IMPROVING APP PERFORMANCE USING GETX & MVVM

SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE INDUSTRIAL TRAINING REPORT

By

Kumawat Lakhan Makhanlal 1906055

UNDER THE SUPERVISION OF

Senior Manager, Troovi

Mr. Gourav Goyat

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY PATNA

(An Institute of National Importance)

MAHENDRU, PATNA, BIHAR - 80000

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY PATNA

This is to certify that **Kumawat Lakhan Makhanlal** (1906055) has completed industrial training in the field entitled "**Improving app performance using getx and MVVM**" under the supervision of **Mr. Gourav Goyat**, Senior Manager, Fleetech Trucking Private Limited (Troovi). This project is bonafide work done by him for the fulfillment of the requirements for the industrial training.

Mr. Gourav Goyat
Supervisor

Dr. Maheshwari Prasad Singh
Head of Department CSE

DECLARATION

I, a 7th semester student, hereby certify that I undertook the industrial training in the field entitled "**Improving app performance using getx and MVVM**" under the supervision of **Mr. Gourav Goyat**, Senior Manager, Fleetech Trucking Private Limited (Troovi). No portion of this work has ever been submitted to another source or copied verbatim from books, essays, or journals.

Kumawat Lakhan Makhanlal
1906055

Place : ----------------------------

Date : ----------------------------

ACKNOWLEDGEMENT

I'd like to take this opportunity to thank everyone who contributed in some way to making this project a success. Without your help, this wouldn't have been possible. Our senior project manager, **Mr. Gourav Goyat**, has been a constant source of inspiration, and I want to express my sincere thanks, respect, and duty to them. Words cannot adequately explain how grateful we are to our parents and friends for their constant support. I bow down to the almighty in particular because of all of his many favors in my life.

Kumawat Lakhan Makhanlal

1906055

Table of Contents

ABSTRACT

Troovi is an app built for the Indian logistics sector by digitizing the manual processes and paperwork involved in the day-to-day activities of transporters, fleet owners, and brokers.We want to change the way the transport business in India is operated at ground level by providing a range of solutions from expense management to driver management, electronic proof of deliveries to informative dashboards, balance settlements to payment reminders.

1. INTRODUCTION

This is an app based on the Indian logistics sector by digitizing the manual processes and paperwork involved in the day-to-day activities of transporters, fleet owners, and brokers. It allows you to manage your transport business digitally to eliminate paperwork difficulties. If you are a fleet or truck owner or you are a company managing transport services then troovi is the best fit for you , it lets you manage all your brokers , fleets owners , trucks , charges , road trips and balance all at one place , just like khata book but in transport system but with more added features . The app provides users with multiple user interfaces .

Some of multi app features are listed below :

Transporters

Manage Supplier Expenses
Create unique LRs/Biltis digitally
Download PDFs and share bills on Whatsapp

Brokers

Track and settle customer due balances
Pay advances and view bills
Record payment made and received systematically

Fleet Owners

Create trips with details like source,destination,freight charges etc
Check driver related expenses
Upload proof of deliveries and track POD status

Drivers

Manage and settle expenses done by your driver exclusively
Switch our app easily into driver mode curated especially for drivers
Approve or reject expenses by a driver and get easily notified about PODs.

