CHAPTER 1

INTRODUCTION

- This Mobile commerce offers many benefits to both consumers and businesses. For consumers, it provides a convenient and easy way to shop online, them to browse and purchase products from anywhere at any time.
- For businesses, it opens up new opportunities to reach customers and increase sales, as well as providing valuable insights into customer behaviour and preferences. Mobile commerce can take many forms, including mobile websites, mobile apps, and mobile payments.
- Mobile websites are optimized for viewing on mobile devices and allow users to shop online using their mobile browser. Mobile apps are dedicated applications that can be downloaded and installed on mobile devices, providing a more seamless shopping experience.
- Mobile payments enable users to make purchases using their mobile devices, often through digital wallets or mobile payment platforms.
- Firebase is a cloud-based platform for developing mobile and web applications. Firebase simplifies the development process by providing easy-to-use tools and services that allow developers to focus on building their application's features rather than managing infrastructure.
- The Existing system contains agents between buyer and seller . Also, the existing system results in downturn in economy of retail shops as well as middle level family. This existing system makes maximum profit to the third party agent. This makes retail shoppers to become economically unstable.
- The Proposed System will make profit to the both buyer and seller as well as to make retail shop and middle level family to become economically stable.
- Chat box will be between the buyer and seller for making better interaction. This also makes profit to the retail shops and also additional to that rental system is included in this system which makes to reduce the useless product.

1.1 PROBLEM DEFINITION

- M-commerce is a rapidly growing trend that offers many benefits for both vendors and customers.
- In M-commerce Third Party Agent is making more Profit than the customer and vendor.
- How to overcome such Situation and how we can improve Rental System in M-commerce?

CHAPTER 2

LITERATURE REVIEW

The Kai Fan* et.al[1] explained about issues in authentication of mobile commerce. For enhancement in authentication in payment ,they came up with Secure Mutual Authentication Protocol (SMAP) which is based on the Universal 2nd Factor(U2F) protocol for mobile payment .This enhances security of user's account and improve payment through very less time consumption.

Ju Ouyang, and Xianping Chen[2] proposed a way to avoid adata leakage of customers in E-commerce .They introduced two dimensional code in encryption system for logistics service.They used QR code scanner for verification of details. This is the one-time scanner details that can't be used forsecond time.

Junyi He et.al[3] explained the existing system contains onlydata size. so,they explained about the parameters of data rate and temporal requirements. they classified the parameters on two basis, Homogeneous model and a Heterogeneous model. Both the auction satisfy the desired properties and it alsoshows efficiency of proposed system.

Falah Y H Ahmed ET.AL[4] developed an application in inMalaysia for Vehicle Rental System named as EZGO. Thisapplication allows the user to rent a car for their own purposes without the need to purchase and own for themselves. Using questionnaries, they performed a survey of perpectivecustomers and design and development of mobile apps, UML diagrams for the card rental system they used agile approach.

Eric Hseuh-Chan Lu and Zhan-Qing Lin[5] developed a Bicycle-Sharing System (BSS) which allows users to rent the bicycle from any Automatic rental station placed in the city. This research uses the concept of Recurrent Neural Network(RNN) to predict the rental from users.

Fumin Zhu ET.AL[6] developed a model to detect fault clicks on Pay per Click(PPC) dynamically and interpreting databased on Machine Learning. The proposed tens or transformation algorithm with locality-sensitive hashing(LSH)is tested by extensive experiments using real-world data.

CHAPTER 3

THEORETICAL BACKGROUND

3.1 IMPLIMENTATION ENVIRONMENT

The implementation environment involves the following:

Identification of retail shops: The first step in the implementation is the identification of retail shops. This is done through a process of survey and verification by the respective State Governments and Union Territory Administrations.

3.2 SYSTEM ARCHITECTURE

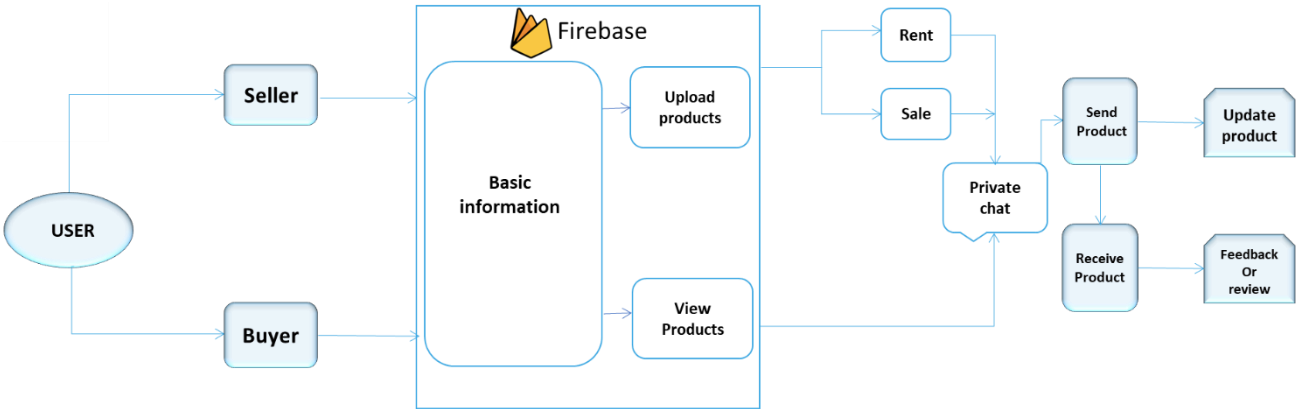


Fig3.1 .Architecture diagram for DIMEMERCE

In Fig 3.1 explains Customer needs to sign up and login in the app. after that he can able to select whether he is coming for buying, sell or renting the product. If Customer tries to buy or Getting product in rent, they can chat with seller directly to get additional information of product. If seller tries to sell the product, seller should upload product details along with photo and when product is sold. He/she should update the product quantity as they are stored in shop. If customer gets product, they will have an option of giving feedback to the product they bought. The feedback are later verified by admin.

3.3 PROPOSED METHODOLOGY

Data Set Description

This is normally represented as the data about data. It is also termed as metadata sometimes which gives the data about the data stored in the database. It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or the processes. Following are some rules, which defines the construction of data dictionary entries:

Words should be defined to understand for what they need and not the variable need by which they may be described in the program.

Each word must be unique. We cannot have two definitions of the same client.

Aliases or synonyms are allowed when two or more enters shows the same meaning. For example, a vendor number may also be called as customer number.10

A self-defining word should not be decomposed. It means that the reduction of any information in to subpart should be done only if it is really required that is it is not easy to understand directly.

3.3.1a Eligibility Checking Table:

COLUMN NAME	DATATYPE	DESCRIPTION	CONSTRAINT
USERNAME	VARCHAR(100)	NAME OF THEUSER	NOTNULL
PASSWORD	VARCHAR(100)	USERPASSWORD	NOTNULL
NAME	VARCHAR(100)	NAME OF THEUSER	NOTNULL
EMAIL	VARCHAR(100)	USEREMAILID	NOTNULL

Fig 3.3.1a Eligibility checking table for DIMEMERCE

3.3.1b Verification Table:

COLUMN NAME	DATATYPE	DESCRIPTION	CONSTRAINT
NAME	VARCHAR(25)	NAME OF THEUSER	NOTNULL
GOVT ID	BIGINT(12)	USERGOVT ID	NOTNULL
PHONENUMBER	BIGINT(20)	USER PHONENUMBER	NOTNULL
ADDRESS	TEXT	USER ADDRESS	NOT NULL

Fig 3.3.1b Verification table for DIMEMERCE

3.3.1 c Approval/ Reject Table:

COLUMN NAME	DATATYPE	DESCRIPTION	CONSTRAINT
APPLICATION_ID	VARCHAR(100)	USERAPPLICATIONID	NOTNULL
NAME	VARCHAR(100)	NAME OF THEUSER	NOTNULL
GOVT ID	BIGINT(20)	USERGOVT ID	NULL
SERVICE	VARCHAR(10)	USER SERVICE	NOTNULL

Fig 3.3.1c Approval/reject table for DIMEMERCE

3.3.2 Input Design

SOFTWARE REQUIREMENT

- Windows10& Above
- FLUTTER Framework
- GOOGLE Firebase
- API Integration
- VS code
- ML Integration

HARDWARE REQUIREMENT

- Processor: Minimum 1GHz
- Memory(RAM):4 GB
- Android or Ios
- Proper Network Connection

