**An Industry Oriented Mini Project Report on**

**YIELD DRIVE**

**Submitted to the Department of Computer Science & Engineering, GNITS in the partial fulfillment of the academic requirement for the award of B. Tech (CSE) under JNTUH**

By

**P Gayatri Varma (17251A0508)**

**K Venkata Srihitha (17251A0542)**

**Thota Sudharani (18255A0506)**

under the guidance of

**Dr. A.Sharada**

**Professor**



**Department of Computer Science & Engineering**

**G. Narayanamma Institute of Technology & Science (for Women)**

**Accredited by NBA & NAAC &Affiliated to JNTUH**

**Hyderabad-500 104, T.S, INDIA**

**February, 2021**

**Department of Computer Science & Engineering**

**G. Narayanamma Institute of Technology & Science (for Women)**

**Accredited by NBA & NAAC &Affiliated to JNTUH**

**Hyderabad-500 104, T.S, INDIA**

****

**Certificate**

This is to certify that the Industry Oriented Mini Project report on “**Yield drive**” is a bonafide work carried out by **P Gayatri Varma (17251A0508), K Venkata Srihitha (17251A0542), Thota Sudharani (18255A0506)** in the partial fulfilment for the award of B.Tech degree in Computer Science & Engineering, G. Narayanamma Institute of Technology & Science (For Women), Shaikpet, Hyderabad, affiliated to Jawaharlal Nehru Technological University, Hyderabad under our guidance and supervision.

The results embodied in the Industry Oriented Mini Project have not been submitted to any other University or Institute for the award of any degree or diploma.

**Internal Guide Head of the Department**

**Dr.A.Sharada Dr.M.Seetha**

**Professor Professor and Head**

**Department of CSE Department of CSE**

**External Examiner**

**Acknowledgements**

We would like to express our sincere thanks to **Dr K. Ramesh Reddy, Principal**, GNITS, for providing the working facilities in the college.

Our sincere thanks and gratitude to **Dr. M.Seetha, Professor and HOD**, Dept. of CSE, GNITS for all the timely support and valuable suggestions during the period of our mini project.

We are extremely thankful to **Mrs. Ch. Mandakini, Asst. Professor, Mrs. T. Swapna, Asst. Professor, Mrs. B. Geetha Kumari, Asst. Professor,** Dept. of CSE, GNITS for their encouragement and support throughout the project.

We are extremely thankful and indebted to our internal guide, **Dr.A.Sharada, Professor ,** Dept. of CSE, GNITSforher constant guidance, continuous advice, encouragement and moral support throughout the mini project.

Finally, we would also like to thank all the faculty and staff of CSE Department who helped us directly or indirectly, parents and friends for their cooperation in completing the mini project work.

**P Gayatri Varma (17251A0508)**

**K Venkata Srihitha (17251A0542)**

**Thota Sudharani (18255A0506)**

**ABSTRACT**

In developing countries like India, Farmers take loans for farming and harvesting. Every harvested crop needs to be transported, either directly from the field to the market, or to the packing house and storage. Transport is a burning component of post-harvest crop management. If a farmer spends more money in the transportation, he might end up in debts. Currently there are no such apps which connects various farmers so that they can split the transportation cost. If a farmer has space left to accommodate the harvest of another farmer in the same vehicle he wants to transport, then both of them can share the vehicle and save the money. The price differs in different places and some farmers can’t afford the transportation charges.

To overcome the problems the framer who is willing to share the vehicle can make use of this app. The proposed app does this by pooling and sharing the transport facility. In this app, a farmer can create a ride, find a ride. To create a ride, farmer must give the details of harvest like no of bags, time of ride, so that the remaining space of the vehicle could be efficiently used. To find a ride, farmer must give information about the source and destination so that the list can be displayed. Using these features, farmers can make use of transport vehicle efficiently.

**Table of Contents**

Abstract…………………………………………………………………………………………………………… iv

[1.INTRODUCTION 1](#_Toc64017169)

[**1.1** **Existing system** 1](#_Toc64017170)

[**1.2** **Advantages and disadvantages** 1](#_Toc64017171)

[**1.3** **Proposed system** 2](#_Toc64017172)

[**1.4** **Objectives** 2](#_Toc64017173)

[**1.5** **Methodology** 2](#_Toc64017174)

[**1.6** **Software requirements** 3](#_Toc64017175)

[**1.7** **Hardware requirements** 3](#_Toc64017176)

[**1.8** **Organization of the project** 3](#_Toc64017177)

[2. DESIGN AND ARCHITECTURE 4](#_Toc64017178)

[**2.1** **Architecture** 4](#_Toc64017179)

[**2.2** **Module Description** 5](#_Toc64017180)

[3. IMPLEMENTATION 7](#_Toc64017181)

[**3.1** **Technologies used** 7](#_Toc64017182)

[**3.1.1** **Android** 7](#_Toc64017183)

[**3.1.2** **JAVA** 8](#_Toc64017184)

[**3.1.3** **SQLITE** 10](#_Toc64017185)

[**3.2** **Working of the project** 10](#_Toc64017186)

[**3.3** **Database Implementation** 12](#_Toc64017187)

[**3.4** **UML Diagrams**](#_Toc64017188) 15

[**3.4.1** **UML DIAGRAMS FOR YIELD DRIVE APPLICATIONS**](#_Toc64017189) 16

[4. RESULTS AND DISCUSSIONS 20](#_Toc64017190)

[**4.1** **Results of the modules implementation** 20](#_Toc64017191)

[**4.2** **Screenshots of project implementation** 21](#_Toc64017192)

[6. CONCLUSIONS AND FUTURE ENHANCEMENTS 29](#_Toc64017193)

[REFERENCES 30](#_Toc64017194)

[GLOSSARY 31](#_Toc64017195)

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig.No** | **Description** | **Page No** |

Fig 2.2.1 System Architecture 4

Fig 3.3.1 Farmer registration table 12

Fig 3.3.2 Transport registration table 13

Fig 3.3.3 Vehicle booking table 14

Fig 3.3.4 Vehicle registration table 15

Fig 3.4.1 Use case diagram for farmer 16

Fig 3.4.2 Use case diagram for transport provider 17

Fig 3.4.3 Activity diagram for farmer login 18

Fig 3.4.4 Activity diagram for transporter login 19

Fig 4.2.1 Flash screen 21

Fig 4.2.2 Home page 22

Fig 4.2.3 Farmers Sign up 23

Fig 4.2.4 Transport provider sign up 24

Fig 4.2.5 Farmers home page 25

Fig 4.2.6 Vehicle booking 26

Fig 4.2.7 Vendors Homepage 27

Fig 4.2.8 Vehicle registration 28

# **INTRODUCTION**

Agriculture plays an important role in Indian economy. Over 70 percent of the rural households take up agriculture as their source of income. Post-harvest management is one of the phases of farming where a farmer spends more money before the yield is sold. This phase mainly includes transportation and storage of the yield. For example, if the amount of yield produced is less and the crop yield is to be transported for larger distances, then amount of profit is relatively less.

Every farmer needs to hire his own vehicle for transportation irrespective of the quantity of the yield produced. When a farmer needs to hire a vehicle, its pricing depends on the various factors like geographic location of farm. In some cases, the farmer cannot afford for transportation also. When each farmer uses a separate vehicle, the efficiency of the vehicle is also not completely utilized. So, to increase the efficiency of vehicle and reduce the transportation cost, vehicle pooling can be adopted. The current project is an android mobile application that implements this feature.