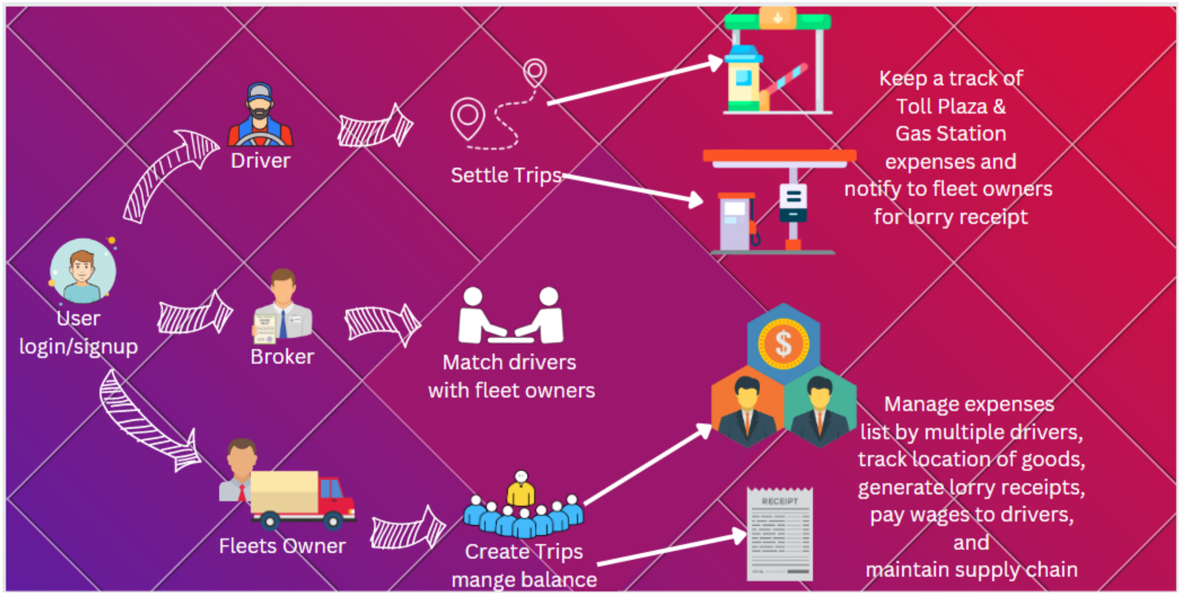

Driver
Settle Trips
Keep a track of Toll Plaza & Gas Station expenses and notify to fleet owners for lorry receipt
User login/signup
Broker
Match drivers with fleet owners
Manage expenses list by multiple drivers, track location of goods, generate lorry receipts, pay wages to drivers, and maintain supply chain
Fleets Owner
Create Trips mange balance
RECEIPT

Fig 1.1 Working Diagram of Drivers,Brokers and Fleets owners

2. OBJECTIVE

Given below are the key objectives of the application:

1. Expense Management
2. Settlement Reminders
3. Driver Management
4. User Management
5. Reports and Dashboard
6. Document Generation

Manage Supplier and Trip Expenses, Generate supplier challans, Track due balances and send settlement reminders to parties and customers through WhatsApp or SMS, Manage driver-related expenses independently and get electronic POD submissions from drivers directly, Grant Role-Based Access to your teammates and manage them centrally through the admin panel, Keep track of all trips, payments, and balances from one central dashboard. Download and share reports directly through the app Create, scan update all the trip documents from Invoices to LR's

3. TOOLS & TECHNOLOGIES USED

Frontend : Flutter & Dart on mobile and ReactJs on Web
Backend : AWS Amplify and Python, AWS Cognitive Services

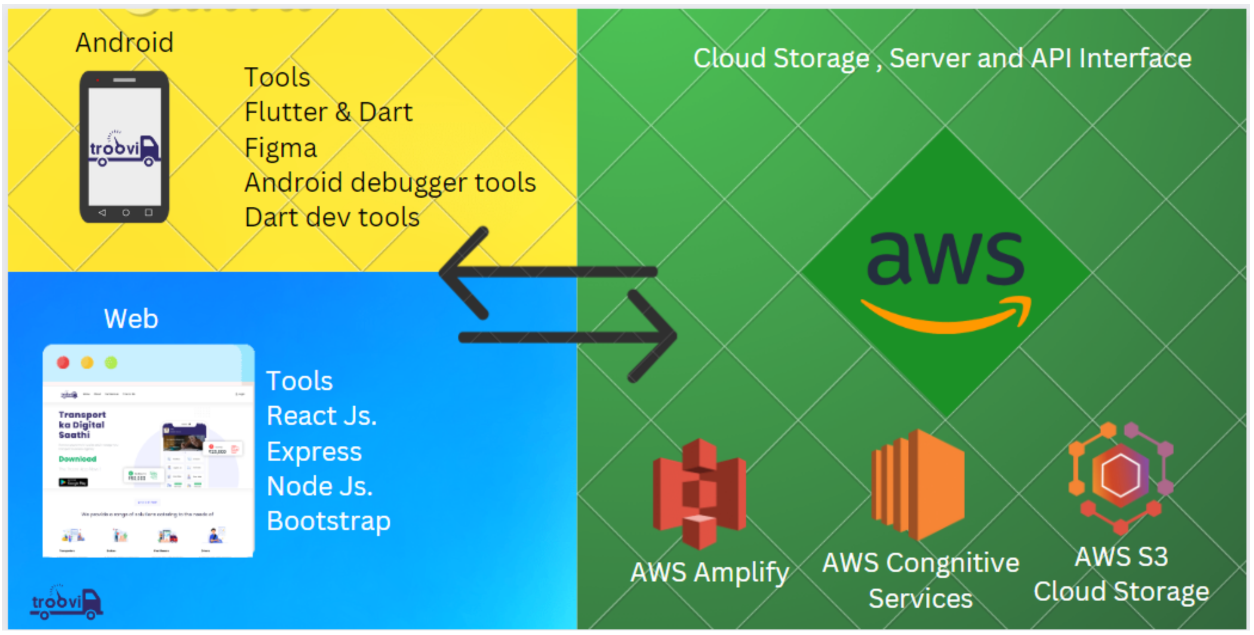


Figure 3.1 Framework Diagram for the Proposed Model

4. PROBLEM STATEMENT

Perform analysis and develop a solution for app optimization, refactor app architecture and integrate new better plugins to minimize intrinsic operations.