3.3.2Module Design

UML DIAGRAMS

3.3.3aUsecasediagram

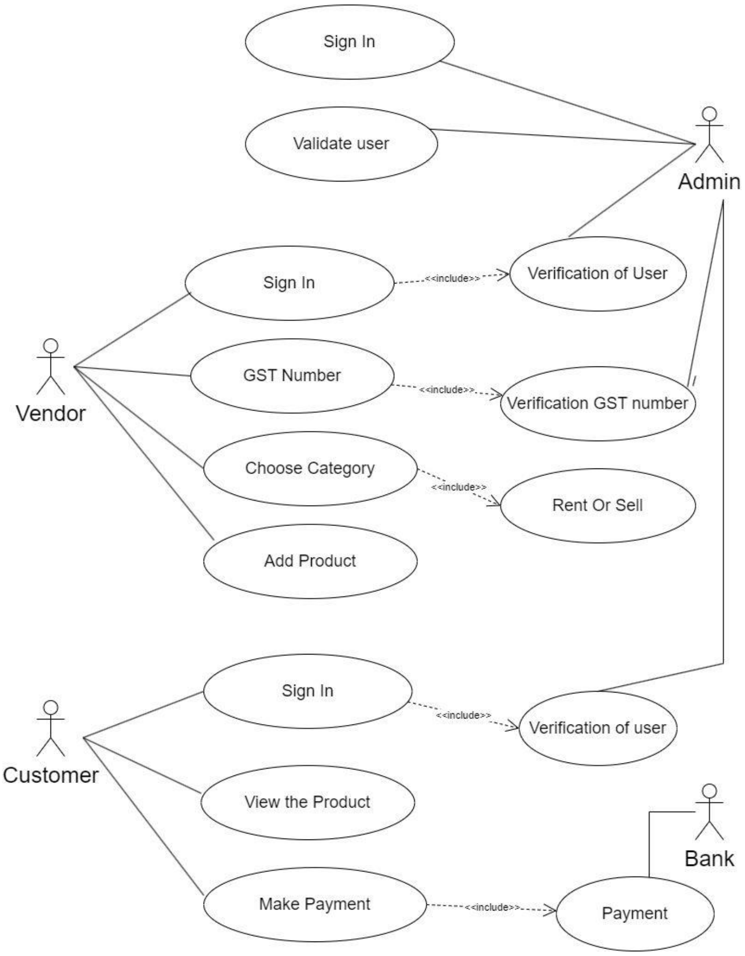


Fig3.3.3a Use case diagram for DIMEMERCE

This use case diagram refers to activities done by vendors and customer and their corresponding usecases. This use case diagram provides a high-level overview of the interactions between users and the DIMEMERCE system, outlining key functionalities and scenarios for both beneficiaries and administrators.

3.3.3b Class diagram

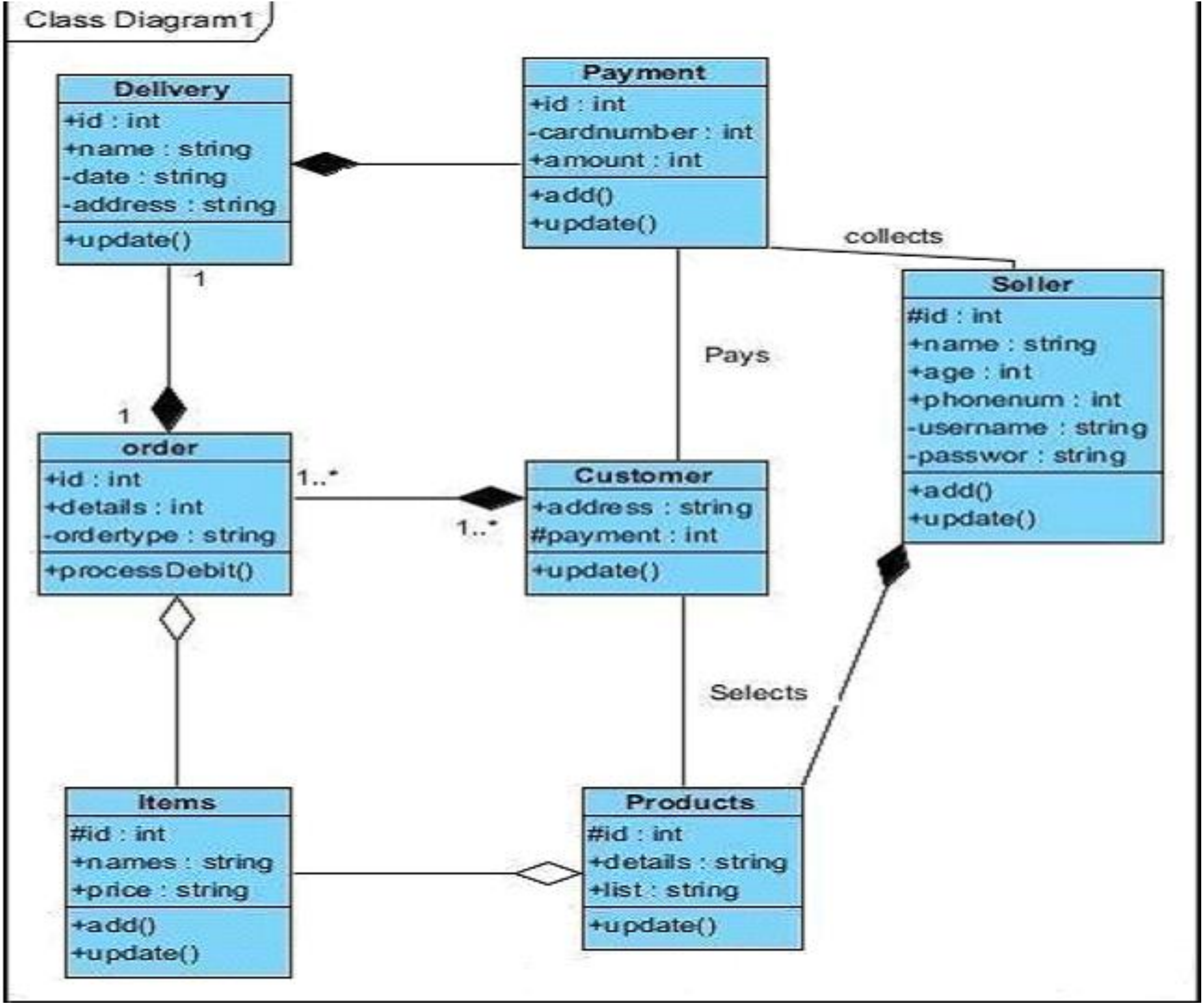


Fig3.3.3b Class diagram for DIMEMERCE

Creating a class diagram for the E-Commerce (DIMEMERCE) involves identifying the key classes and their relationships within the system. This use case diagram refers to activities done by technical ,support and service team and their corresponding use cases. In this class diagram, relationships such as aggregation and composition are not explicitly shown for simplicity. However, you can further refine the diagram by adding associations between classes and indicating the multiplicity of relationships based on the requirements of the DIMEMERCE system.

