

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



## **Fundamentals of Software Engineering CS-2004**

---

### **Deliverable 1: Iteration 0 Planning**

#### **Submitted to:**

Ma'am Maheen Arshad

#### **Submitted by:**

- 1. Ismail Ramzan (20i-0941)**
- 2. Musaab Imran (20i-1794)**
- 3. Muhammad Usman Shahid (20i-1797)**

## Contents

<b>Fast Bus Management System .....</b>	<b>3</b>
<b>Nature of Project: .....</b>	<b>3</b>
<b>Scope: .....</b>	<b>3</b>
<b>Functional Requirements: .....</b>	<b>3</b>
<b>User Stories: .....</b>	<b>4</b>
<b>Tasks: .....</b>	<b>14</b>
<b>Project Backlog: .....</b>	<b>20</b>
<b>Team Duties: .....</b>	<b>21</b>

# **Fast Bus Management System**

## **Nature of Project:**

Our project is an android application.

## **Scope:**

The project would cover the whole management of the fast bus system, not only managing but also fulfilling the uses of the rider, driver, and the admin. While showing the path, route, and the details of the travel to the rider, it would also give the driver a clear picture of the travelers. Any requirements and needs except the given functionalities would be out of this project's scope. Any change in the given deliverables or core functionalities would cause a change order in the schedule.

## **Functional Requirements:**

<b>Requirement #</b>	<b>Description</b>
1	Login system for students, bus drivers, and admin.
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.
6	Students and drivers register who are traveling for the first time against their Gmail id.
7	Daily bus and route allotment by the admin.
8	Account deletion of students and drivers who have left.
9	And adding new students and drivers, also creating their accounts.
10	Maintaining buses data, also adding and removing the buses.
11	Sign out or log out of the account after the user has used stopped using the app.

# User Stories:

01

<b><u>ID: US-01</u></b>	<b><u>Title: User Login</u></b>	<b><u>Priority:</u></b> High
<ul style="list-style-type: none"><li>➤ <b>Created by:</b> Group</li><li>➤ <b>Date created:</b> 01/03/2022</li><li>➤ <b>Last updated by:</b>04/03/2022</li></ul> <p>As a customer, I want to have proper login for different users which includes students, drivers, and admin. This must include two-factor authentication for the admin because he would be accessing all the information.</p> <b><u>Description:</u></b> <ul style="list-style-type: none"><li>o Users will have to log in.</li><li>o Users would log in through their Gmail ids.</li><li>o There must be a strong check on the passwords.</li><li>o There must be a captcha.</li></ul> <b><u>Acceptance criteria:</u></b> <ul style="list-style-type: none"><li>o Check for new users to make their accounts and add the information to the database.</li><li>o If the user is already registered then allow him/her to enter.</li></ul>		<b><u>Estimate hours:</u></b>  5-6





High

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

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

## 4-5

- 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 an arrival time of different stopping points.

<p><b><u>ID: US-05</u></b>    <b><u>Title: Student Info Log</u></b></p> <ul style="list-style-type: none"> <li>➤ <b>Created by:</b> Group</li> <li>➤ <b>Date created:</b> 01/03/2022</li> <li>➤ <b>Last updated by:</b>04/03/2022</li> </ul> <p>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.</p> <p><b><u>Description:</u></b></p> <p>Things to be displayed to the driver:</p> <ul style="list-style-type: none"> <li>○ Boarding number against the names of the students.</li> <li>○ Picking points of the same route students.</li> </ul> <p><b><u>Acceptance criteria:</u></b></p> <ul style="list-style-type: none"> <li>○ The boarding number should be the same in the list of drivers and shown to students. (important)</li> <li>○ The picking points must be near the predefined routes.</li> </ul>	<p><b><u>Priority:</u></b></p> <p>High</p> <p><b><u>Estimate hours:</u></b></p> <p>4-5</p>
--	--





<p><b><u>ID: US-07</u></b>    <b><u>Title: Bus Allotment</u></b></p> <p>➤ <b>Created by:</b> Group</p> <p>➤ <b>Date created:</b> 01/03/2022</p> <p>➤ <b>Last updated by:</b>04/03/2022</p> <p>As a customer, I want that the admin should weekly or daily update the status of the buses their routes and which driver would be driving which vehicle.</p>	<p><b><u>Priority:</u></b></p> <p>Low</p>
<p><b><u>Description:</u></b></p> <p>The admin can see the list of the bus, drivers, and routes so he can allocate a bus to a driver for a specific route. This has to be updated till 4 pm so that students can book the bus till 12 pm-midnight.</p>	<p><b><u>Estimate hours:</u></b></p> <p>5-6</p>
<p><b><u>Acceptance criteria:</u></b></p> <ul style="list-style-type: none"> <li>○ The admin has to update the bus, driver, and route status till 4 pm.</li> <li>○ More than one bus can't be allocated to one driver.</li> <li>○ Travel details to be updated regularly.</li> </ul>	