5. PROPOSED SOLUTION

5.1 Architecture

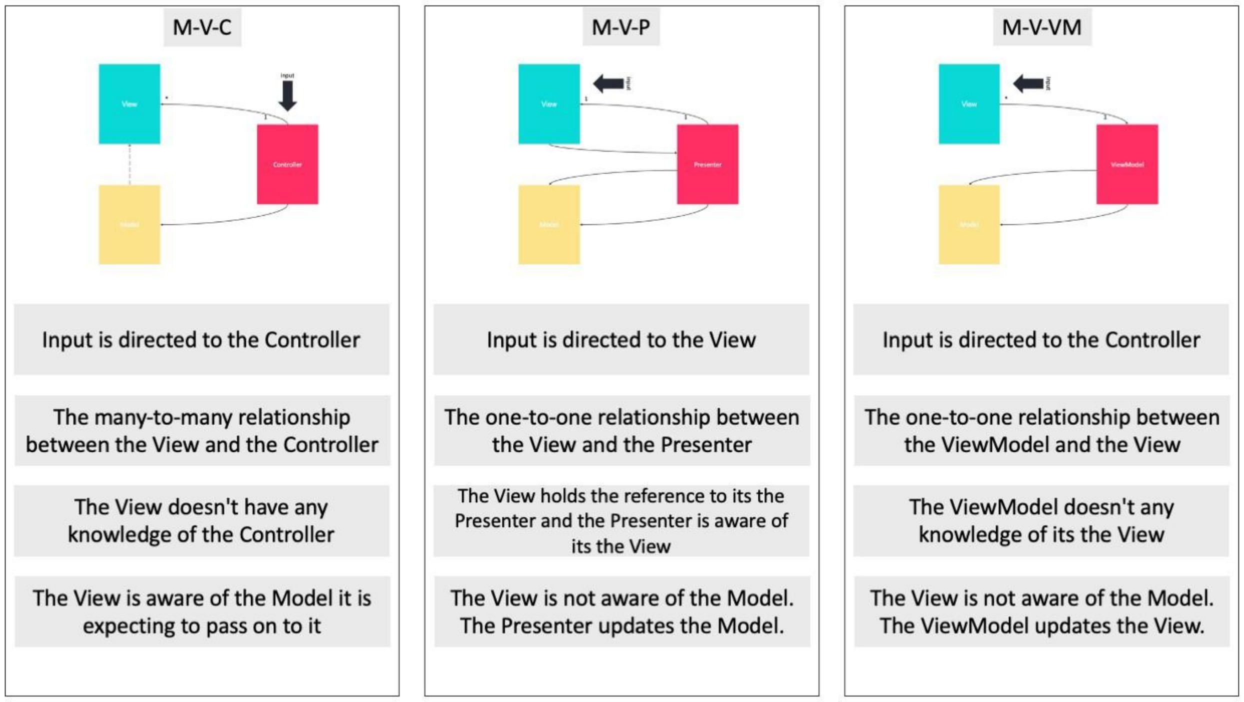
Use of GETX Flutter with MVVM Architecture

The aim of these architectures is to separate the responsibilities of visualizing, processing, and data management for UI applications. The main idea of all these patterns is to organize the project in a proper way so that all the codes get covered in the Unit Testing. Moreover, it is very helpful in the maintenance of the software, to add and remove features and developers can keep a track of various crucial logic parts.

They help in Modularity

Flexibility

Testability



M-V-C	M-V-P	M-V-VM
Input is directed to the Controller	Input is directed to the View	Input is directed to the Controller
The many-to-many relationship between the View and the Controller	The one-to-one relationship between the View and the Presenter	The one-to-one relationship between the ViewModel and the View
The View doesn't have any knowledge of the Controller	The View holds the reference to its the Presenter and the Presenter is aware of its the View	The ViewModel doesn't any knowledge of its the View
The View is aware of the Model it is expecting to pass on to it	The View is not aware of the Model. The Presenter updates the Model.	The View is not aware of the Model. The ViewModel updates the View.

Figure 5.1 Model-View- ViewModel (MVVM), Model-View-Presenter (MVP), and Model-View-Controller

Widgets

Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the app. And the structure of the code of an app is a tree of widgets. There are broadly two types of widgets in the flutter:

1. Stateless Widget

The life cycle of stateless widgets is simple; there's only one stage: the build method. As soon as the widget gets built, the build method gets automatically called where you are supposed to create whatever appearance you want to add up in your application.

2. Stateful Widget

When a Flutter builds a StatefulWidget, it creates a State object. This object is where all the mutable state for that widget is held.