## **Existing system**

According to the existing system, a farmer may use various modes of transport depending on his affordability and necessity. When every farmer hires a separate vehicle for him, average cost of transportation is increased.

## **Advantages and disadvantages**

If the amount of crop yield is less, he can choose a smaller transportation vehicle like auto. When the amount of crop yield is high, he must look for larger sources of transportation. When it is not available, he might have to get compromised with the ones available at that point of time. In some cases, transportation vehicles might not be available due to various reasons. During these situations, he cannot transport his crop yield.

## **Proposed system**

The proposed system helps farmers in transportation pooling where a single transportation vehicle can be used by multiple farmers at the same time to transport their crop yield. Here, yield of multiple farmers can be transported within the cost of one farmer. So, the amount spent by farmers on transportation depends on the quantity of crop yield being transported. By transportation pooling method, the space in the vehicle can also be utilized efficiently.

## **Objectives**

The main objective of this project is to develop an android app for pooling the transport of harvested crop. By doing so, the efficiency of the vehicle used for transportation can be increased. By transportation pooling, the average cost of transportation spent by farmer on the yield can be significantly reduced, leading to increase in the profits.

* Develop an android app for pooling the transport of harvested crop
* To utilise the capacity of vehicle efficiently
* To save transportation cost of yield
* Transportation pooling based on location will be implemented

## **Methodology**

Most farmers give more priority to transportation as the selling price of the crop depends on the condition of the crop. Post-harvest management is one of the most important phase of farming which when managed efficiently helps in significant increase of profits. Post-harvest management includes both transportation and storage of the crop yield. In this project we made an attempt to reduce the transport expenditure.

Vehicle pooling is one of the ways to reduce transportation expenditure. In this mobile application, more than one farmer can make use of a single transportation vehicle at the same time. To implement this, every farmer who want to use vehicle pooling and the vendor providing the transport vehicles need to sign up. Later the vendor need to register his vehicle providing all the necessary details required by the farmer. Now, a farmer can book a vehicle depending on his requirement.

1. **Software requirements**

* Android Studio
* Android Software Development Kit (SDK)
* Operating System: Mac OS X, Windows 7, Windows 10

## **Hardware requirements**

1. PROCESSOR : (min) Intel core i3processor

2. RAM : 1GB

3. Hard Disk : 20 GB

## **Organization of the project**

There are various chapters in the documentation of this project describing necessary elements involved. It first begins with abstract in which the overview of the project and the various fields involved in it are described in short. The chapter introduction has seven topics it talks about existing system its advantages and disadvantages. And further it discusses about the proposed system, objectives describing what will be implemented, methodology, requirement gathering ,general software and hardware requirements.

The second chapter – system architecture gives the architecture which is the diagram or structure of the project, module description- the different types of modules involved in detail and the packages used for implementation of the code.

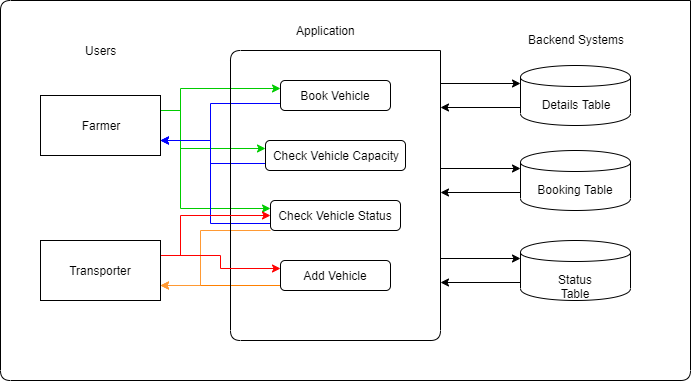
The next chapter implementation includes the description of technologies used. The front end, backend and the database used is described in detail. The UML diagrams such as use case diagrams, activity diagram and sequence diagram are drawn.

In the chapter results, the detailed discussion on the results is made along with the screenshots of each screen in the app. In the last chapter, conclusions and scope for future enhancements, the scope of the project is discussed in detail.

# **2. SYSTEM ARCHITECTURE**

## **Architecture**

The main advantage of the project “Yield Drive” is reducing the cost of transporting the agriculture products. In this application, a farmer can easily book the vehicle at lower cost. This application provides the communication between farmer and transport department, based on the vehicle capacity they can book the vehicle and remaining capacity will be confirmed by another farmer.



# **Fig 2.1.1** : System Architecture

After identifying all the behaviors of the application, the structure of the program has been constructed, where the key parts of the software program have been initialized. The structure of the application mainly has three parts. The idea of the first part is to allow interaction between

the user and the application. The second part is invisible to the user.The last part is responsible to store the details and actions of the user.

There are two users in the project. They are farmers and transport vendors. The above architecture diagram is representing about the users application and backend systems.

To access the features provided the users must sign up to the application and access them.

A farmer can check for available vehicle, book a vehicle and also check capacity status of the vehicle. A transport vendor can add a vehicle, check the bookings and also check capacity status of the vehicle.

When a farmer raises a request to check for available vehicles, the system accesses the vehicle details table and displays the list of vehicles that are available for the farmer after filtering according to requirement. When a farmer raises request to book a vehicle, the system checks if all the necessary conditions like required capacity is less than available capacity, are fulfilled, then the vehicle is booked. After the booking is made, it is updated in booking table. When a farmer raised the request to check the capacity status of vehicle, the required information is retrieved from the booking table.

When a transport vendor raises the request to add a vehicle for transportation pooling, the system updates the vehicle details table. When the transport vendor raises the request to check the bookings, the system retrieves this information from the booking table. When the transport vendor raises the request to check the capacity status of vehicle, the required information is retrieved from the booking table.

## **Module Description**

In this application there are two different modules.

1. **Transportation Vendor Module:**

* In this module, transportation vendor can register by providing his details like user id, password, mobile number and email id.
* Transportation vendor can gain access to the app by logging in using his credentials.
* After successful login he can register a vehicle for transportation pooling by providing information about the vehicle like vehicle number, driver name, type of crop to be transported in the vehicle, maximum capacity of the vehicle.
* This vehicle information is shared with farmers for booking the vehicle.

1. **Farmer Module:**

* In this module, farmers can register by providing his details like user id, password, mobile number and email id.
* After registration, farmer can login to use the transport pooling feature. For this, the farmer need to book a vehicle.
* To book a vehicle, farmer can select a vehicle of his choice depending on his requirements like type of crop, required capacity of vehicle.
* When the booking is made, details of farmer are conveyed to transport vendor about the booking date, time, source, destination.

# **3. IMPLEMENTATION**

## **Technologies used**

To develop the android application “Yield drive”, app development technology is used. Android Studio is the application used to develop this project. Technologies used to create the project in android studio are java, xml, SQLite.

#### **Android**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

The following features are provided in the current stable version:

• Gradle-based build support

• Android-specific refactoring and quick fixes

• Pro-Guard integration and app-signing capabilities

• Template-based wizards to create common Android designs and components

• A rich layout editor that allows users to drag-and-drop UI components, option

to preview layouts on multiple screen configurations

• Support for building Android Wear apps

• Android Virtual Device (Emulator) to run and debug apps in the Android studio.