3.3.3cSequencediagram

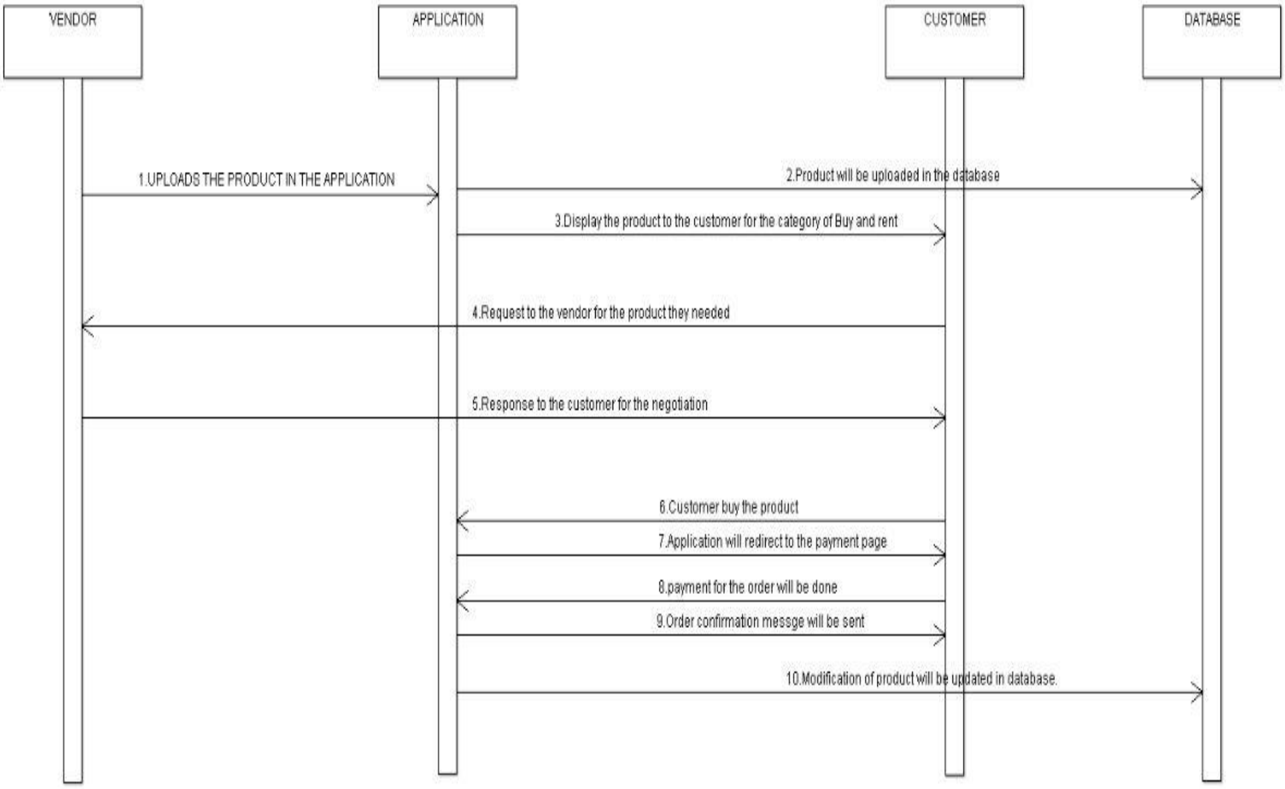

VENDOR
APPLICATION
CUSTOMER
DATABASE
1.UPLOADS THE PRODUCT IN THE APPLICATION
2.Product will be uploaded in the database
3.Display the product to the customer for the category of Buy and rent
4.Request to the vendor for the product they needed
5.Response to the customer for the negotiation
6.Customer buy the product
7.Application will redirect to the payment page
8.payment for the order will be done
9.Order confirmation messge will be sent
10.Modification of product will be updated in database.

Fig3.3.3c Sequence diagram for DIMEMERCE

The sequence diagram of DIMEMERCE String matching shows the sequence of how technical team getting details from the government. Creating a sequence diagram for the E-commerce (DIMEMERCE) involves illustrating the interactions between various components or actors in the system over time. This sequence diagram provides a step-by-step visualization of the interactions between the beneficiary and the DIMEMERCE system during the application, verification, calculation, payment ,and receipt of benefits process.

3.3.3d Collaboration diagram

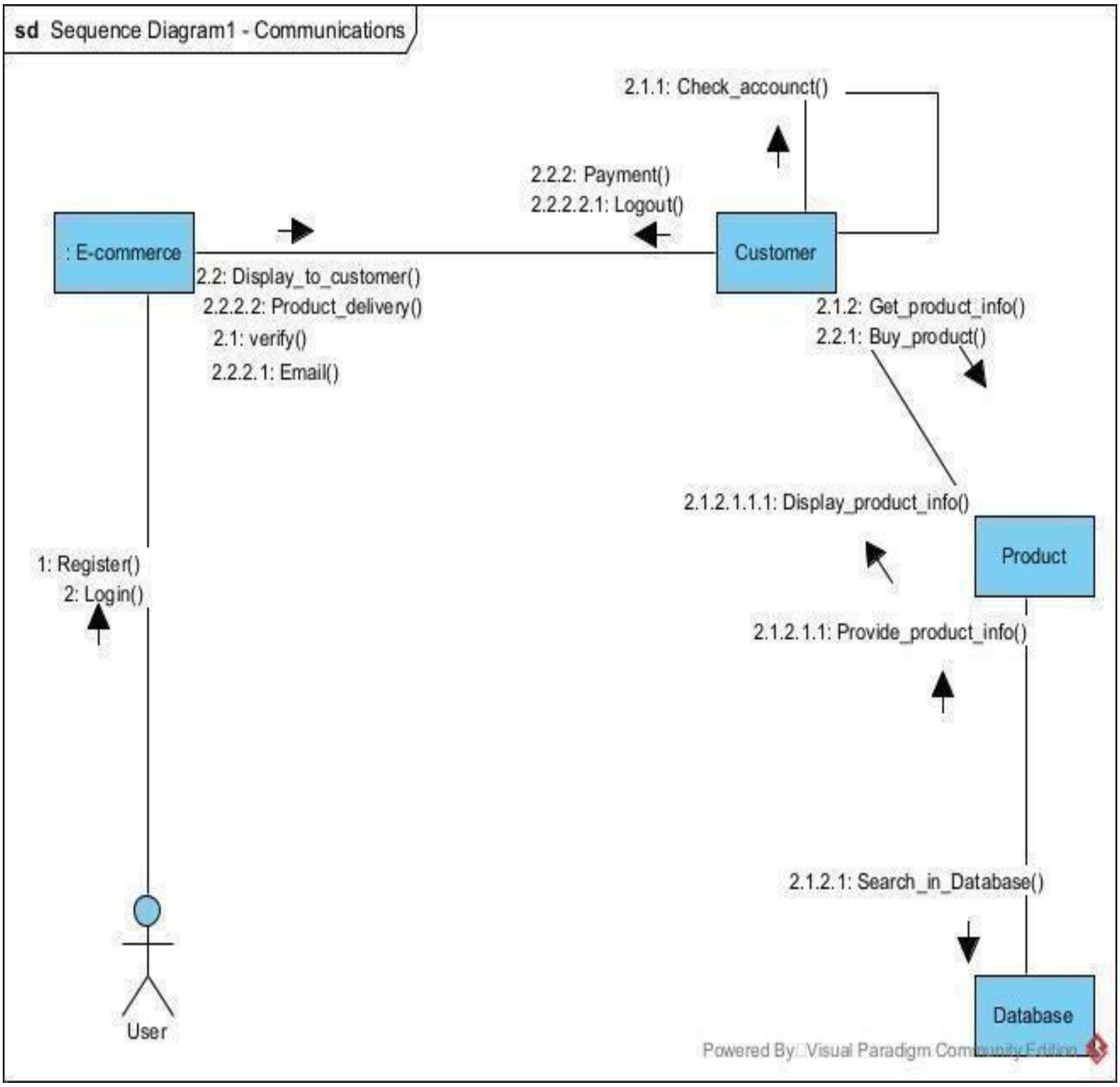

sd Sequence Diagram1 - Communications
2.1.1: Check_account()
2.2.2: Payment()
2.2.2.2.1: Logout()
: E-commerce
Customer
2.2: Display_to_customer()
2.2.2.2: Product_delivery()
2.1: verify()
2.2.2.1: Email()
2.1.2: Get_product_info()
2.2.1: Buy_product()
2.1.2.1.1.1: Display_product_info()
Product
1: Register()
2: Login()
2.1.2.1.1: Provide_product_info()
2.1.2.1: Search_in_Database()
User
Database
Powered By: Visual Paradigm Community Edition

Fig3.3.3d Collaboration diagram for DIMEMERCE

The collaboration diagram of DIMEMERCE String matching shows the sequence of activities of creating a awareness for people. A collaboration diagram, also known as a communication diagram, illustrates how objects collaborate to achieve a specific task or scenario. This collaboration diagram demonstrates the interactions between the beneficiary, application, administrator, and DIMEMERCE. Each step involves communication between different components of the system to achieve the overall objective of the E-Commerce.