The concept of state

1. data used might change.
2. data can't be read synchronously when the widget is built.

5.2 GETX Flutter

GetX is an extra-light and powerful solution for Flutter. It combines high-performance state management, intelligent dependency injection, and route management quickly and practically.

GetX has 3 basic principles which are listed below:

PERFORMANCE: GetX is focused on performance and minimum consumption of resources. GetX does not use Streams or ChangeNotifier.

PRODUCTIVITY: GetX uses an easy and pleasant syntax. No matter what you want to do, there is always an easier way with GetX. It will save hours of development and will provide the maximum performance your application can deliver.

Generally, the developer should be concerned with removing controllers from memory. With GetX this is not necessary because resources are removed from memory when they are not used by default. If you want to keep it in memory, you must explicitly declare "permanent: true" in your dependency.

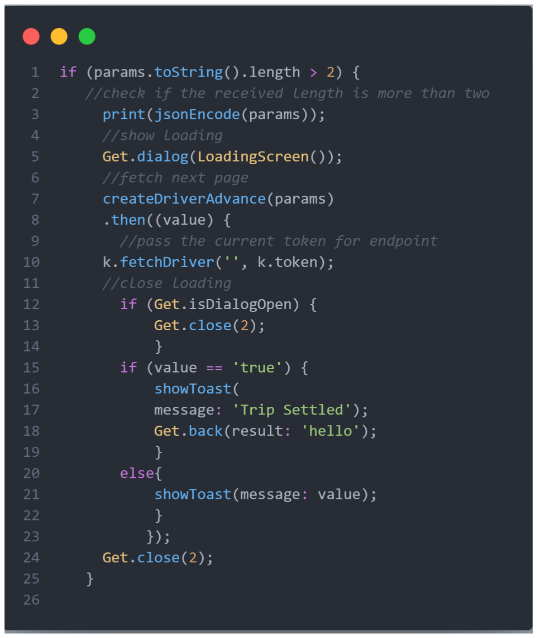
ORGANIZATION: GetX allows the total decoupling of the View, presentation logic, business logic, dependency injection, and navigation. You do not need context to navigate between routes, so you are not dependent on the widget tree (visualization) for this. You don't need context to access your controllers/blocs through an inheritedWidget, so you completely decouple your presentation logic and business logic from your visualization layer. You do not need to inject your Controllers/Models/Blocs classes into your widget tree through MultiProviders. For this, GetX uses its own dependency injection feature, decoupling the DI from its view completely.

With GetX you know where to find each feature of your application, having clean code by default. In addition to making maintenance easy, this makes the sharing of modules something that until then in Flutter was unthinkable, something totally possible.

6. MODEL DESCRIPTION

6.1 ALGORITHMS

Pagination & Smooth Scrolling



```
if (params.toString().length > 2) {
   //check if the received length is more than two
     print(jsonEncode(params));
     //show loading
     Get.dialog(LoadingScreen());
     //fetch next page
     createDriverAdvance(params)
     .then((value) {
       //pass the current token for endpoint
     k.fetchDriver('', k.token);
     //close loading
       if (Get.isDialogOpen) {
           Get.close(2);
           }
       if (value == 'true') {
           showToast(
           message: 'Trip Settled');
           Get.back(result: 'hello');
           }
       else{
           showToast(message: value);
           }
          });
     Get.close(2);
   }
```

