SPRINT - 1(ADMIN MODULE)

26-08-2022(FRIDAY)

Team Member	User Story	TO DO	In Progress	Impediment	To Verify	Done
1. [Jitin Sharma]	[1]. Bus reservation project need to link up with the frontend and backend simultaneously. For that Api Handlers and Angular Services and Components are required. The services will help to integrate with WebApis and component will integrate the services and UI. [1.1] For that Database schema and admin Api Handlers are to be created.	[1.1] Create a DB Schema for Admin with hardcoded values. [1.2] An Angular App for frontend [1.3]Services for Admin to validate the admin.(Post) [1.4] Component for Admin (Logic And UI)	[1.1] Created DB for Admin. [1.2] Created Angular app [1.3]Post Validate Service for admin Created		[1.1]Verify logic and service with UI	[1.1] Create a DB Schema for Admin with hardcoded values. [1.2] An Angular App for frontend [1.3] Services for Admin to validate the admin.(Post) [1.4] Component for Admin (Logic And UI) [1.5]Verified the SignIn
[Anshul Bansal]	2.Admin Module funtionality for editing the details of bus reservation system, to enable administrators for conducting tasks like monitoring, maintaining and controlling the business according to business requirement.	[2.1] Controller for Admin, [2.2] Models Created for Admin [2.3] Api for getList of Admins	[2.1] Created Controller for Admin, [2.2] Created Models for Admin [2.3] Created Api for getList of Admins (Verify With Postman)		[2.1] Verification with postman is to be done with postman with intergration from frontend	[2.1] Created Controller for Admin, [2.2] Models Created for Admin [2.3] Created Api for getList of Admins
[Omkar Bhagwat]	3.The functionalities which admin can do after logging in like - See the list of Buses, Edit any Bus details, Add a Bus in list, Delete particular Bus from records etc	[3.1] Controller for Bus [3.2] Models Created for Bus [3.3] Api for getList of Buses	[3.1] Created Controller for Bus [3.2] Created Models for Bus [3.3] Working on Api for getList of Bus (Verify With Postman)		[3.1] Verification with postman is to be done with postman with intergration from frontend	[3.1] Created Controller for Bus [3.2] Models Created for Bus [3.3] Created Api for getList of Bus
[Divya Marikanti]	4.Creating the Homepage for the project so that user and admin can perfom there task respectively,then after admin loginig there should be a side bar to see his tasks and also create db for bus	[4.1]DB Bus [4.2] Homepage Navbar [4.3] AdminSidebar	[4.1]Created a homepage [4.2]Created DB for Bus [4.3] Creating component for sidebar of the admin		[4.1]Verification with the team about UI,db created	[4.1]Created DB Bus [4.2]Created Homepage Navbar [4.3] Created AdminSidebar [4.4]Verified whole thing with the team

27-08-2022(Saturday)