#### **JAVA**

Java is an object-oriented language and is very similar to C++. Java is simplified to eliminate language features that cause common programming errors. Java source code files are compiled into a format called bytecode, which can then be executed by a Java interpreter. Java is used for the backend code development. Important features of java include platform independent, simplicity, robust, portability, object oriented and integration. Due to these features, we can integrate both java and xml codes to do the android application.

**Home.java**

In this module, there are two text boxes available for users to enter their respective user id and passwords. If the credentials entered doesn’t match any of the user, neither farmer nor transport vendor, a login failed error is displayed. If the user is using the app for first time, he need to register. He can login with the credentials only after successful registration. If the user registers as farmer, new row is added to “freg” table and if the user registers as transportation vendor, new row is added to “treg” table.

**Freg.java**

This is used to register the farmer by checking for his details in the “freg” table. If there is no existing farmer, a new row is created in the table with all the details entered. Next time, when a farmer tries to login, user id and password are retrieved from this and checked. User is given access to his account only if the login credentials entered is correct.

**Treg.java**

This is used to register the transportation vendor by checking for his details in the “treg” table. If there is no existing vendor, a new row is created in the table with all the details entered. Next time, when a vendor tries to login, user id and password are retrieved from this and checked. User is given access to his account only if the login credentials entered is correct. After gaining access, he can register his vehicles that are available for transportation.

**Fbookingconf.java**

In this module, farmer checks the available vehicles and accesses the vehicle he requires based on his requirements like capacity, date, time, type of crop. To do the booking, a mapping is done between the farmer and vehicle. After the booking is done, the capacity status of the vehicle is updated in the “vbooking” table of database.

**Fbookinginformation.java**

In this module, all the booking made by that farmer are displayed. For this, details of all the bookings are retrieved using the user id from “vbooking” table.

**Taddvehicle.java**

In this module, transportation vendor can add the vehicles that are available for transportation. After adding a vehicle, it is added as new row in the “vreg1” table of database.

**Tbookinginformation.java**

In this module, the transportation vendor can see all the bookings made. These bookings are retrieved from farmer database where user id is same as vendor id.

**Fhome.java**

In this module, the farmer can see the activities that he can perform, like vehicle booking, booking information, farmer profile and logout. Back button is disabled. So, farmer need to logout from his current account when he wants to sign in again.

**Thome.java**

In this module, the transport vendor can see the activities he can perform, like vehicle registration, booking information, driver/transport profile and logout.

#### **SQLITE**

SQLite implements most of the SQL-92 standard for SQL, but it lacks some features. A standalone program called sqlite3 is provided to create a database, define tables within the database, insert and change rows, run queries and manage an SQLite database file. The source code for SQLite is in the public domain.

SQLite is a popular choice for local SQL storage within a web browser and within a rich internet application framework. This may be because SQLite's dynamically typed storage matches the web browser's core languages of JavaScript and XML. SQLite uses an unusual type system for an SQL-compatible DBMS. Instead of assigning a type to a column as in most SQL database systems, types are assigned to individual values; in language terms it is dynamically typed.

## 

## **Working of the project**

The current project “Yield Drive” helps farmers reduce cost of transportation by adopting a method called transportation pooling. Transportation pooling is using same transportation by different users based on their time and place of travel. By using transportation pooling in post-harvest management, more than one farmer can use the same vehicle at same time to transport their yield produced. By doing so, transportation vehicle can be utilized efficiently. Thereby, the average cost spent by each farmer over transportation can be reduced. We mainly have 2 users for this application. They are farmer and transportation vendor.

**Farmer as user:**

Initially, a farmer needs to register himself providing his details like name, email Id, phone number, password. When a farmer registers, his details are stored under “freg” table. This table can be accessed to retrieve information about farmers.

After registering, a farmer needs to login to book the vehicle. Under Vehicle booking tab, all the vehicles available for the booking are displayed. He can select the vehicle based on his requirement. He can search for the vehicle based on the crop he needs to transport and then select the vehicle. After selecting the vehicle, he can book the vehicle by giving his details like source, destination, date and time of transport, required capacity of the vehicle. If the required capacity is less than or equal to the available capacity of the vehicle, the booking is made.

After the booking is made, all these details will be updated in the “vbooking” table along with the current status of the vehicle capacity. When the required capacity is higher than the available capacity of the vehicle, the booking is not made and requests you to reduce the required capacity. At this point, the farmer can either change the capacity or even change the vehicle depending on his choice.

After the booking is successfully done, the details of the vehicle can be viewed under “booking information” tab. Details of vehicle include vehicle number, driver name, driver number.

Details of the farmer can be viewed under “Farmer profile” tab. This page displays all the information about farmer like his name, email, password, phone number. On clicking logout button, the farmer logs out from his current account.

**Transportation Vendor as user:**

A transportation vendor is the person who provides the vehicles to farmers. Initially he needs to register himself to get the authentication to the app. After registering, transport vendor details are updated to the “treg” table.

After registering, he needs to register each of his vehicle which can be used to transport by giving details like vehicle number, driver name, driver number, type of crop to be transported, maximum capacity that vehicle can accommodate. After registering, these details are stored into “vreg1” table.

Whenever a booking is made by the farmer, all the booking details like Vehicle number, Driver name, Mobile number, total vehicle capacity, Source, Destination, Date, time, capacity booked by farmer are retrieved from “vbooking” table and displayed.

All the details of the vehicle vendor are displayed under Driver/Transport Profile tab. The details like Driver id, email, mobile number are retrieved from “treg” and displayed.

## **Database Implementation**

Whenever the android app is launched for the first time, a database named yield is created. As we use the app further, the tables are updated according to the modules that are being implemented. The database yield mainly consists of four tables namely,

freg, treg, vbooking, vreg1

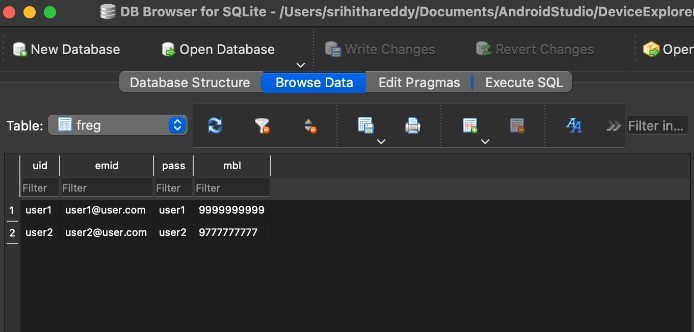
1. **freg table :**

The table freg is initially created when the database is created. This table consists of four columns – uid, emid, pass, mbl. When a farmer registers using farmer registration page, all the rows are checked if any other farmer has same details. If there is no such farmer, a new row is added to the table with the details as follows

**Farmer’s id – uid, Farmer’s email – emid, Farmer’s Password – pass, Farmer’s mobile number – mbl**

Where email id and uid are unique keys. Farmers id is the primary key.

Whenever a farmer tries to login, the freg table is checked for matching emid and pass pair. If any such pair exists, then the farmer is given access to his account so that he can book vehicles, know his recent bookings.