MVVM Pattern

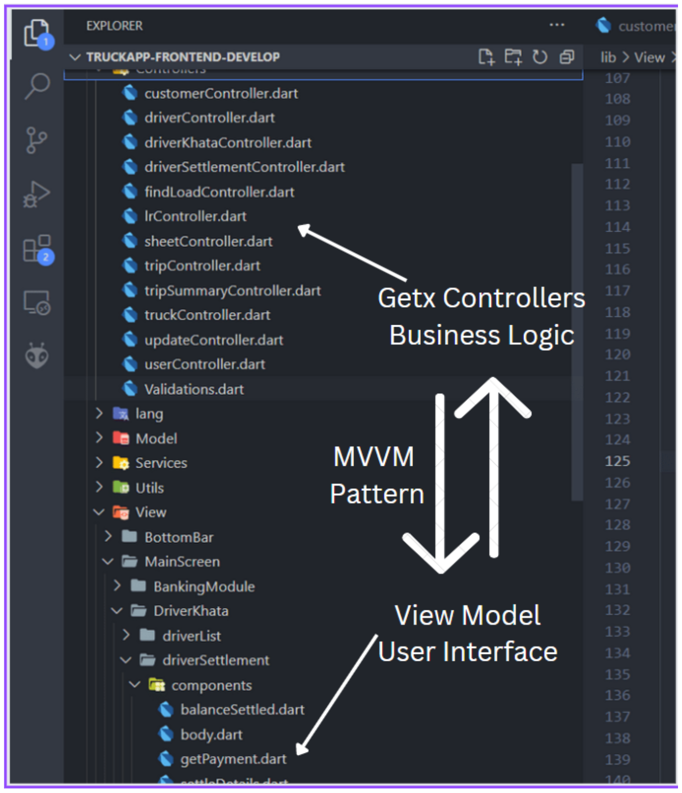
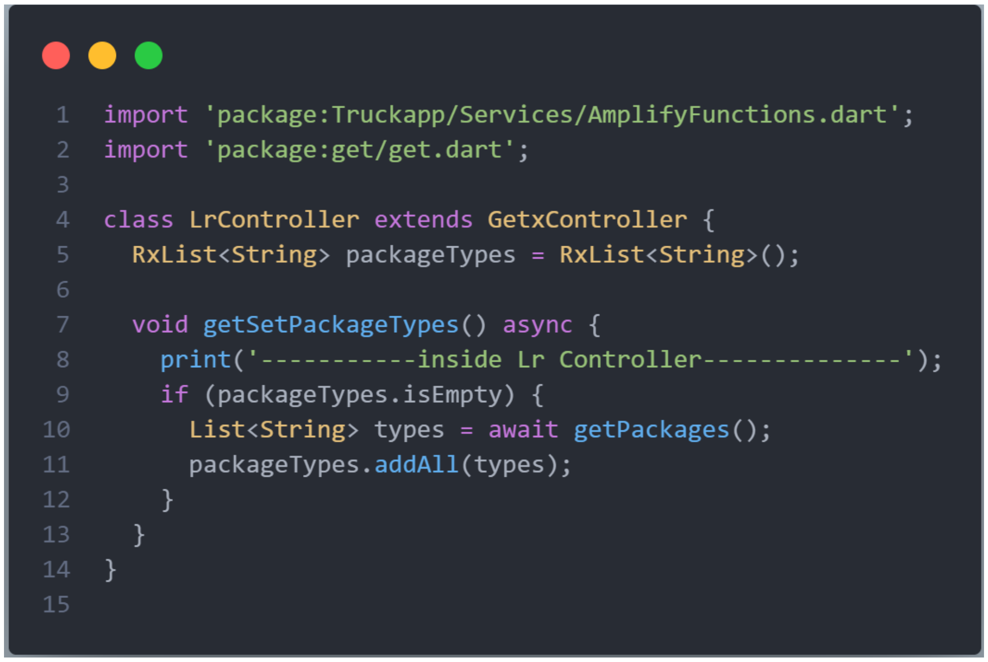


Fig 6.2: MVVM Style pattern in folder structure and modeling



```dart
import 'package:Truckapp/Services/AmplifyFunctions.dart';
import 'package:get/get.dart';

class LrController extends GetxController {
  RxList<String> packageTypes = RxList<String>();

  void getSetPackageTypes() async {
    print('-----------inside Lr Controller--------------');
    if (packageTypes.isEmpty) {
      List<String> types = await getPackages();
      packageTypes.addAll(types);
    }
  }
}