Team Member	User Story	TO DO	In Progress	Impediment	To Verify	Done
[Jitin Sharma]	Logging In in Admin, Perform Operation on Bus Components to give admin a control.	[1.1]Functionality to show the FirstName of Admin in NavAdmin [1.2]Components for Bus [1.2.1]List Component [1.2.2]Find Component [1.2.3]Add Component [1.2.4]Edit Component [1.2.5] Remove Component	[1.2.1]List Component Completed [1.2.2]Find Component Completed [1.2.3]Add Component Completed	While Creating the Interface, CamelCasing is required at the client side. To Link up with Components the attributes should go in camel casing. This created some obstruction while displaying the data in port.	Design for all the components is to be verified and checked by the team. Input Forms are made using css and Bootstrap in parallel.	[1.2]Components for Bus [1.2.1]List Component Done (GET) [1.2.2] Find Component Done(GET) [1.2.3]Add Component Done (POST) [1.2.4]Edit Component Done (PUT) [1.2.5] Remove Component Done(DELETE)
[Anshul Bansal]	To give the admin control there is need to verify the authenticity of the credentials from the database.	[2.1] Api to send the Admin Details [2.2] Api for posting the credentials to validate it from the database during SignIn	[2.1] Created Api for getting details of specific admin(Verify With Postman) [2.2] Working on Api for Validating the SignIn Request		[2.1] Validate Api is to be verified with Postman and Integration with frontend left for the Apis	[2.1] Api for getting details of specific admin, [2.2] Api for Validating the SignIn Request
[Omkar Bhagwat]	To give admin, the control of System after log in, A CRUD ops is to be implemented on Bus details. For that APIs are required to be created.	[3.4] Api for getiing details of specific bus [3.5] Api for Adding a bus record [3.6] Api for Editing any bus record [3.7] Api for Deleting a specific bus	[3.4] Created Api for getting details of specific bus [3.5] Created Api for Adding a bus record [3.6] Working on Api for Editing any bus record [3.7] Working Api for Deleting a specific bus		[3.1] Each Api need to be verified with Postman and integration with frontend left for the Apis	[3.4] Created Api for getting details of specific bus [3.5] Created Api for Adding a bus record [3.6] Created Api for Editing any bus record [3.7] Created Api for Deleting a specific bus
[Divya Marikanti]	After Admin Sign in he can see the side bar which contains the operations he can perform on the bus components. For that Api Handlers are required.	[4.1]Create Service for Bus which Contain: [4.1.1]Get List of Buses(GET) [4.1.2] Find the details of Bus(GET) [4.1.3]Edit the bus Details(PUT) [4.1.4] Remove the bus (Delete) [4.1.5] Add the Bus(POST) [4.2] Designing the UI for the Bus Components	[4.1]Created the Bus service [4.2]Designing the UI for the components		Validation of all the services and UI of the component	[4.1]Created Service for Bus which Contain: [4.1.1]Get List of Buses(GET) [4.1.2]Find the details of Bus (GET) [4.1.3] Edit the bus Details(PUT) [4.1.4] Remove the bus (Delete) [4.1.5] Add the bus(POST) [4.2] Designed the UI for the Bus Components

SPRINT-2 (Customer Module)

29-08-2022(Monday)

	User Story	TO DO	In Progress	Impediment	To Verify	Done
1. [Jitin Sharma]	As the Login is created for customer, once searching for buses, the customer should select the seat from available number of seats that we are getting from the details of the seat.	[1.1]Create Api for Getting the Available Seats by busId [1.2]Create Api for getting the details of seats by busId [1.3] Create a service for getting the seat details by by bus Id [1.4]Work on the Seat Mapping Component	[1.1]Created Api for Getting the Available Seats by busId [1.2]Created Api for getting the details of seats by busId [1.3]Working on Service for seat mapping		[1.1]Verification of APIs on postaman is left [1.2] Intergration of Service and component is to be done	[1.1]Created Api for Getting the Available Seats by busId [1.2]Created Api for getting the details of seats by busId [1.3] Created a service for getting the seat details by by bus Id [1.4]Worked on the Seat Mapping Component
[Anshul Bansal]	Some schema is needed to store the state of availability for each seat of each bus. It is needed to develop the seat map for the bus reservation. Also to create a new transaction after a successful payment, we need to pass data across the component so that we can make an API call from the summary component.	[2.1] Create table for storing the Seat Map of each bus. [2.2] API for getting seats list for transaction id. [2.3] API to delete transaction. [2.4] Create services for handling transactions. [2.5] Create services for fetching available seats information for a specific bus. [2.6] For passing the data across the components a class with static members is needed to act like a central store.	[2.1] Created a SeatMap table in database. [2.2] Created API for GetSeatList for getting seats list for transaction id. [2.3] Created API to delete transaction. [2.4] Created services for handling transactions	Passing Data across the Components was one of our major challenges. We tried sending the data in the parameters of the router link, but it was not able to transfer the complex objects in the URL. So, after searching on this we found two approaches to deal with it. [1] Storing and Accessing data in the data variables of the ts file of the Parent Component [2] Creating a class with static members, so that it can act like a central store, and it can be accessible by each component. We went with the creation of the class, as it was easy to implement and more relevant to our scenario.	[2.1] Verification of the APIs with Postman then with the Client services is pending. [2.2] Verification of the services with the API and the ts file of the components is pending.	[2.1] Completed a SeatMap table in database. [2.2] Completed API for GetSeatList for getting seats list for transaction id. [2.3] Completed API to delete transaction. [2.4] Completed services for handling transactions. [2.5] Completed services for fetching available seats information for a specific bus. [2.6] Created a class which can act like a central store for each component for passing the data across the components. [2.7] Also tested integration of APIs and services.
[Omkar Bhagwat]	On the home page, User should be able to Register on the system. While Registering, user has to enter his/her details to get registered.	[3.1] Api GetCustList [3.2] Api for GetiCustList by CId [3.3] Api for post Customer to add details of customer [3.4] Create service for Post Registration for Customer [3.5] Create Component for Customer Registration	[3.1] Completed Api for GetCustList [3.2] Completed Api for GetCustList by CId [3.3] Completed on service for Post Registration Customer [3.4] Working on component for Customer Registraion		[3.1] Verification of Api using Postman need to be done [3.2] Integration of service and component is left	[3.1] Completed Api for GetCustList [3.2] Completed Api for GetCustList by CId [3.3] Completed the Service for Post Registration of Customer [3.4] Done with the component for Customer Registraion
[Divya Marikanti]	After User login to the account he/she can navigate to the page with sidebar there they can view their transaction details of their current, pervious bookings and passengers related to that booking by using transaction Id	[4.1]Api Get List Of Passenger By Tid [4.2] Api PostPassengerByTid [4.3]Service Transaction by custid [4.4] SideBar for customer [4.5]Database for passenger	[4.1]Completed Db for Passenger [4.2]Completed Api get list of passenger by Tid [4.3]Completed Api post passenger by Tid		[4.1] Verification of Db,Api's	[4.1]Completed Api Get List Of Passenger By Tid [4.2]Completed Api PostPassengerByTid [4.3]Done with Service Transaction by custid [4.4]Completed SideBar of customer [4.5] Created Database for passenge