**Fig 3.3.1:** Farmer registration table

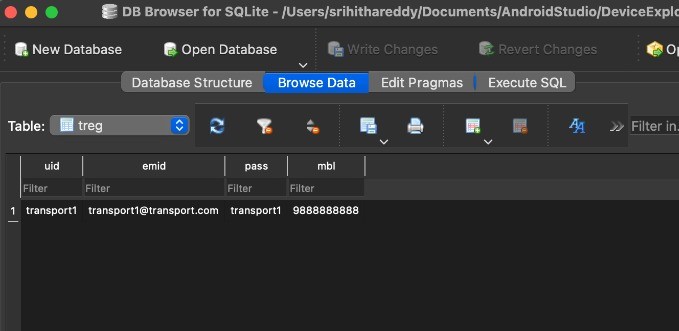
1. **treg table :**

The table treg is initially created when the database is created. This table consists of four columns – uid, emid, pass, mbl. When a transportation vendor registers using transport registration page, all the rows are checked if any other transport vendor has same details. If there is no such vendor, a new row is added to the table with the details as follows

**Vendor’s id – uid, Vendor’s email – emid, Vendor’s Password – pass, Vendor’s mobile number – mbl**

Where uid is primary key.

Whenever a vendor tries to login, the treg table is checked for matching emid and pass pair. If any such pair exists, transport vendor is given access to register his vehicles, get access to database to know the list of bookings made.

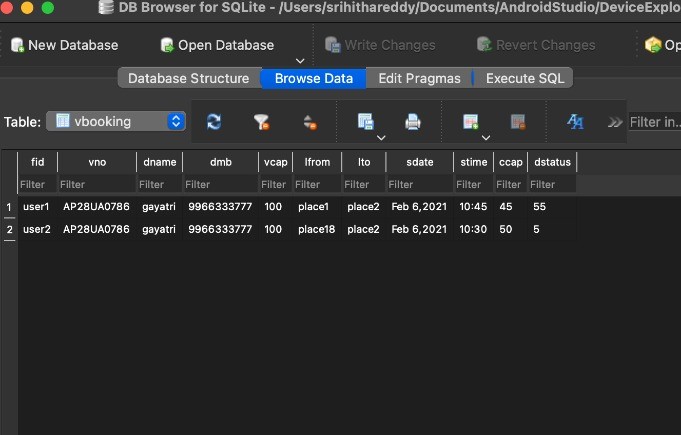


**Fig 3.3.2:** Transport registration table

1. **vbooking:**

The table vbooking is initially created when the database is created. This table consists of eleven columns – **farmer’s id fid, vehicle number vno, drive name dname, driver mobile number dmb, maximum vehicle capacity vcap, source location lfrom, destination location lto, date of transport sdate, time of transport stime, capacity booked ccap, available capacity dstatus**. Where farmer id is primary key, vehicle no is unique. When a farmer books a vehicle, his details along with the status of capacity are updated into this table. Based on the requirement, details can be filtered and retrieved.

When a farmer logs into his account and selects booking information tab, data is retrieved from vbooking table where user id is same as farmer id. Whenever another farmer wants to book, the ccap value is of particular vehicle is retrieved where vehicle number is same.



**Fig 3.3.3:**  Vehicle Booking Table

1. **vreg1 :**

The table vreg1 is initially created when the database is created. This table consists of 5 columns – **vehicle number vno, driver name dname, driver mobile number dmb, maximum vehicle capacity vcap, transport crop name cnamet**. Where vehicle no is unique key. When a vehicle is registered by authenticated vendor, a new row is added to this table. When a farmer wants to book a vehicle, all the vehicles available are displayed from this table. When a farmer searches for a particular crop, details of vehicles with that same cnamet are retrieved and displayed.



**Fig 3.3.4 :** Vehicle registration table

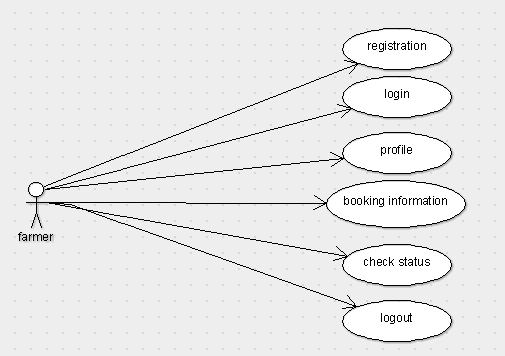
1. **UML Diagrams**

The unified modeling is a standard language for specifying, visualizing, constructing and documenting the system and its components is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure and control information about the systems.

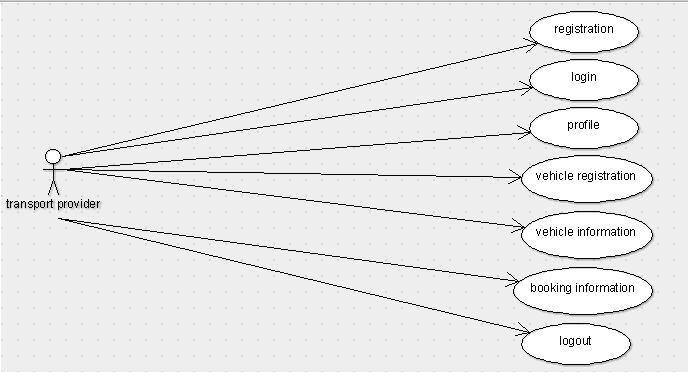
The UML addresses the documentation of a system's architecture and all of its details. The UML also provides a language for expressing requirements and for tests. Finally, the UML provides a language for modeling the activities of project planning and release management.

**3.4.1 UML DIAGRAMS FOR YIELD DRIVE APPLICATIONS**

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a app. The "actors" are farmers and transport provider.

****

**Fig 3.4.1 :** Use case diagram for farmer

****

**Fig 3.4.2:** Use case diagram for transport provider

Above figure represents the use case diagram of the current project Yield drive.

Farmer who uses the app for the first time need to register. After registering, he can login using his login credentials. He can access his profile to know the details he gave. Farmer can get the vehicle information, so that he can know the various vehicles available for him to book. He can also check information of his bookings and even status of the capacity of vehicle. He can also logout.

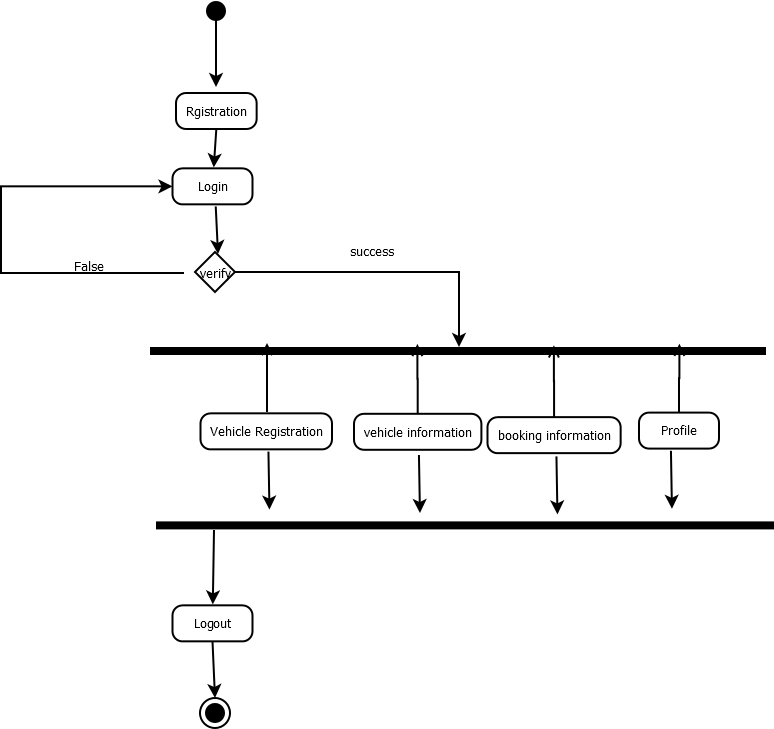
Transportation vendor who uses the app for first time should register. After registering, he can login using his credentials. After logging in, he can register his vehicles, view his profile. A transport vendor can also receive information of all the information related to booking. He can even check the status of the capacity of the vehicle. He can logout.