```

Fig 6.3: GetX Class Controller

7. IMPLEMENTATION RESULTS

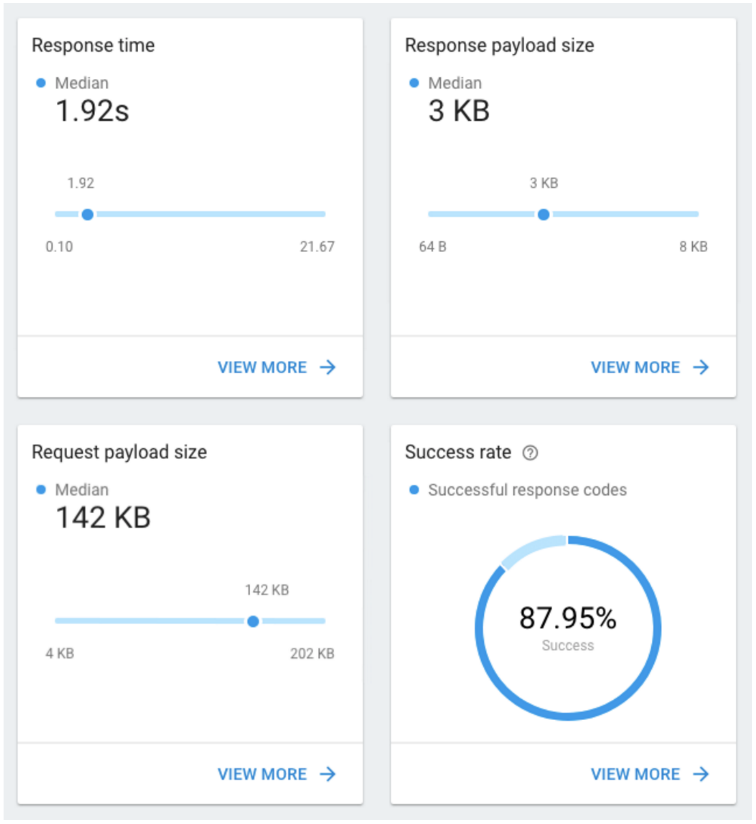


Fig 7.1 Performance Analysis Firebase Tools

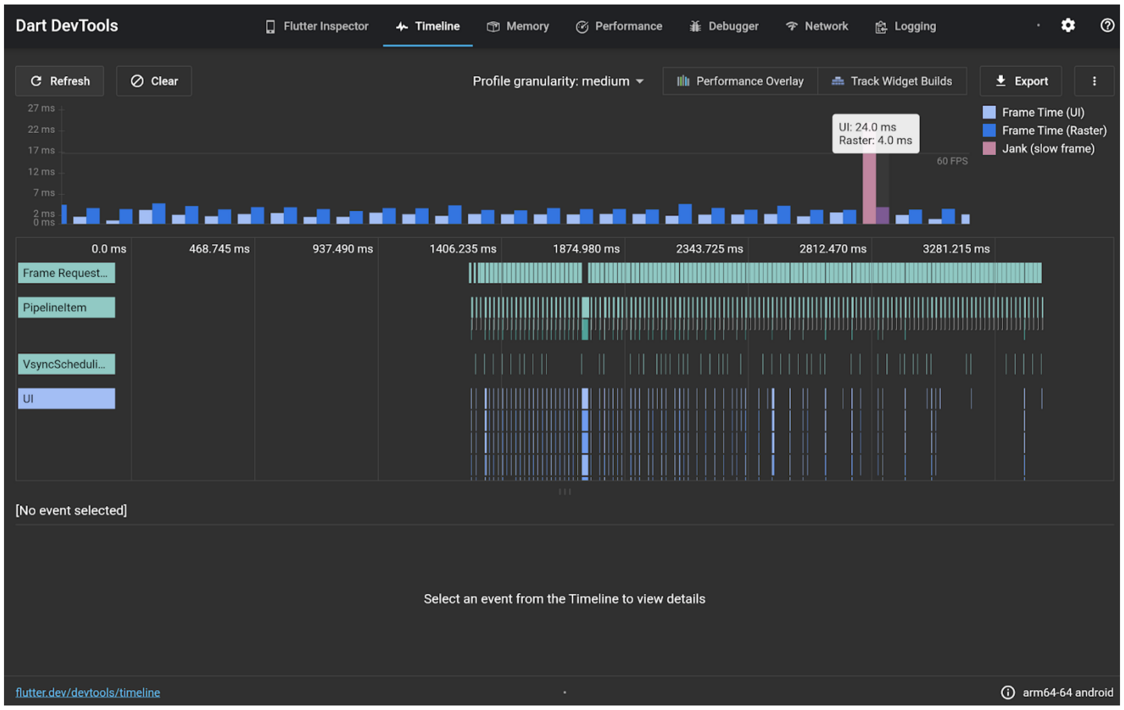
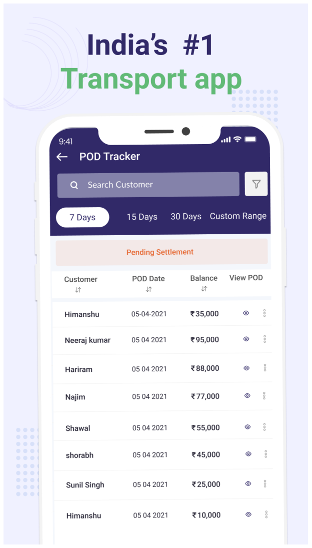


Fig 7.2 Dart Dev Tools (Performance Analysis)

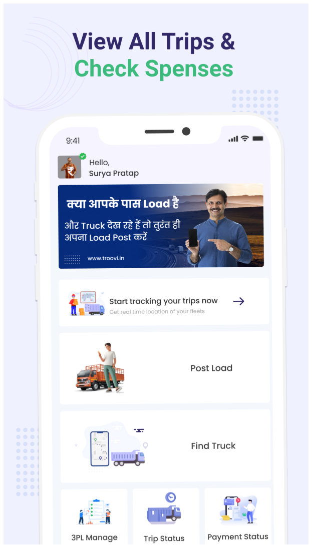
8. PREVIEW



India's #1
Transport app

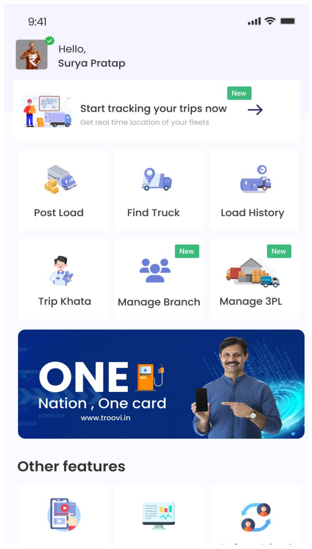
9:41
POD Tracker
Search Customer
7 Days | 15 Days | 30 Days | Custom Range
Pending Settlement

