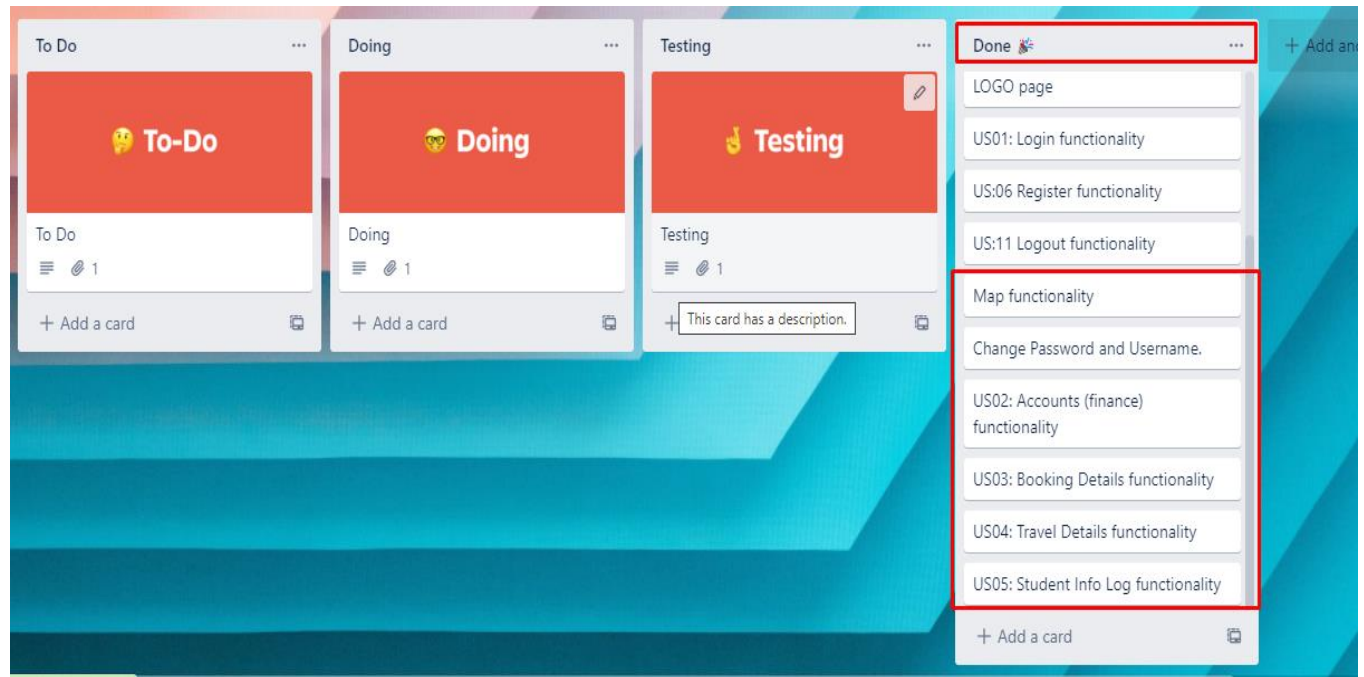
- When 60% of the project was completed:



- When 85% of the project was completed:



- At the end of the project:



Tasks:

<p>US02: Accounts (Finance)</p> <p>(Driver, Admin, Student)</p>	<p>Full Front-End Implementation:</p> <p>T201: For the front of accounts firstly we set up the status check of the fee of the students.</p> <p>T202: The fee status is seen by students after paying the fee manually or in the bank.</p> <p>T203: The bool variable was set to see the fee status</p> <p>T204: The bool variable was used to update the salary status</p> <p>T205: The admin can change the bool variables of salary and fee.</p> <p>T206: If the bool variable for fee status is true then the student can book a ride otherwise, he cannot.</p> <p>T207: The finance widget was set for dealing with this user story.</p> <p>Full Back-End Implementation:</p> <p>T208: Firstly, Saved the information of the driver, student, and admin in the database.</p> <p>T209: Fetching the Information for the particular user</p> <p>T210: Getting the information in a synchronized manner to display it on the front-end.</p>
---	--