3.3.3e Activity diagram

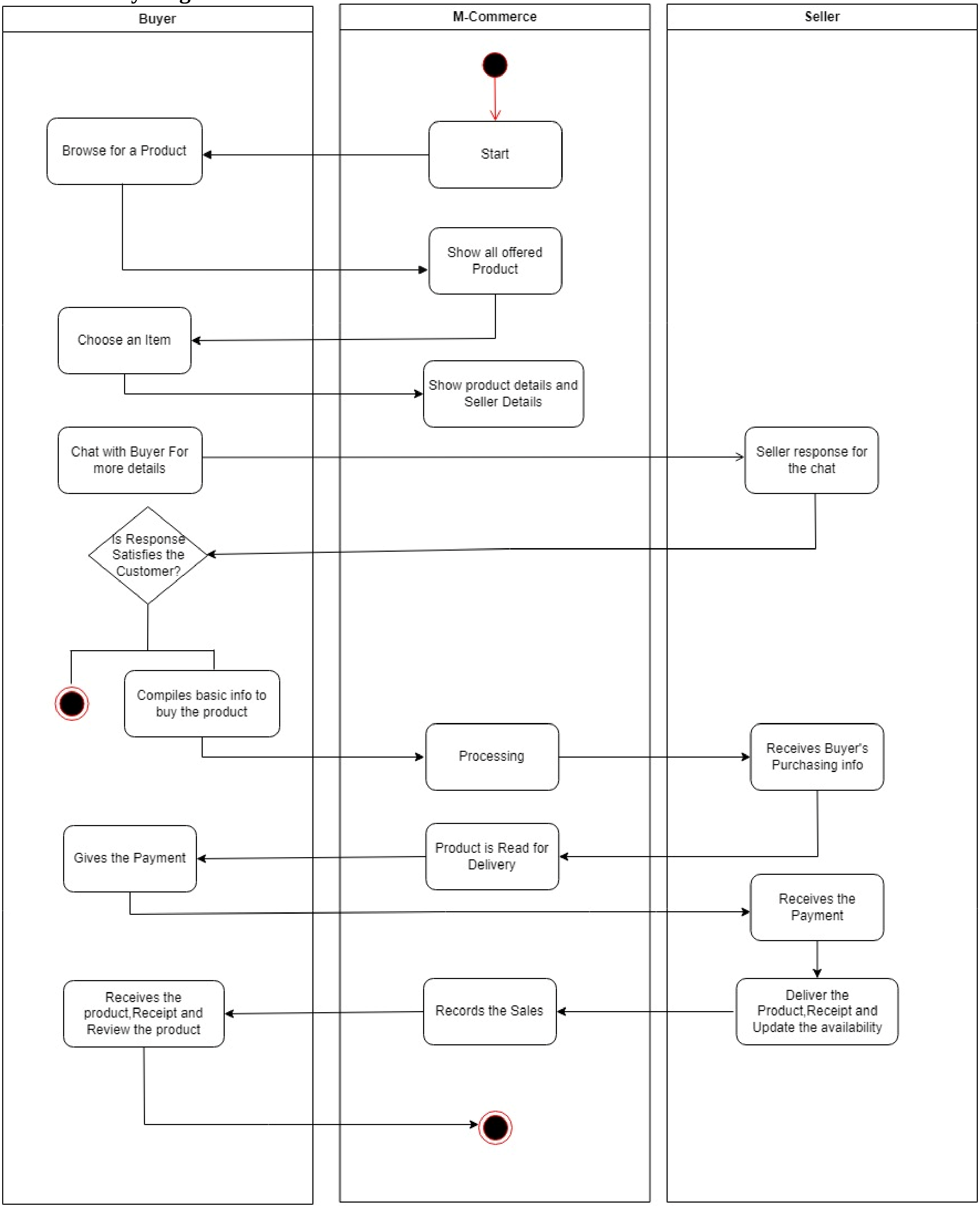

Buyer
M-Commerce
Seller
Start
Browse for a Product
Show all offered Product
Choose an Item
Show product details and Seller Details
Chat with Buyer For more details
Seller response for the chat
Is Response Satisfies the Customer?
Compiles basic info to buy the product
Processing
Receives Buyer's Purchasing info
Gives the Payment
Product is Read for Delivery
Receives the Payment
Receives the product,Receipt and Review the product
Records the Sales
Deliver the Product,Receipt and Update the availability

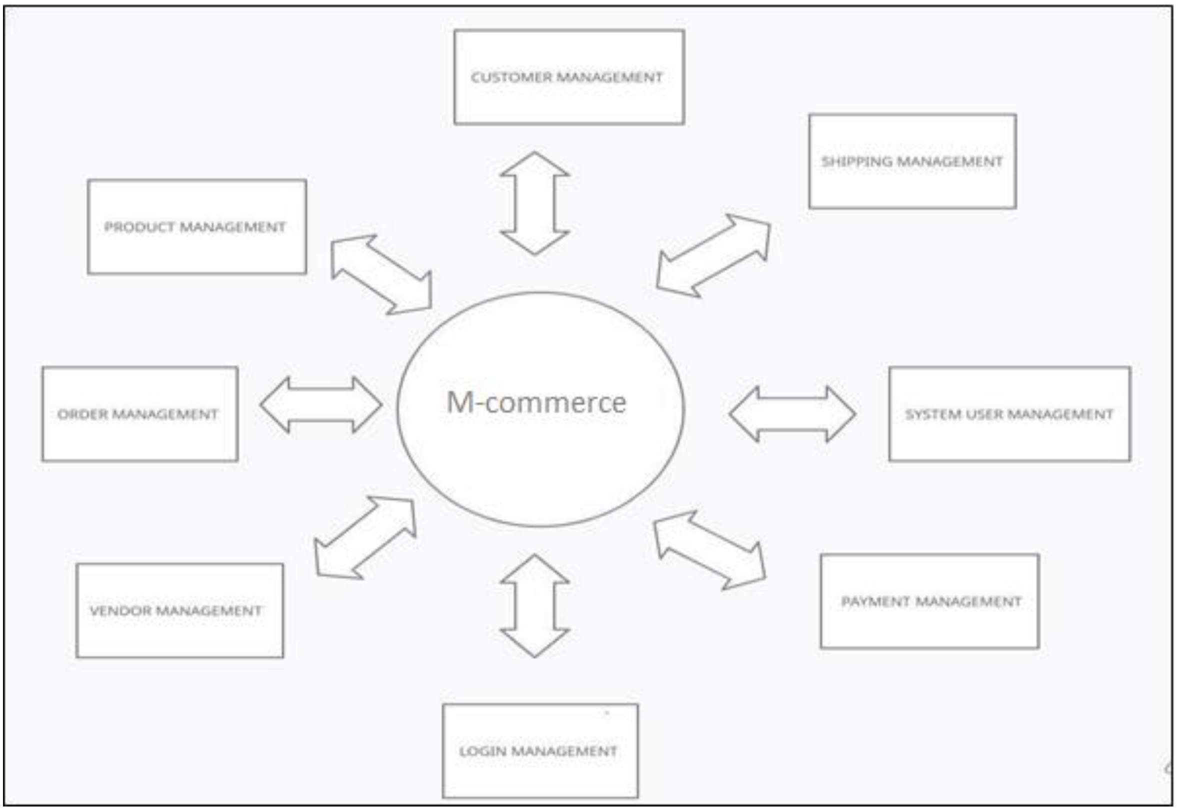
Fig3.3.3e Activity diagram for DIMEMERCE

The activity diagram shows the step-by-step process for DIMEMERCE String Matching. The above is a simplified activity diagram for E-Commerce (DIMEMERCE), illustrating the steps involved in the application process. This activity diagram provides a visual representation of the sequential steps involved in the application process for E-Commerce. Each step leads to the benefits are disbursed to the eligible beneficiary.

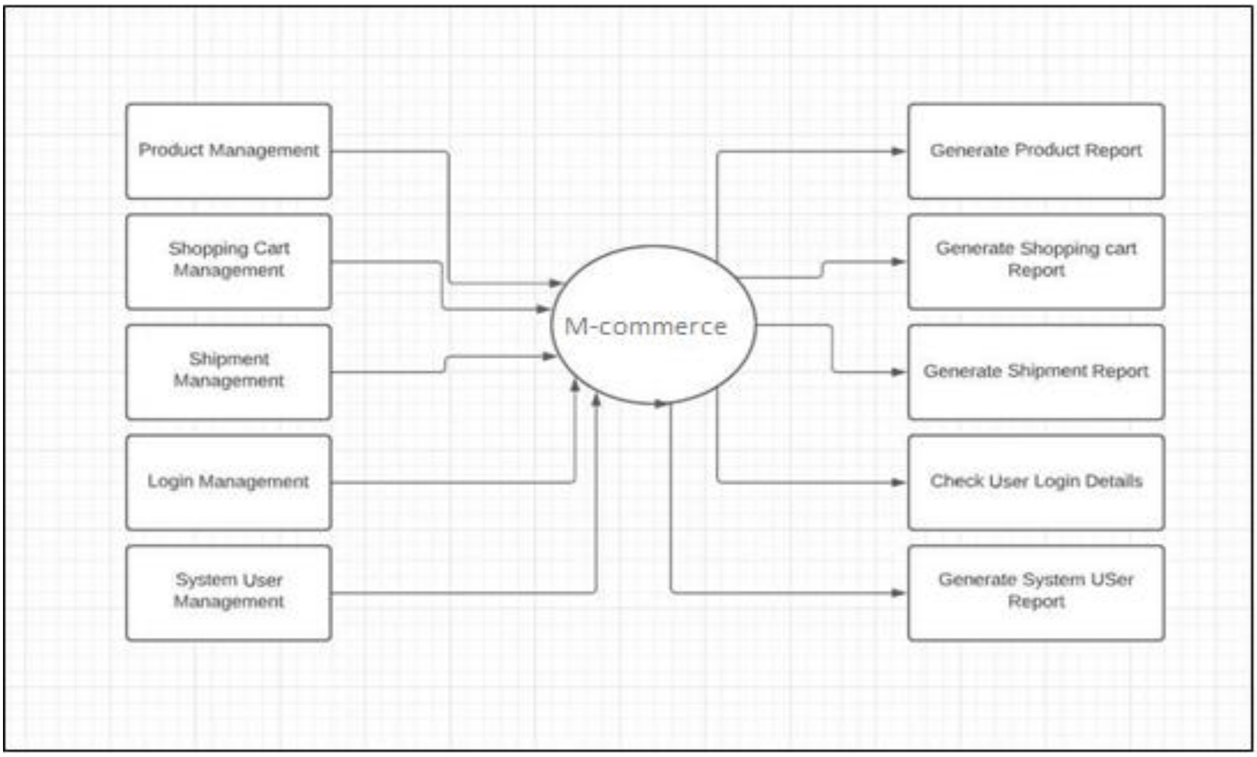
DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of dataprocessing.