Customer	POD Date	Balance	View POD
Himanshu	05-04-2021	₹35,000	
Neeraj kumar	05 04 2021	₹95,000	
Hariram	05 04 2021	₹88,000	
Najim	05 04 2021	₹77,000	
Shawal	05 04 2021	₹55,000	
shorabh	05 04 2021	₹45,000	
Sunil Singh	05 04 2021	₹25,000	
Himanshu	05 04 2021	₹10,000	

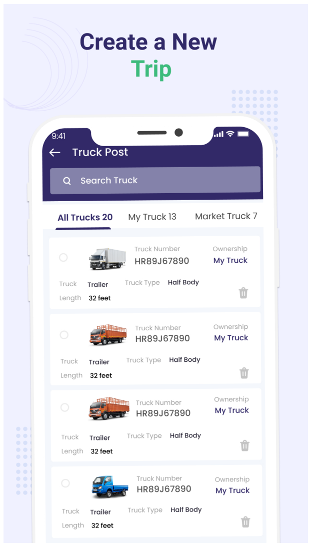


View All Trips &
Check Spenses

9:41
Hello,
Surya Pratap
क्या आपके पास Load है
और Truck देख रहे हैं तो तुरंत ही अपना Load Post करें
www.troovi.in
Start tracking your trips now
Get real time location of your fleets
Post Load
Find Truck
3PL Manage | Trip Status | Payment Status



9:41
Hello,
Surya Pratap
New
Start tracking your trips now
Get real time location of your fleets
Post Load | Find Truck | Load History
Trip Khata | New Manage Branch | New Manage 3PL
ONE
Nation , One card
www.troovi.in
Other features



Create a New
Trip

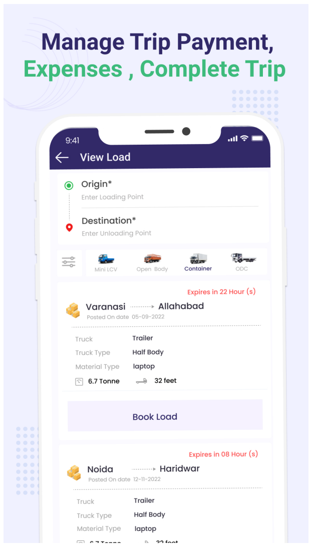
9:41
Truck Post
Search Truck
All Trucks 20 | My Truck 13 | Market Truck 7

Truck Number HR89J67890 — Ownership My Truck — Truck Trailer — Truck Type Half Body — Length 32 feet

Truck Number HR89J67890 — Ownership My Truck — Truck Trailer — Truck Type Half Body — Length 32 feet

Truck Number HR89J67890 — Ownership My Truck — Truck Trailer — Truck Type Half Body — Length 32 feet

Truck Number HR89J67890 — Ownership My Truck — Truck Trailer — Truck Type Half Body — Length 32 feet

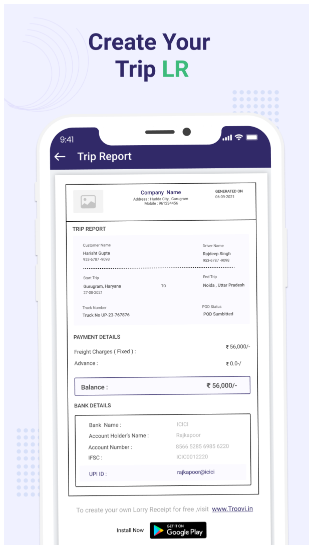


Manage Trip Payment,
Expenses , Complete Trip

9:41
View Load
Origin*
Enter Loading Point
Destination*
Enter Unloading Point
Mini LCV | Open Body | Container | ODC

Expires in 22 Hour (s)
Varanasi → Allahabad
Posted On date 05-09-2022
Truck: Trailer
Truck Type: Half Body
Material Type: laptop
6.7 Tonne — 32 feet
Book Load

Expires in 08 Hour (s)
Noida → Haridwar
Posted On date 12-11-2022
Truck: Trailer
Truck Type: Half Body
Material Type: laptop



Create Your
Trip LR

9:41
Trip Report
Company Name
Address : Hudda City , Gurugram
Mobile : 961234456
GENERATED ON 06-09-2021