	<p>T211: Management of backend security to not allow one user to fetch the information of another user.</p> <p>T212: Admin information in the different documents in a non-relational database.</p>
<p>US03: Booking Details (Student)</p>	<p>Full Front-End Implementation:</p> <p>T301: The booking details functionality was implemented but asked for information from students.</p> <p>T302: The student if he/she has paid the fee can book a ride.</p> <p>T303: The student is asked:</p> <ul style="list-style-type: none"> - Address - Phone number - Destination address - Coupon number <p>T304: After this information is asked the student is led to the seat selection page. A seat selection widget was made to display all the available and unavailable seats.</p> <p>T305: After booking the seats, the student is shown the next user story's travel details.</p> <p>Full Back-End Implementation:</p> <p>T306: Store the booking details of the student in the database</p> <p>T307: Fetch the booking detail of the student from the database</p> <p>T308: Fetch the information of only the student who is logged into the system</p> <p>T309: Implementation of the security so that the student cannot access the information of another student</p>
<p>US4: Travel Details (Student, Admin)</p>	<p>Full Front-End Implementation:</p> <p>T401: When the student has entered all of the booking information, he/she would be taken to the ticket/travel details page.</p> <p>T402: The travel details page would show the student</p> <ul style="list-style-type: none"> - Name - Phone number - Address - Boarding number - Bus number

	<ul style="list-style-type: none"> - Driver number - Driver name <p>T403: This information is important for the travel.</p> <p>T404: The same information would be sent to the admin and driver for no future trouble.</p> <p>Full Back-End Implementation:</p> <p>T405: Store the destination and the arrival of the user</p> <p>T406: Store the booking details of the student in the backend</p> <p>T407: Fetch the booking information of the student</p>
<p>US5:</p> <p>Student Info Log</p> <p>(Driver)</p>	<p>Full Front-End Implementation:</p> <p>T501: Display all of the information that was collected in the booking details sections.</p> <p>T502: The information would be of the address, number, boarding number, and seat number.</p> <p>T503: The info log of all the students contains the contact information for contacting the students</p> <p>T504: List of all the routes and buses that are allocated.</p> <p>Full Back-End Implementation:</p> <p>T504: Store the information of the driver</p> <p>T505: Fetch the information of the driver as required</p> <p>T506: The driver fetches the information of the students to see what students have booked a ride.</p>
<p>MAP</p> <p>Implementation</p>	<p>T101: Took the Google API and added it manifest xl file.</p> <p>T102: Adding dependencies of markers, geolocation, and polylines.</p> <p>T103: Getting the longitude and latitude of the current location and making markers at those points.</p> <p>T104: Hardcoding the longitude and latitude of the FAST university as it would be the default destination location.</p>

<p>Change user name and password</p>	<p>T101: Enter the new name T102: Enter the password T103: Verify the credentials T104: change the username</p> <p>T105: Enter the old password T106: Enter the new password T107: Again, Enter the new Password T108: Verify the credentials T109: click change password</p> <p>T110: Fetch the current username and update the username from the backend. T112: Update the information in the backend named field T113: Once the information is updated show it in the front. T114: Make sure that the user is authenticated T115: Once the user is authenticated its password is replaced with the new one</p>
---	--

Team Duties:

Mr. Musaab Imran: Scrum Master, developer

- Assign roles, and tasks, check and report process.
- Implementation of booking details(frontend).
- Implementation of FAQ, fee, and salary status(frontend).
- Documentation.

Mr. Usman Shahid: Administrator, designer, developer

- Implementation of travel details (frontend).
- Implementation of Student info log(frontend).
- Implementation of change password and username functionality(frontend).
- Integration of project.

Mr. Ismail Ramzan: User experience engineer, developer, tester

- Connecting project to firebase(backend).
- Implementation of change password and username functionality (backend).
- Implementation of Student info log(backend).
- Unit and integration testing.