**ACTIVITY DIAGRAM**

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram . Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram . Activities modeled can be sequential and concurrent. In both cases an activity diagram will have a beginning (an initial state) and an end (a final state).

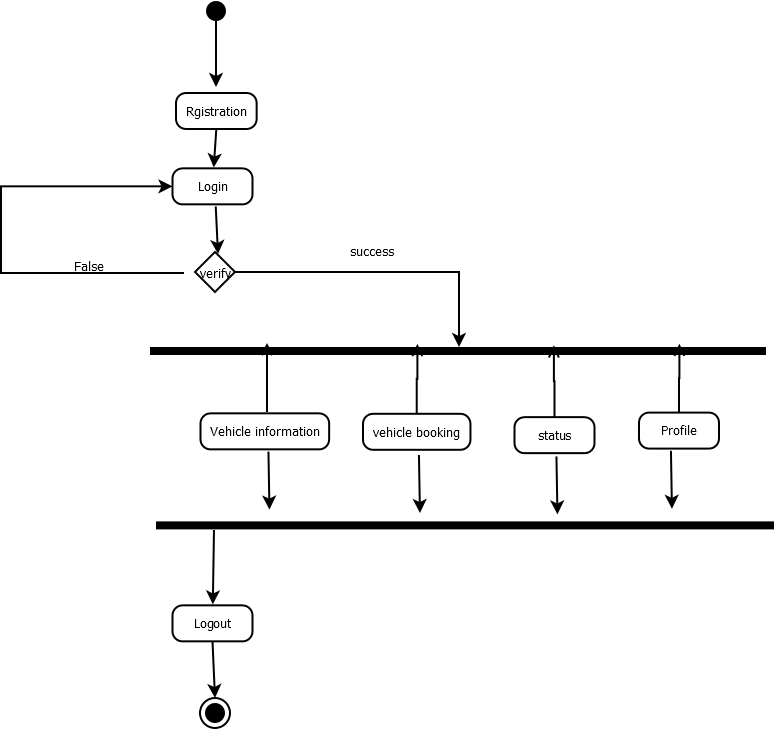
There are two activities respective to the user ,they are:

* Farmer login
* Transporter login



**Fig 3.4.3:** Activity diagram for transportation vendor login

Above figure represents the activity diagram of the transportation vendor login. After registering, a vendor can now login using his credentials. If the login credentials are verified, he is taken to home page, where there are four tabs Vehicle Registration, Vehicle Information, Booking Information, Profile. He can directly logout by clicking logout button.

****

Profile

**Fig 3.4.4 :** Activity diagram of Farmer login

Above figure represents the activity diagram of the farmer login. After registering, a farmer can now login using his credentials. If the login credentials are verified, he is taken to home page, where there are four tabs Vehicle Information, Vehicle Booking, Profile. He can directly logout by clicking logout button.

# **4. RESULTS AND DISCUSSIONS**

## **Results of the modules implementation**

In this present situation transportation is the main problem to the farmers to sell their agriculture goods. To overcome this problem we are introducing this project, yield drive. Yield drive is a mobile application based on android environment.

In present situation communication between farmers and transportation vendors is one of the major issues .To overcome this, Yield drive mobile application acts as an interface between the famers and transportation vendors. In this application the main users are transportation vendor and farmer. Each module has their own roles and responsibilities.

In transportation vendor module first they have to register with their details and can add the vehicle information. This stores in SQLite database. They also have a chance to verify their vehicle information. Based on this data, farmers can get this information from database. Based on farmer’s requirement, he can select the transport vehicle depending on time, date, type of crop and capacity. After a vehicle is booked, this information is updated in the profiles of respective farmer and the transportation vendor.

We have tested the system with various testcases, where the system was able to give the

correct outputs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.no | Input | Expected output | Actual Output | Result |
| 1 | If the user is not registered | Not valid details | Not valid details | Pass |
| 2 | If username is given wrong | Login failed | Login failed | Pass |
| 3 | If the email is not in correct format | Wrong email | Wrong email | Pass |
| 4 | If password is given wrong | Login failed | Login failed | Pass |
| 5 | If we want to create a new ride for transportation | Ride created successfully | Ride created successfully | Pass |
| 6 | If you want to book a ride | Ride booked successfully | Ride booked successfully | Pass |
| 7 | If the weight of crop is high | Cart overloaded | Cart overloaded | Pass |

1. **Screenshots of project implementation**

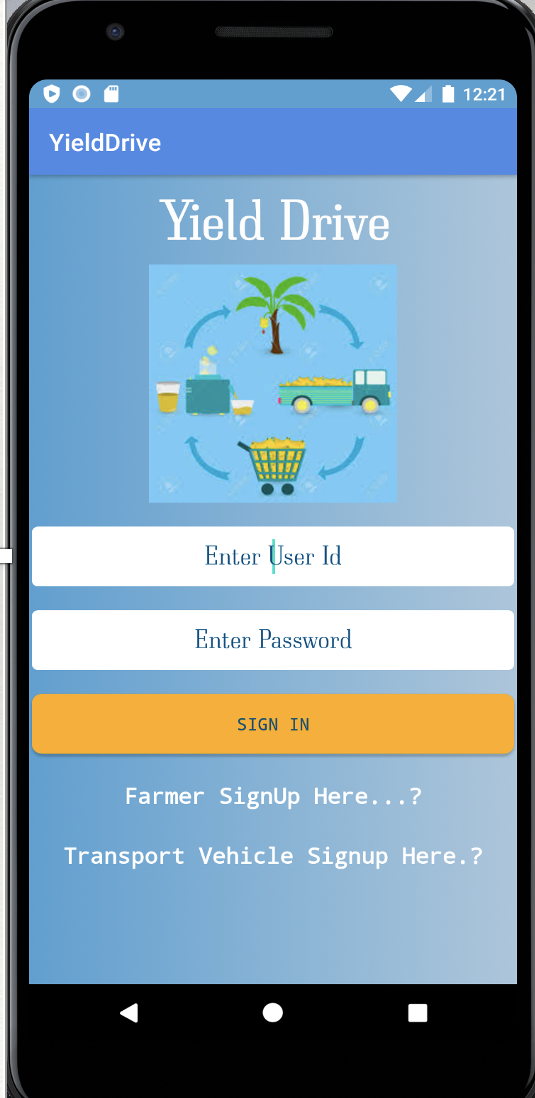
**Flash Screen:**

****

**Fig 4.2.1 :** Flash screen

Above figure is the screen shot of flash screen that appears when the android app is launched.