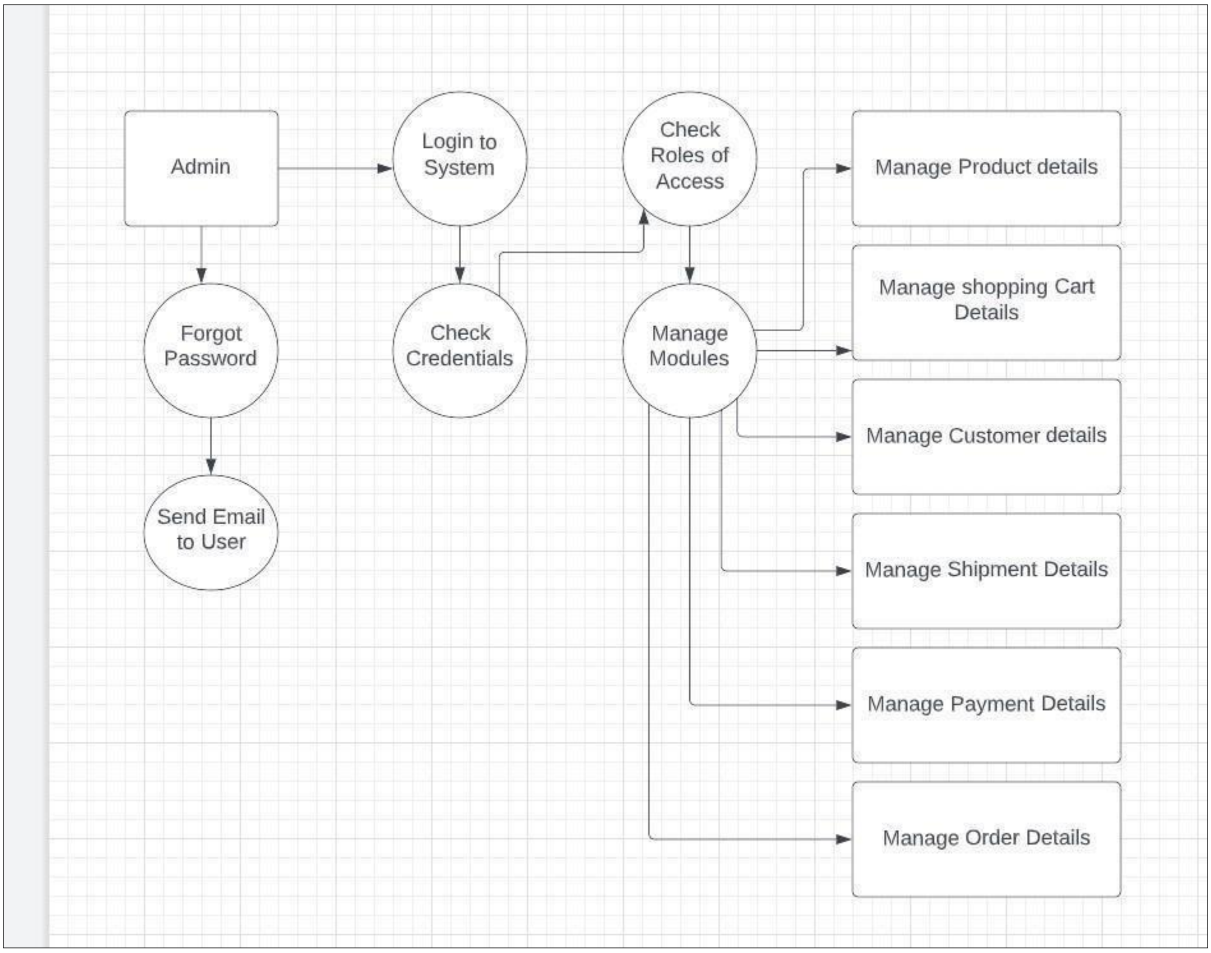
Level 0:



Level1:


Product Management
Shopping Cart Management
Shipment Management
Login Management
System User Management
M-commerce
Generate Product Report
Generate Shopping cart Report
Generate Shipment Report
Check User Login Details
Generate System USer Report

Level2:


Admin
Login to System
Check Roles of Access
Manage Product details
Manage shopping Cart Details
Forgot Password
Check Credentials
Manage Modules
Manage Customer details
Send Email to User
Manage Shipment Details
Manage Payment Details
Manage Order Details

CHAPTER4
SYSTEM IMPLEMENTATION

M-Commerce consists of 4 modules:
They are

- Authentication and Validation module
- Category module
- Cart and payment module
- Review and Feedback Module

4.1 Authentication and Verification Module:

- Authentication is done by Firebase Authentication SDK.
- The Firebase Authentication SDK provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

4.2 Category module:

The category features in the app are Buying , Selling and Renting.

- The buying products are assured to be trustable products as the sellers are allowed to be upload products only if they provide any national Id.
- The renting products also assured as the lessor identification is validated with their national Id.
- The selling products must be assured such that the seller has to give the national Id.

4.3 Card and Payment module:

- Razorpay is a payment gateway in which the app is integrated.
- Razorpay is one of the most commonly used payment gateways in India and it enables to make payments through various payment modes such as debit/credit cards ,net banking ,UPI ,digital wallets etc.

4.4 Review and Feedback module:

- Each and every products in the app is valued through its review and feedback.
- The most positive review and feedback will make the product to sell at high rate through our recommended product lists.

CHAPTER 5
RESULTS AND DISCUSSION

5.1 TESTING

TEST CASE ID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
1.	Selecting “SIGNUP” button	Display signup page	Display signup page	Pass
2.	Selecting the “BECOME A VENDOR” button	Display vendor sign up page	Vendor sign up page is displayed	Pass
3.	Select “SIGN IN” Button	Display sign-in box	Sign in page is displayed	Pass
4.	Selecting the “MOBILES” button	Enter into mobile product page	Enter into mobile product page	Pass
5.	Selecting “ELECTRONICS” Button	Enter into electronics product page	Enter into electronic product page	Pass
6.	Selecting the “UPLOAD” button	Uploads the details of the product	Uploads the details of the product	Pass
7.	Selecting “ADD PRODUCT” Button	Display the details should add for the products	Displays the details that should add for the product	Pass

Testcases and Report Table for DIMEMERCE

5.2 RESULT

Assessing the results of the E-Commerce (DIMEMERCE)involves examining various aspects of its performance and impact on the targeted beneficiaries. Here are some key results and outcomes associated with DIMEMERCE:

Increased financial security: DIMEMERCE has provided a source of regular income to millions of elderly individuals, widows, and persons with disabilities who were living below the poverty line. This has contributed to their financial security and helped them meet their basic needs.

Improved health outcomes: The financial assistance provided under DIMEMERCE has enabled beneficiaries to access healthcare services and essential medications, leading to improved health outcomes and wellbeing among the targeted population.

Reduction in vulnerability: By providing a safety net to vulnerable sections of society, DIMEMERCE has helped reduce their vulnerability to economic shocks, natural disasters, and other emergencies.

Promotion of financial inclusion: DIMEMERCE has facilitated financial inclusion by encouraging beneficiaries to open bank accounts and access other financial services, thereby promoting their economic empowerment and resilience.

Recognition of rights: DIMEMERCE has helped raise awareness about the rights of vulnerable groups and the importance of social security, contributing to efforts to protect and promote the rights within society.

5.3 DISCUSSION

Discussing the E-Commerce (DIMEMERCE) involves examining its objectives ,effectiveness, challenges, and potential for improvement. Here are some points to consider in a discussion about DIMEMERCE:

Objectives: DIMEMERCE aims to provide financial assistance and social security to vulnerable sections of society, including the elderly, widows, and persons with disabilities, who are living below the poverty line. It seeks to ensure a minimum standard of living and promote social inclusion among marginalized groups.

Effectiveness: DIMEMERCE has been successful in providing a safety net for millions of beneficiaries, offering them regular financial assistance to meet their basic needs. It has contributed to poverty alleviation efforts, improved health outcomes, and empowered women by providing them with financial independence.