30-08-2022(Tuesday)

	User Story	TO DO	In Progress	Impediment	To Verify	Done
[Jitin Sharma]	Seat Mapping is the major component that provides both great User Interface and User Experince while selecting the seats by the customers. Building the logic and design is essential. Also once we have all the responses by the customer we have to send the responses to the API so that it can give the get request for final summary.	[1.1]Create a DB for transaction details [1.2]Create a DB for Transaction Seat [1.3] Create an Api for accepting the response from frontend called Transaction Details [1.4] Create a Api for PostList by tId [1.5] Work on the component of Seat Mapping and the design of seat Mapping	[1.1]Created a DB for transaction details [1.2] Created a DB for Transaction Seat [1.3]Created the Api for PostList by tId [1.4] Working on Api for accepting the response from frontend called Transaction Details	While working with the Create Transaction Details Api, we were getting a cyclic error issue with the attributes used in Transaction table and other related tables. Thiese tables were used in multiple internal apis. So to resolve this, first we tried changing the schema but still the same cyclic issue was presenting. Thereafter by changing the responses as Array of Arrays, we solved the error. Passenger array is used in transaction details, also the seats array is used, so by this way we resolved the error.	[1.1]Verification of APIs with PostMan is pending [1.2] Component of Seat Mapping and Design is to be integrated to check the working functionality.	[1.1]Created a DB for transaction details [1.2]Created a DB for Transaction Seat [1.3] Created an Api for accepting the response from frontend called Transaction Details [1.4]Created a Api for PostList by tId [1.5] Completed the component of Seat Mapping and the design of seat Mapping
[Anshul Bansal]	After the customer selects the bus for making a reservation, he/she should be able to select the seats using the user-friendly Seat Map. And then after the selection of the seats and passengers, there must be an online payment gateway for the customer to make a booking for himself.	[2.1] API for updating the set of seats status to not available. [2.2] API for updating the set of seats status to available. [2.3] API for getting the transaction details for customer id. [2.4] Implement SeatMap component to show the seat structure for each bus [2.5] Create CardDetais component to accept the card details for completing the transaction.	[2.1] Created API for updating the set of seats status to not available. [2.2] Created API for updating the set of seats status to available. [2.3] Created API for getting the transaction details for customer id. [2.4] Created CardDetais component to accept the card details for completing the transaction.	SeatMap component is the USP of our application. The challenge was that we have to give the functionality for a customer to flexible select the seats from the grid of seats, booked seats should not be available for selection and a customer can flexibly select/deselect any number of seats from the available ones. So we resolved this by maintaining three states (Booked/Available/Selected) for each seat (cell in the grid) and on each click we were changing the state of the seat depending on the previous state of the seat. After the form submission total count and the seat no's of the selected seats should be stored for making a transaction after successful payment.	[2.1] Verification of the APIs with Postman then with the Client services is pending. [2.2] Verification of the component with the services is pending.	[2.1] Completed API for updating the set of seats status to not available. [2.2] Completed API for updating the set of seats status to available. [2.3] Completed API for getting the transaction details for customer id. [2.4] Completed CardDetais component to accept the card details for completing the transaction. [2.5] Completed SeatMap component to show the seat structure for each bus .
[Omkar Bhagwat]	If user is already registered then on the home page, user should be able to Login to further access the functionalities of the system like view the booking details of the Customer. While Logging in, user has to enter his/her Email and Password to get into the system.	[3.1] Api putCustomer to edit customer details [3.2] Create service for customer login [3.3] Create Component for Customer Login [3.4] Create Component for details of bookings for particular TId	[3.1] Completed Api putCustomer to edit customer details [3.2] Completed Create service for customer login [3.3] Completed Create Component for Customer Login [3.4] Working on Create Component for details of bookings for particular TId		[3.1] Verification of Api using Postman is left [3.2] Integration of service and component need to be done	[3.1] Completed Api putCustomer to edit customer details [3.2] Completed the Service for customer login [3.3] Done with the Component for Customer Login [3.4] Done with the Component for details of bookings for particular TId
[Divya Marikanti]	In mybooking component List of passengers can be Viewed by using tid.After seat booking user will be navigate the page where they should enter the passenger details.	[4.1]Service for Passenger by tid [4.2]Passenger Component [4.3]My Booking Component [4.4]Api for Set/Reset Seats	[4.1]completed service for Passenger by tid [4.2]Completed Passenger Component		Verification of Api,services, Components,	[4.1]Completed the Service for Passenger by tid [4.2]Done with Passenger Component [4.3]Done with My Booking Component [4.4]Created Api for Set/Reset Seats