**Home Page:**

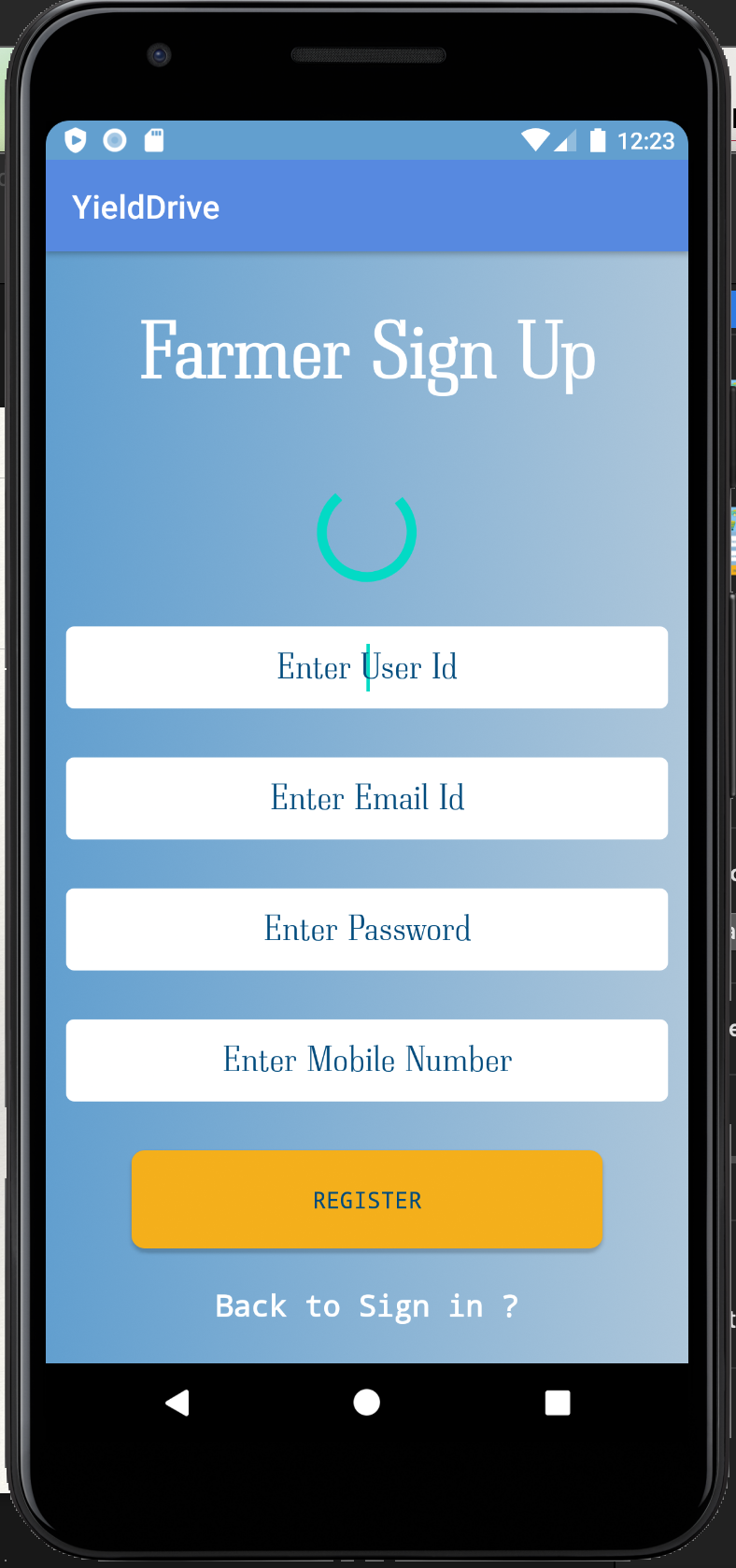


**Fig 4.2.2:** Home page

Above screen shot shows the Home page of the android application Yield Drive. If the user is using the app for first time, he needs to register himself by clicking his respective signup module.

If the user has already signed up, he can login by giving his correct credentials in the text boxes provided.

**Farmer Signup page:**

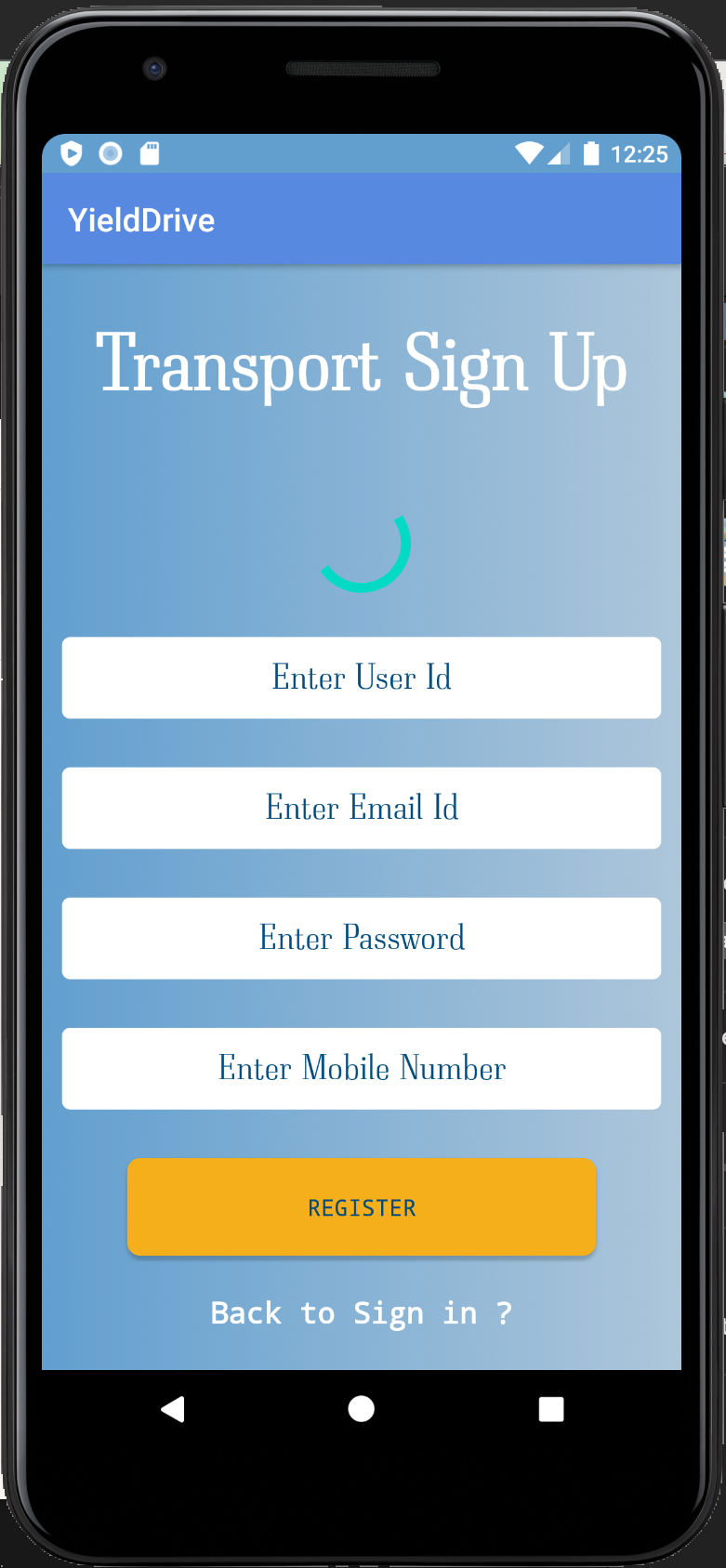


**Fig 4.2.3 :** Farmers Sign up

Above screen shot shows the farmer signup page. When the user is farmer and is using application for the first time, he need to signup here by providing his details like user id, email id, password, mobile number.