Coverage and Reach: Despite its achievements, DIMEMERCE faces challenges related to coverage gaps and reaching all eligible beneficiaries, particularly in remote and marginalized areas. Efforts are needed to enhance outreach and ensure that all deserving individuals can access the benefits of the program.

Administrative Efficiency: Streamlining administrative processes and improving the efficiency of implementation is crucial for maximizing the impact of DIMEMERCE. This includes ensuring timely disbursal of payments, minimizing bureaucratic hurdles, and enhancing transparency and accountability in program management.

Financial Inclusion: Promoting financial inclusion among beneficiaries by facilitating access to banking services and promoting financial literacy can enhance their economic empowerment and resilience.

Sustainability: Ensuring the long-term sustainability of DIMEMERCE requires adequate funding, strategic planning, and regular monitoring and evaluation to assess its impact and effectiveness.

Partnerships and Collaboration: Collaborating with civil society organizations, community groups, and other stakeholders can strengthen the implementation of DIMEMERCE and improve its reach and impact.

.

CHAPTER 6

CONCLUSION AND FUTURE WORK

CONCLUSION:

- In 2022, an increase of 3.4% was estimated of the generated e-waste globally, hitting 59.4Mt , which made the total unrecyclable e-waste on earth to 2022 is over 347 Mt.
- Most of the e-waste is generated due to unused product.
- We can able to convert most of the good condition unused product to useful product which makes reduces of 15-30% e-waste.
- This makes intermediate free , so that both Buyer and seller can able to get profitable. Thus it makes the economy upturn in local retail shops .

FUTURE ENHANCEMENTS

To make all types of products in a Single platform. The category page will also will have a category of price-drop product. Map implementation of all the shops in the entire location .To make the available product during flood or bad weather condition and reduce delivery time.

REFERENCES

1.Kai Fan*,Hui Li, Wei Jiang, Chengsheng Xiao, and Yintang Yang," Secure Authentication Protocol for Mobile Payment ",TSINGHUA SCIENCE AND TECHNOLOGY ISSNll1007-0214 09/10 pp610–620,Volume 23, Number 5, October 2018.

2.Ju Ouyang, and Xianping Chen ,"Personal Information Two-dimensional Code Encryption Technology in the Process of E-commerce Logistics Transportation ",SOUTH AFRICAN INSTITUTE OF ELECTRICAL ENGINEERS Vol.113 (1) March 2022.

3.Junyi He , Di Zhang , Ju Ren,YuezhiZhou,andYaoxueZhang,"Online Market Mechanism for Mobile Data Rate Trading With Temporal Constraints",IEEE INTERNET OF THINGS JOURNAL, VOL. 9, NO. 20, 15 OCTOBER 2022

4.Falah Y H Ahmed,Eizwan Bin Hazlan and Muhammad Irsyad Abdulla,” Enhancement of Mobile-Based Application for Vehicle Rental”,2021 IEEE 11th IEEE Symposium on Computer Applications

& Industrial Electronics (ISCAIE), Penang, Malaysia, 03-04 April 2021.

5. Eric Hsueh-Chan Lu and Zhan-Qing Lin,” Rental Prediction in Bicycle-Sharing System Using Recurrent Neural Network”,IEEE ACCESS, Received April 12, 2020, accepted May 2, 2020, date of publication May 14, 2020, date of current version May 29, 2020.

6. Fumin Zhu , Chen Zhang , Zunxin Zheng and Sattam Al Otaib,” Click Fraud Detection of Online Advertising–LSH Based Tensor Recovery Mechanism”, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 23, NO. 7, JULY 2022.

APPENDICES

A.1 SDGGOALS

The E-Commerce (DIMEMERCE) in India contributes to several Sustainable Development Goals (SDGs) outlined by the United Nations. Here are some of the SDGs that DIMEMERCE aligns with:

GOAL 1: Decent Work and Economic Growth: By providing financial assistance, DIMEMERCE helps beneficiaries meet their basic needs and contributes to economic growth by stimulating local economies and reducing income inequality.

GOAL 2: Reduced Inequalities: DIMEMERCE aims to reduce inequalities by targeting vulnerable populations and providing them with marketed price and social security, thereby promoting social inclusion andequity.

GOAL 3: Peace, Justice, and Strong Institutions: DIMEMERCE contributes to SDG 16 by promoting social justice and strengthening institutions through its efforts to provide financial assistance and support to vulnerable populations, ensuring the rights and dignity are upheld.

A.2 SOURCECODE

HOMEPAGE.DART

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:carousel_slider/carousel_slider.dart';
import 'package:image_picker/image_picker.dart';
import 'package:m_commerce/pages/Post.dart';
import 'package:m_commerce/pages/home/drawer.dart';
import 'package:m_commerce/pages/home/Search.dart';
import 'package:m_commerce/pages/login/rent_splash.dart';
import 'package:m_commerce/pages/viewproduct.dart';
import '../category/seeAll.dart';
import 'package:animated_background/animated_background.dart';

class Homepage extends StatefulWidget {
  const Homepage({super.key});

  @override
  State<Homepage>createState() => _HomepageState();
}

class _HomepageState extends State<Homepage> with TickerProviderStateMixin {
  List upl_image = [];
getData() async {
QuerySnapshotqn =
        await FirebaseFirestore.instance.collection("upl_image").get();
    for (int i = 0; i<qn.docs.length; i++) {
setState(() {
upl_image.add({
          "image": qn.docs[i]["image"],
          "PnameController": qn.docs[i]["pnameController"],
          "PdesController": qn.docs[i]["pdesController"],
          "MrpController": qn.docs[i]["mrpController"],
          "PpriceController": qn.docs[i]["ppriceController"],
          "AddrController": qn.docs[i]["addrController"],
          "categoryController": qn.docs[i]["categoryController"],
          "QuantityController": qn.docs[i]["QuantityController"]
        });
        // print(qn.docs[i]["image"]);
      });
    }
    return qn.docs;
  }

  @override
  void initState() {
getData();
super.initState();
  }

  Widget _buildCategory({required String name, required String photo}) {
    return Card(
shadowColor: Colors.grey,
      elevation: 4,
      child: SizedBox(
        height: 160,
        width: 180,
        child: Column(
mainAxisAlignment: MainAxisAlignment.start,
          children: [
Container(
              height: 140,
              width: 150,
              decoration: BoxDecoration(
                  color: Colors.deepPurple,
                  image: DecorationImage(
                      image: AssetImage(
                        "images/$photo",
                      ),
                      fit: BoxFit.fill)),
),
Text(
              name,
              style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 16),
            )
          ],
        ),
      ),
    );
  }

  Widget _buildShop({required String photos}) {
    return Card(
        color: Colors.white,
        shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(25)),
        elevation: 6,
        child: Container(
            height: 100,
            width: 100,
         List icon = [
Icons.person,
Icons.email_rounded,
Icons.location_on,
Icons.phone_android_rounded,
Icons.headset_mic
  ];
  List name = ["email Id", "Address", "Phone no.", "Help Center"];
ImagePicker? image = ImagePicker();
  Future getImage(ImageSource gallery) async {
    final image = await ImagePicker().pickImage(source: ImageSource.gallery);
    if (image == null) return;
setState(() {});
  }

  final data = ['1', '2'];
  final List<String>imgList = [
    'images/image1.jpeg',
    'images/image2.jpeg',
    'images/image3.jpg'
  ]
                ],
              ),
SingleChildScrollView(
scrollDirection: Axis.horizontal,
                child: Row(
                  children: [
                    _buildCategory(
                        photo: "electronics.jpg", name: "Electronics"),
                    _buildCategory(photo: "dresses.jpg", name: "Fashion"),
                    _buildCategory(photo: "Furniture.jpeg", name: "Furniture"),
                    _buildCategory(photo: "phone.webp", name: "Mobiles"),
                    _buildCategory(photo: "Grocery.png", name: "Grocery"),
                    _buildCategory(photo: "toys.jpeg", name: "Toys"),
                    _buildCategory(photo: "sports.jpg", name: "Sports"),
                    _buildCategory(photo: "Utensils.webp", name: "Home"),
                  ],
                ),
              ),

              const Padding(
                padding: EdgeInsets.only(left: 16.0, top: 20),
                child: Align(
                    alignment: Alignment.topLeft, child: Text("Recommended")),
              ),