<p><b><u>ID: US-10</u></b>    <b><u>Title: Bus Management</u></b></p> <p>➤ <b>Created by:</b> Group  ➤ <b>Date created:</b> 01/03/2022  ➤ <b>Last updated by:</b>04/03/2022</p> <p>As a customer, I want that the admin should have a page where he can view the state of all the buses and the data about when was the last time they went for maintenance after the month a message should be shown to the admin that the bus needs need maintenance.</p> <p><b><u>Description:</u></b></p> <p>The admin should be able to view the state of buses and the last date of their maintenance because the vehicle needs maintenance regularly. For a bus system, the maintenance of the bus is a prime thing.</p> <p><b><u>Acceptance criteria:</u></b></p> <ul style="list-style-type: none"> <li>○ If one month is completed after the maintenance of the bus then it should be shown on the allotment list.</li> <li>○ There should be a list named “maintenance” which must include buses that need immediate maintenance.</li> <li>○ Daily reminders should be given unless the maintenance of the bus is done.</li> </ul>	<p><b><u>Priority:</u></b></p> <p>Moderate</p> <p><b><u>Estimate hours:</u></b></p> <p>5-6</p>
---	--

<p><b><u>ID: US-11</u></b>    <b><u>Title: Log-out</u></b></p> <ul style="list-style-type: none"> <li>➤ <b>Created by:</b> Group</li> <li>➤ <b>Date created:</b> 01/03/2022</li> <li>➤ <b>Last updated by:</b>04/03/2022</li> </ul> <p>As a customer, I want that that whenever the user has done his/her work their account should have a proper logout to have a sense of security. So, their accounts couldn't be accessed by others.</p> <p><b><u>Description:</u></b></p> <p>The user while using the app should be able to sign out or log out. The log-out should be proper and if the user wants to use the app again then he/she should log in again.</p> <p><b><u>Acceptance criteria:</u></b></p> <ul style="list-style-type: none"> <li>○ If the user stops using the app exactly after one min log-out the account of the user for his/her security.</li> </ul>	<p><b><u>Priority:</u></b></p> <p>High</p> <p><b><u>Estimate hours:</u></b></p> <p>5-6</p>
--	--

## **Tasks:**

<b>US01: User login</b>	<b>T101-</b> Design the Frontend UI Prototype using Adobe XD <b>T102</b> – Code the Frontend Using the Adobe XD <b>T103</b> – Fix the Database Structure for the login as specified fields <b>T104-</b> Creating & making sure that the sure solves the ReCAPTCHA using the reCAPTCHA API <b>T105-</b> Send the ReCAPTCHA status and the user credentials to the backend server <b>T106-</b> Server validation of the credentials and the user interface <b>T107-</b> When verified User redirection to the Correct Address
<b>US01 + US06: Admin Registration &amp; Login</b>	<b>T201-</b> Hardcode the credentials of the admin at the backend <b>T202</b> – A Different Private Route for the admin Login <b>T203</b> – State Variables Creation for the Admin <b>T203</b> – Admin login State identifier into the database <b>T205</b> – Verifying Admin credentials & Gaining access to the admin panel
<b>US01 + US06: Driver Registration and Login</b>	<b>T301</b> – Design the User interface & connect it to the Firebase database <b>T302</b> – Take driver information and stores in the database and forward the request to the admin <b>T303</b> - Ask for ReCAPTCHA to login & validate the credentials with the database
<b>US06: Admin Panel &amp; its Functionality</b>	<b>T401-</b> Create the UI After creating the Prototype <b>T402-</b> Creation of different Routes for the admin <b>T403-</b> Adding Restriction to the admin panel <b>T405-</b> Hardcoding the Driver Credentials into the database <b>T406-</b> Create functionality of secure Admin Login by restricting its panel and the login form.

<p><b>US06:</b> <b>Admin</b> <b>Registration</b></p>	<p><b>T501</b> – Create the Prototype of the User Registration Form</p> <p><b>T502</b> – Code for the frontend User interface development using the Prototype</p> <p><b>T503</b> – Create the Database By using the Firebase</p> <p><b>T504</b> – Connect the Flutter Application with the database</p> <p><b>T505</b> – Make Authentication using the Firebase and Backend as a service</p> <p><b>T506</b> – Admin do not need registration he's credentialed will be hardcoded with a secure password</p> <p><b>T507</b> – Once, the user clicks register state variables will be updated and he will be redirected to the login screen and the database will be updated with the new information</p>
<p><b>US11:</b> <b>Logout</b></p>	<p><b>T601-</b> Remove the login State from the database</p> <p><b>T602-</b> Uses the backend as a Service to get the state variables</p> <p><b>T603-</b> Change the state variables in the frontend</p> <p><b>T604-</b> Making sure to have data correctly stored &amp; do not change on sign on again</p> <p><b>T605-</b> Using the Permanent store data to show to User even when logged out using the Firebase Auth service</p>
<p><b>US03:</b> <b>Booking</b> <b>Details</b></p>	<p><b>T701-</b>When entered the data from the user its state will be stored at the firebase database</p> <p><b>T702-</b> Once the state is stored, it can be easily fetched from the backend server to display.</p> <p><b>T703-</b> The state of the Booking of the user will also be visible to the bus driver fetched from the database</p> <p><b>T704-</b> The database will be cleared when the bus reaches its destination so the state does not affect the next drives</p> <p><b>T705-</b> Once the user books a ride. It's variable in the backend and the backend will be updated so that he cannot book multiple seats and a random id will be generated against the user.</p>