If the farmer is already registered he can click on Sign in.

**Transport Signup page:**

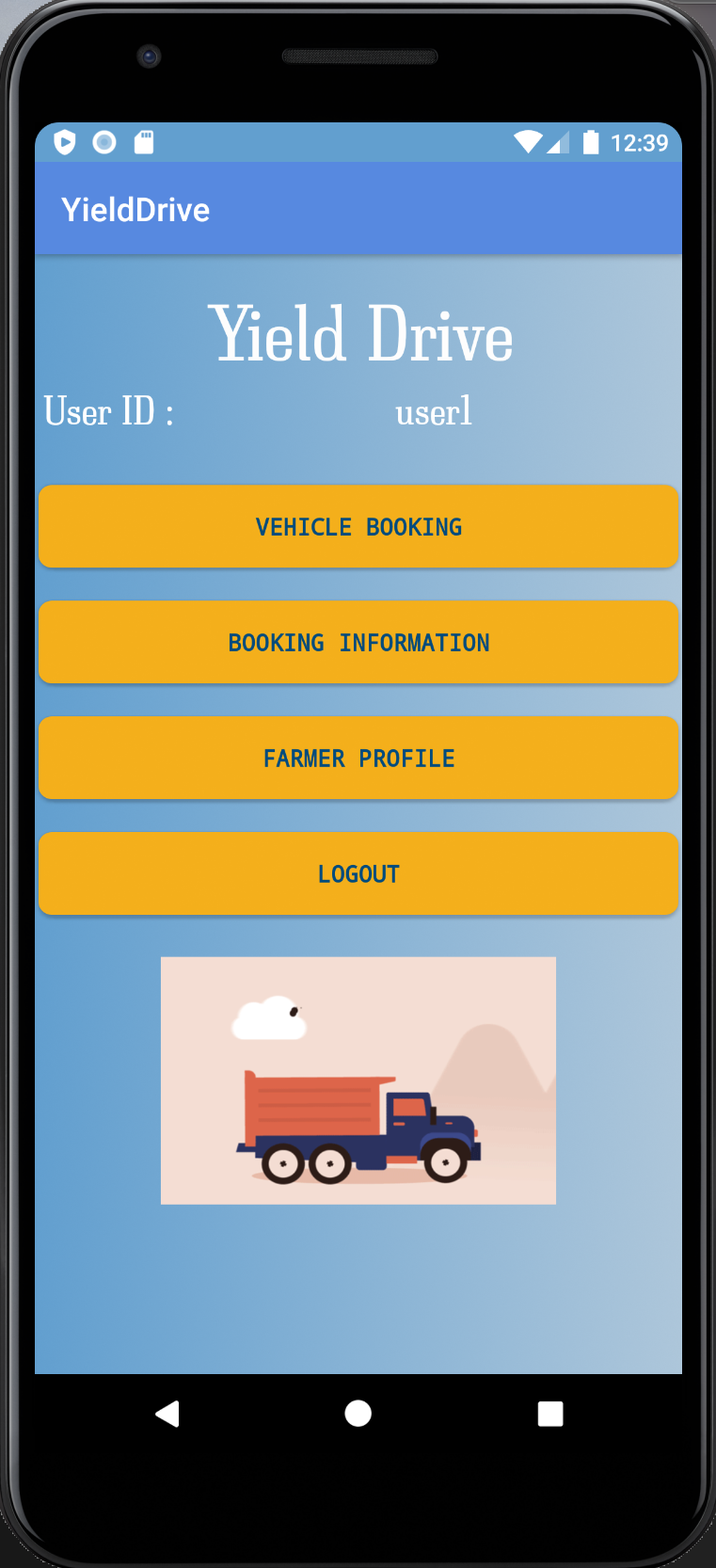


**Fig 4.2.4:** Transport provider Sign up

Above screen shot shows the transport vendor signup page. When the user is a vendor and is using application for the first time, he need to signup here by providing his details like user id, email id, password, mobile number.

If the vendor is already registered he can click on Sign in.

**Farmer Home Page:**



**Fig 4.2.5 :** Farmers Home page

Above screen shot shows the farmer home page after successful login of the farmer. There are four tabs – vehicle booking, booking information, farmer profile, logout