SizedBox(
    height: 950,

                                    children: [
TextSpan(
                                        text:
                                            '\u{20B9}${upl_image[i]["MrpController"]}',
                                        style: const TextStyle(
                                            decoration:
TextDecoration.lineThrough,
fontSize: 11,
                                            color: Colors.grey),
                                      ),
TextSpan(
                                    text:
                                        ' \u{20B9}${upl_image[i]["PpriceController"]}',
                                    style: const TextStyle(
fontWeight: FontWeight.bold,
fontSize: 18)),
                              ],
                            ),
                          ),
                        )
              ])),
                   ),
                ],
              ),
            ),
          ],
        ),
      ),
    ),
floatingActionButton: Row(
mainAxisAlignment: MainAxisAlignment.spaceAround,
       children: [
Padding(
           padding: const EdgeInsets.only(left: 30.0),
           child: FloatingActionButton.extended(
heroTag: "btn1",
focusColor: Colors.amber,
backgroundColor: Colors.black,
              icon: const Icon(
Icons.shopping_cart_checkout_rounded,
                color: Colors.amber,
              ),
onPressed: () {
Navigator.push(context,
MaterialPageRoute(builder: (context) => const Splash()));
              },
              label: const Text(
                "Rental Products",
                style: TextStyle(color: Colors.amber),
              )),
        ),
Padding(
           padding: const EdgeInsets.only(left: 35.0),
           child: FloatingActionButton.extended(
heroTag: "btn2",
             elevation: 30,
             label: const Text(
               "Upload",
               style: TextStyle(color: Colors.white),
  ),
backgroundColor: Colors.amber[600],
             icon: const Icon(
      Icons.add,
                color: Colors.white,
              ),
      onPressed: () {
      Navigator.push(context,
      MaterialPageRoute(builder: (context) => const Post()));
              },
            ),
          ),
        ],
      ),
    );
  }
}
```

CATEGORY.DART

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:carousel_slider/carousel_slider.dart';
import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import '../viewproduct.dart';

class CategoryPage extends StatefulWidget {
CategoryPage({this.category, this.slider});
  var category;
  var slider;
  @override
  State<CategoryPage>createState() => _CategoryPageState();
}

class _CategoryPageState extends State<CategoryPage> {
  @override
  void initState() {
    // TODO: implement initState
super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
appBar: AppBar(
        leading: IconButton(
onPressed: () {
Navigator.of(context).pop();
              },
              icon: Icon(
Icons.arrow_back_ios,
                color: Colors.black,
              )),
automaticallyImplyLeading: false,
toolbarHeight: 70,
centerTitle: true,
backgroundColor: Colors.white,
          elevation: 5,
          title: Text(widget.category,
              style: GoogleFonts.poppins(
                color: Colors.black,
fontSize: 27,
fontWeight: FontWeight.w500,
              )),
        ),
        body: Column(children: [
Container(
            height: 200,
            width: 410,
            color: Colors.amber,
            child: CarouselSlider(
                items: widget.slider
.map<Widget>(
                      (item) =>Container(
                        child: Padding(
                          padding: const EdgeInsets.symmetric(vertical: 8.0),
                          child: Center(
                            child: CachedNetworkImage(
imageUrl: item,
                              fit: BoxFit.cover,
                              width: 1000,
                            ),
                                          image: documentSnapshot['image'],
                                          name: documentSnapshot[
                                              'PnameController'],
                                          old:
documentSnapshot['MrpController'],
addr: documentSnapshot[
                                              'AddrController'],
                                        ))),
                            child: Container(
                              decoration: BoxDecoration(
                                  color: Colors.white,
borderRadius: BorderRadius.circular(13)),
                              child: Column(
                                children: [
Container(
                                      height: 160,
                                      color: Colors.white,
                                      child: CachedNetworkImage(
imageUrl: documentSnapshot["image"],
                                      )),
Container(
                                    alignment: Alignment.center,
                                    height: 75,
                                    color: Colors.white,
                                    child: Padding(
                                      padding: const EdgeInsets.symmetric(
                                          horizontal: 4.0),
                                      child: Text(
documentSnapshot["PnameController"]),
                                    ),
                                  ),
Expanded(
                                    child: Container(
                                      child: Text.rich(
TextSpan(
                                          children: [
TextSpan(
                                              text: '\u{20B9}' +
documentSnapshot[
                                                    "MrpController"],
                                              style: const TextStyle(
                                                  decoration: TextDecoration
.lineThrough,
fontSize: 11,
                                                  color: Colors.grey),
                                            ),
TextSpan(
                                              text: '\u{20B9}' +
documentSnapshot[
                                                      "PpriceController"],
                                              style: TextStyle(
fontWeight: FontWeight.bold,
fontSize: 18)),
                                          ],
                                        ),
                                      ),
                                    ),
                                  )
                                ],
                              ),
                            ));
                      }
                    },
                  );
                }),
          )
        ]));
    }
}
```

PRODUCT.DART

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:m_commerce/pages/home/Search.dart';
import 'package:url_launcher/url_launcher.dart';
import 'package:velocity_x/velocity_x.dart';

class ViewProduct extends StatefulWidget {
  var image;
  var name;
  var des;
  var old;
  var New;
  var addr;
  var quan;
ViewProduct(
      {this.New,
this.des,
this.addr,
this.image,
      this.name,
this.old,
this.quan});

  @override
  State<ViewProduct>createState() => _ViewProductState();
}

class _ViewProductState extends State<ViewProduct>
    with SingleTickerProviderStateMixin {
  static Future<void>openMAp(String addr) async {
    String googlemapUrl =
        "https://www.google.com/maps/search/?api=1&query=$addr";

    if (await canLaunch(googlemapUrl)) {
      await launch(googlemapUrl);
    } else {
      throw "Could not Open the Map";
    }
  }
  late AnimationController controller;
  List review = [
    "Video",
    "Reviews",
    "Privacy Policy",
    "Return Policy",
    "Support Policy"
  ];
  @override
  void initState() {
super.initState();
    controller =
AnimationController(duration: Duration(seconds: 3), vsync: this);
controller.addStatusListener((status) async {
      if (status == AnimationStatus.completed) {
Navigator.pop(context);
      }
    });
  }
  @override
  void dispose() {
controller.dispose();
super.dispose();
  }
  Future addtocart() async {
    final FirebaseAuth _auth = FirebaseAuth.instance;
    var currentUser = _auth.currentUser;
                ),
              ),
Padding(
                padding: const EdgeInsets.all(10.0),
                child: Row(children: [
Text.rich(
TextSpan(
                      children: [
TextSpan(
                          text: '\u{20B9}${widget.old}',
                          style: const TextStyle(
                              decoration: TextDecoration.lineThrough,
fontSize: 13,
                              color: Colors.grey),
                        ),
TextSpan(
                            text: ' \u{20B9}${widget.New}',
                            style: const TextStyle(
fontWeight: FontWeight.bold, fontSize: 18)),
                      ],
                    ),
                  ),
                ]),
              ),
              10.heightBox,
Padding(
                padding: const EdgeInsets.symmetric(horizontal: 8),
                child: Row(
mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: [
ElevatedButton.icon(
onPressed: () {},
                    style: ButtonStyle(
                      elevation: const MaterialStatePropertyAll(10),
                      shape: MaterialStatePropertyAll(
RoundedRectangleBorder(
borderRadius: BorderRadius.circular(20))),
backgroundColor:
MaterialStatePropertyAll(Colors.amber)),
                    icon: Icon(Icons.message),
                    label: const Text("Message")),
ElevatedButton.icon(
onPressed: () {
openMAp(widget.addr);
                    },
                    style: ButtonStyle(
                      elevation: MaterialStatePropertyAll(10),
                      shape: MaterialStatePropertyAll(
                padding:
                  const EdgeInsets.only(left: 18.0, right: 18, bottom: 40),
                child: ListView.builder(
itemCount: review.length,
itemBuilder: ((context, index) {
                   return Padding(
                    padding: EdgeInsets.all(8),
                    child: ListTile(
contentPadding: EdgeInsets.all(3),
                      title: Text(
                        '${review[index]}',
                        style: TextStyle(
fontWeight: FontWeight.w600, fontSize: 18),
                      ),
                      trailing: Icon(
Icons.arrow_forward_outlined,
                        color: Colors.black,
                      ),
                    ),
                  );
                })),
            ))
          ],
        ),
      ),
    ),
floatingActionButton: StreamBuilder(
        stream: FirebaseFirestore.instance
.collection("Favorite_item")
            .doc(FirebaseAuth.instance.currentUser!.email)
.collection("item")
.where("PnameController", isEqualTo: widget.name)
.snapshots(),
        builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
          try {
```