SPRINT - 3(Advanced Functionalities)

01-09-2022(Thursday)

	User Story	TO DO	In Progress	Impediment	To Verify	Done
1. [Jitin Sharma]	1. Once the bookign is made, therse should also be a fucntionality to cancel the booking, so the money should be refunded to the wallet. Development of that functionality is a necessity.	[1.1]Create a service for Cancelling the booking that will lead to refund initiation [1.2]Create a Booking Cancel Component and design it to redirect it to wallet component, also create a button to cancel in My Booking [1.3] Search on print functionality.	[1.1]Created a service for Cancelling the booking that will lead to refund initiation [1.2] Working on the booking cancel component.		[1.1] Verification is to be done with final UI once prepared. [1.2] Check for the navigation to Wallet Component.	[1.1]Created a service for Cancelling the booking that will lead to refund initiation [1.2] Created a Booking Cancel Component and design it to redirect it to wallet component, also create a button to cancel in My Booking [1.3] Got the method print functionality.
[Anshul Bansal]	Functionality to change password for registered customer so that he/she should be able to change his/her password when they are logged in.	[2.1] Create change password API. [2.2] Create service method for change password. [2.3] Create change password Component.	[2.1] Created a change password API [2.2] Created a service method for change password. [2.3] Working on the change password component.		[2.1] Verification of the Api with the Postman is pending. [2.2] Integration of the service method with the Api is pending.	[2.1] Completed change password API. [2.2] Completed service method for change password. [2.3] Completed change password Component. [2.4] Done the integration of Api and service method.
[Omkar Bhagwat]	If Customer is booking seats or canceling the already booked seats, money from his/her Wallet must be credited or deducted.	[3.1] Update the DB for 'Customer' and add new column 'Wallet' [3.1] Create Wallet Service for Customer [3.2] Create Component for Customer to view Wallet	[3.1] Updated the DB for 'Customer' and add new column 'Wallet' [3.1] Created Wallet Service for Customer [3.2] Created Component for Customer to view Wallet		[3.1] Verification of the Api with the Postman is pending. [3.2] Integration of the service method with the Api is pending.	[3.1] Updated the DB for 'Customer' and add new column 'Wallet' [3.1] Created Wallet Service for Customer [3.2] Completed the Wallet component
[Divya Marikanti]	User can cancel their active bookings.Also able to print the details of transaction and give the feedback	[4.1]Delete transaction API [4.2]Feedback Form [4.3] Print transaction Deatils	[4.1]Created Api For Delete Transaction Api [4.2]Created Feedback		Verification of api and print functionality	[4.1]Delete transaction API [4.2] Feedback Form [4.3]Print transaction Deatils