**Farmer Vehicle booking Page:**

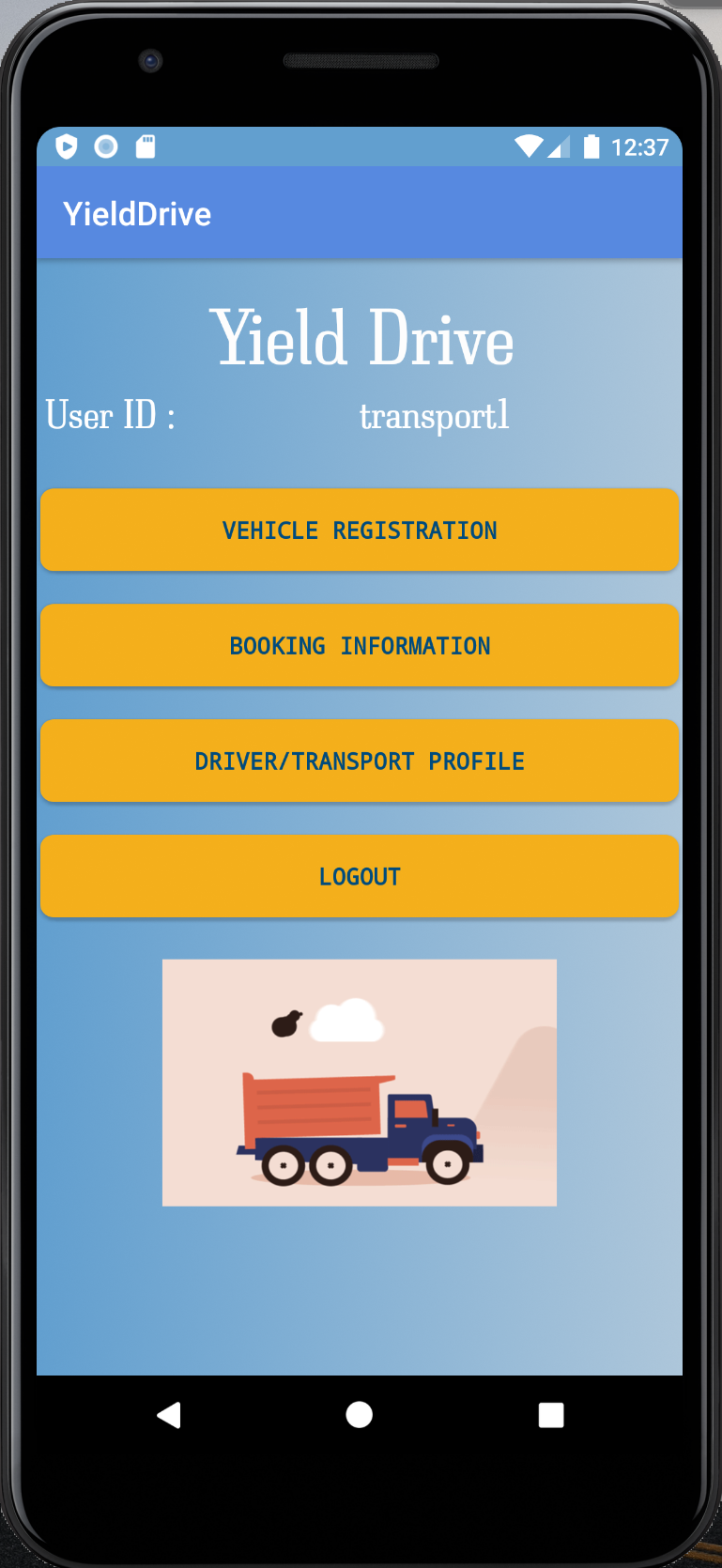
A screen shot of a cell phone

Description automatically generated with low confidence

**Fig 4.2.6 :** Vehicle booking

Above screen shot is the vehicle booking page of farmer. A farmer can search the vehicles depending on the crop he wants to transport.

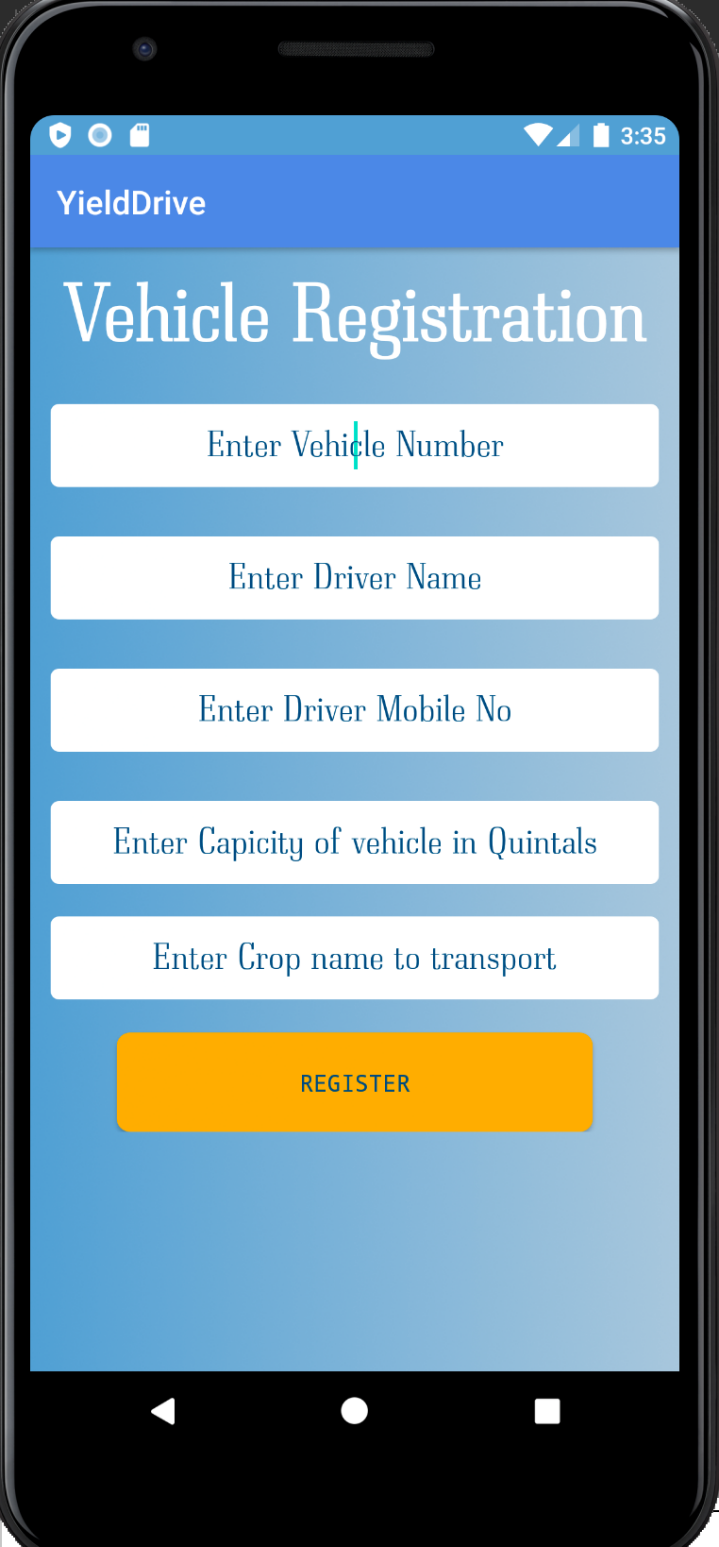
**Transport Vendor Home Page:**



**Fig 4.2.7:** vendors home page

Above screen shot shows the transport vendor home page after successful login of the vendor. There are four tabs – vehicle registration, booking information, driver/transport profile, logout

**Transport vendor Vehicle registration Page:**



**Fig 4.2.8 :** Vehicle Registration

Above screen shot is the vehicle registration page of the transportation vendor. To register a vehicle to be available for transportation pooling, the vendor need to provide details like vehicle number, driver name, driver mobile number, maximum capacity of the vehicle, name of the crop to transport.

# **6. CONCLUSIONS AND FUTURE ENHANCEMENTS**

**CONCLUSION**

At the end of this application, At the end of this application, it is a sophisticated approach where the farmer can transport his yield at low cost. It gives better performance for transport provider so that he can easily add the vehicle information. They can view and get the bookings from farmers. This initiative of complaining online made easy for farmers to book the transport vehicle. This is even convenient to transport and farmers to view and monitor the process. There is also a facility for the user to check the status of booking vehicle.

**FUTURE ENHANCEMENTS**

In future enhancement we can add the GPS system in this application then farmers get the live location. We can also add the billing system into this application.

# **REFERENCES**

1. <https://www.researchgate.net/publication/261570852_Transporting_the_yield_Appropriate_transport_for_agricultural_production_and_marketing_in_Sub-Saharan_Africa>
2. <https://blog.agrivi.com/post/transport-as-an-important-factor-for-a-farmer-s-success>
3. <https://agricultureandfoodsecurity.biomedcentral.com/articles/10.1186/2048-7010-1-19>
4. 230555082\_The\_impact\_of\_transportation\_on\_agricultural\_production\_in\_a\_developing\_country\_a\_case\_of\_kolanut\_production\_in\_Nigeria

**WEBSITES:**  **REFERRED URLS:**

www.android.com http://developer.android.com/index.html

www.google.com <http://en.wikipedia.org/wiki/SQLite>

# **GLOSSARY**

**XML-** Extensible markup language

**SDK-** Software development kit

**NDK-** Native development Kit

**AVD-** Android Virtual device manager

**OOPS-** Object oriented programing