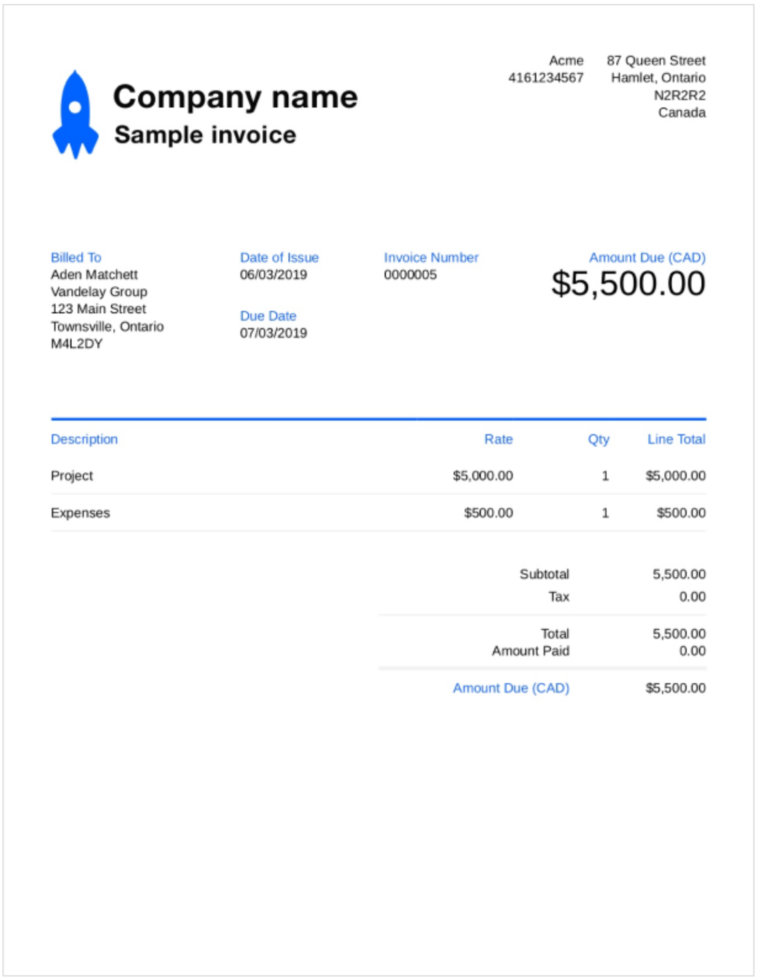
TRIP REPORT
Customer Name: Harisht Gupta, 953-6787 -9098
Driver Name: Rajdeep Singh, 953-6787 -9098
Start Trip: Gurugram, Haryana, 27-08-2021 TO End Trip: Noida , Uttar Pradesh
Truck Number: Truck No UP-23-767876
POD Status: POD Sumbitted

PAYMENT DETAILS
Freight Charges (Fixed) : ₹ 56,000/-
Advance : ₹ 0.0-/
Balance : ₹ 56,000/-

BANK DETAILS
Bank Name : ICICI
Account Holder's Name : Rajkapoor
Account Number : 8566 5285 6985 6220
IFSC : ICIC0012220
UPI ID : rajkapoor@icici

To create your own Lorry Receipt for free ,visit www.Troovi.in
Install Now — GET IT ON Google Play

Table 8.1: Android app preview



Company name
Sample invoice

Acme
4161234567

87 Queen Street
Hamlet, Ontario
N2R2R2
Canada

Billed To	Date of Issue	Invoice Number	Amount Due (CAD)
Aden Matchett Vandelay Group 123 Main Street Townsville, Ontario M4L2DY	06/03/2019 Due Date 07/03/2019	0000005	$5,500.00

Description	Rate	Qty	Line Total
Project	$5,000.00	1	$5,000.00
Expenses	$500.00	1	$500.00

Subtotal	5,500.00
Tax	0.00
Total	5,500.00
Amount Paid	0.00
Amount Due (CAD)	$5,500.00

Fig 8.2 : Generated Lorry Receipt Preview

9. CONCLUSION

Given the influence these technologies have on our modern societies and crucial infrastructure, research and development efforts addressing the security and optimization of an application are unquestionably of paramount importance. I tried to deduce, measure and provide solutions to improve app performance and reduce the complexity. Improve app user interface and user experience.

10. REFERENCES

[1] Jonataslow et al. designed the Getx architecture library in flutter.
[2] Jitesh Mohite et al. Getting started with MVVM architecture in flutter.
[3] Fanhar Tanvir et al. Flutter best practices for improving Performance.
[4] mvvm-model-view-viewmodel-architecture-pattern-in-android
[5] A simple guide on how to set up and how to use Firebase Analytics in Flutter.