<p><b>US04: Travel Details</b></p>	<p><b>T801-</b> Once the student Registers the ride the google map API will come into play and we show the user his current location</p> <p><b>T802-</b> After showing the current location we show the location of the bus stop which also uses Google API</p> <p><b>T803-</b> Driver will be able to identify the seats of the student using the random id (created with dart math library) and present at the database at the time of registration of bus</p> <p><b>T804-</b> The driver will be provided with that random list and he could match the student's name with his id to check whether they are the right candidate or not</p> <p><b>T805-</b> Once a driver reaches its destination the backend information (random id &amp; student seats number) will be removed.</p>
<p><b>US05: Student Info Log</b></p>	<p><b>T901-</b> Student Enters the credentials and its credentials are validated from the backend service by firebase</p> <p><b>T902-</b> If credentials are correct then student Challain information and other info will be displayed on the frontend, fetched from the database</p> <p><b>T903-</b> Creating &amp; making sure that the student solves the ReCAPTCHA using the reCAPTCHA API</p> <p><b>T904-</b> After verifying the ReCAPTCHA status &amp; comparing hashes with the database Student could see his account and other information</p> <p><b>T905-</b> State Variables will be fetched and will be updated in the application provided by the backend service</p>
<p><b>US07 + US10: Bus Allotment And Bus Management</b></p>	<p><b>T10/01-</b> Bus information will be hardcoded in the database at the initial stage but will be changeable by admin ONLY</p> <p><b>T10/02-</b> Admin Clicks on the add button to add the bus to the database &amp; to add a driver to it.</p> <p><b>T10/03-</b> Admin can click on allocate Button on admin panel to allocate the bus to a particular driver</p> <p><b>T10/04-</b> The details of the allocation will be updated on the backed &amp; database</p> <p><b>T10/05-</b> The driver &amp; student can see the bus Number and its driver on their user interface which will be fetched from the database as updated by the admin.</p> <p><b>T10/06-</b> The admin would have a database link showing the time of bus service.</p>

	<p><b>T10/07-</b> after a month the admin would see a notification for the service of the buses respectively.</p>
<p><b>US08+US09:</b> <b>Account Management</b> <b>01 + 02</b></p>	<p><b>T11/01-</b> Fetch the Driver Requests from the database.  <b>T11/02-</b> Adding Driver Allocation Service to the admin Panel.  <b>T11/03-</b> ADD/REMOVE Driver Button for the Admin with Functionality.  <b>T11/04-</b> Add/Remove Bus Functionality for the Admin.  <b>T11/05-</b> Bus allocation to the driver by Admin.  <b>T11/06-</b> Remove bus or driver Functionality for admin using the state variables.</p>
<p><b>US09:</b> <b>Driver Panel Creation</b></p>	<p><b>T12/01-</b> Creation of UI having the several Buttons, See Student Route, Bus Number &amp; Map  <b>T12/02-</b> Fetch the Student Name, Seat Number &amp; Random Token (created by rand manager) and Display The UI</p>
<p><b>US09+US06:</b> <b>Accounts Management</b></p>	<p><b>T13/01-</b> Three accounts functionality will be available to the backend server to validate from (Firebase database)  <b>T13/02-</b> Admin account credentials will be hardcoded for the admin and it will use the strong password policy whose password hash will also be in the database  <b>T13/03-</b> User registration will be carried out from the backend service provided by the firebase  <b>T13/04-</b> Admin account will be responsible for using the firebase API to send the data to the database  <b>T13/05-</b> Each state of activity (logged in or out) will be checked from the state variables</p>

<b>US02: Accounts</b>	<b>T14/01-</b> Connect to the firestore to check the payment of such a student. <b>T14/02-</b> Check if the fee has been paid or not. <b>T14/03-</b> Message the driver on 30 <sup>th</sup> to check the bank account for salary check.
---------------------------	---

## **Project Backlog:**

### **Sprint 1 (2.5 weeks):**

US01- User Login

US06- User Registration

US11- Logout

### **Sprint 2 (2 weeks):**

US03- Booking Details

US04- Travel Details

### **Sprint 3 (2 weeks):**

US08- Account Management 01

US09- Account Management 02

### **Sprint 4 (2 weeks):**

US07- Bus Allotment

US10- Bus Management

### **Sprint 5 (2 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 user story is 7 days.

## **Team Duties:**

### **Mr. Musaab Imran: Scrum Master, developer**

- Assign roles, tasks, check and report process.
- Design app prototype on Adobe XD.
- Develop core functions of the application.
- Documentation.

### **Mr. Usman Shahid: Administrator, designer, developer**

- Design app prototype on Adobe XD.
- Design and develop firebase.
- Unit and integration testing.
- Develop core functions of the app.

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

- Design and develop cloud database.
- Design app prototype on Adobe XD.
- Unit and integration testing.
- Develop core functions of the app.