```
          return Theme(
              data: Theme.of(context).copyWith(
floatingActionButtonTheme:
                    const FloatingActionButtonThemeData(
extendedSizeConstraints: BoxConstraints.tightFor(
                          height: 50, width: 180))),
              child: FloatingActionButton.extended(
                  elevation: 10,
backgroundColor: Colors.amber,
splashColor: Colors.amber.shade800,
onPressed: () =>addtocart(),
                  icon: const Icon(Icons.add_shopping_cart),
                  label: const Text("ADD TO CART")));
        } catch (e) {
          print(e);
        }
        return Container();
      },
    ),
  );
  }
}
```

SEARCH.DART

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lottie/lottie.dart';
import 'package:m_commerce/pages/viewproduct.dart';
import 'package:velocity_x/velocity_x.dart';
class SearchPage extends StatefulWidget {
  const SearchPage({super.key});
  @override
  State<SearchPage>createState() => _SearchPageState();
}
class _SearchPageState extends State<SearchPage> {
  String? _search;
  @override
  Widget build(BuildContext context) {
    double height = MediaQuery.of(context).size.height;
    double width = MediaQuery.of(context).size.width;
    return Scaffold(
appBar: AppBar(
backgroundColor: Colors.amber,
toolbarHeight: height * 0.09,
centerTitle: true,
automaticallyImplyLeading: false,
        title: Row(mainAxisAlignment: MainAxisAlignment.start, children: [
IconButton(
onPressed: () {
```

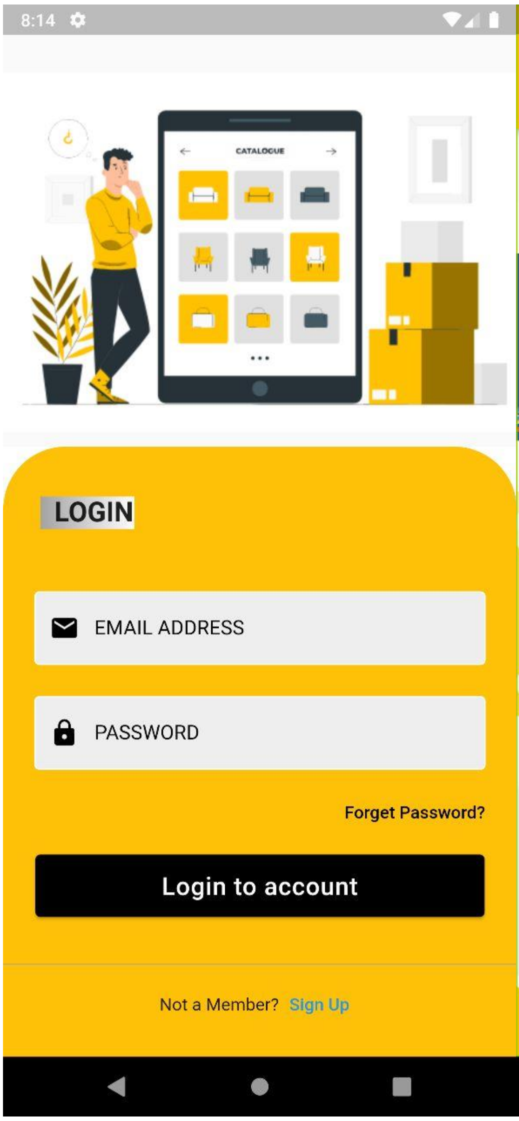
```
Navigator.of(context).pop();
              },
              icon: const Icon(Icons.arrow_back_ios)),
SizedBox(
            width: width * 0.75,
            child: TextFormField(
                autofocus: true,
onChanged: ((value) {
setState(() {
                    _search = value;
                  });
                }),
                decoration: const InputDecoration(
                    filled: true,
fillColor: Colors.white,
prefixIcon: Icon(Icons.search),
hintText: "Search products",
labelStyle: TextStyle(
fontSize: 17,
                      color: Colors.black,
fontWeight: FontWeight.w400,
                    ),
                    border: OutlineInputBorder(
borderRadius: BorderRadius.all(Radius.circular(30))))),
          ),
        ]),
      ),
      body: Stack(children: [
StreamBuilder(
            stream: FirebaseFirestore.instance
.collection('Seller_req')
                .doc("A3YtV51DsJGf7ruoFV6x")
.collection("item")
.where('PnameController', isEqualTo: _search?.toLowerCase())
.snapshots(),
            builder:
                (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
              if (!snapshot.hasData) {
                return const Center(
                  child: CircularProgressIndicator(),
                );
              } else if (_search != null) {
                return ListView.builder(
itemCount: snapshot.data?.docs.length,
itemBuilder: (context, index) {
DocumentSnapshotdocumentSnapshot =
snapshot.data!.docs[index];
                      String _name =
snapshot.data?.docs[index]['PnameController'];
                      String img = snapshot.data?.docs[index]['image'];

                      return Padding(
                        padding: const EdgeInsets.all(5.0),
                        child: Card(
                          elevation: 3,
                          child: ListTile(
                            title: Text(_name),
                            leading: Container(
                              child: Image.network(
img,
                                height: 100,
                                fit: BoxFit.cover,
                                width: 100,
                              ),
                            ),
onTap: () {
Navigator.push(
                                  context,
MaterialPageRoute(
                                      builder: (context) =>ViewProduct(
                                            New: documentSnapshot[
                                                "PpriceController"],
addr: documentSnapshot[
                                                'AddrController'],
                                            des: documentSnapshot[
                                                'PdesController'],
                                            image: documentSnapshot['image'],
                                            name: documentSnapshot[
                                                'PnameController'],
                                            old: documentSnapshot[
                                                'MrpController'],
quan: documentSnapshot[
                                                "QuantityController"],
                                          )));
                            },
                          ),
                        ),
                      );
                    });
              } else {
                return Container();
              }
            }),
Center(
          child: LottieBuilder.network(
              "https://assets10.lottiefiles.com/packages/lf20_yhetm7ld.json"),
        ),
      ]),
    );
  }
}
```

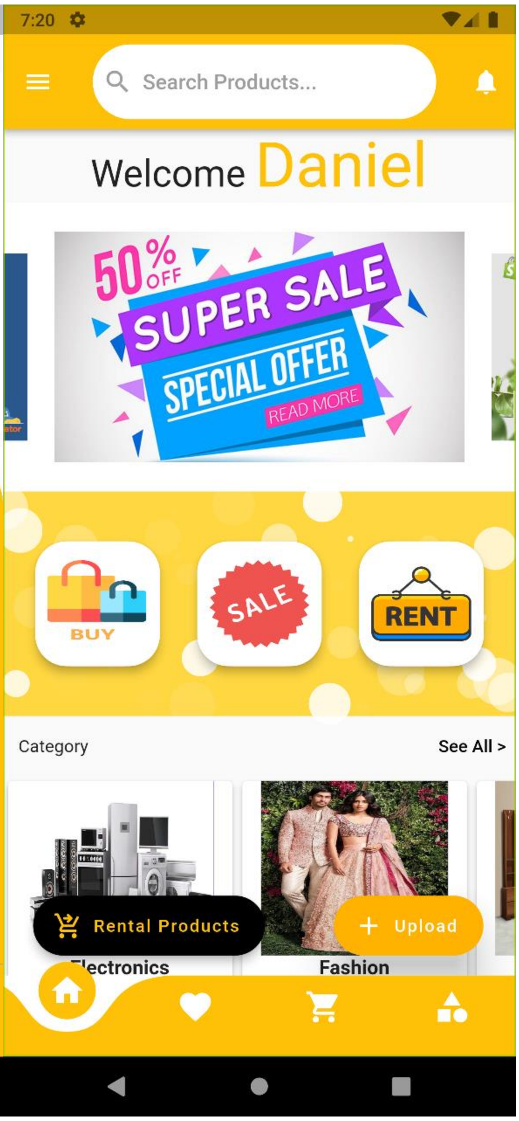
ADD TO CART.DART

```
import 'package:cached_network_image/cached_network_image.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:m_commerce/pages/viewproduct.dart';

class Cart extends StatefulWidget {
  const Cart({super.key});

  @override
  State<Cart>createState() => _CartState();
}

class _CartState extends State<Cart> {
  @override
  Widget build(BuildContext context) {
    double height = MediaQuery.of(context).size.height;
    double width = MediaQuery.of(context).size.width;
    int i = 0;
    double add = 0;

    return Scaffold(
appBar: AppBar(
automaticallyImplyLeading: false,
toolbarHeight: 80,
centerTitle: true,
backgroundColor: Colors.white,
        elevation: 5,
        title: Text("My Cart",
            style: GoogleFonts.poppins(
              color: Colors.black,
fontSize: 27,
fontWeight: FontWeight.w500,
            )),
      ),
      body: Stack(
        children: [
Center(
            child: Image.asset(
              "images/cart1.png",
              color: Colors.white.withOpacity(0.3),

          style: TextStyle(fontWeight: FontWeight.w500, fontSize: 20),
        ));
    return Scaffold(
appBar: AppBar(
          title: Text("Product Upload Page"),
        ),
        body: SingleChildScrollView(
          child: Container(
            margin: EdgeInsets.all(20),
            child: Column(
              children: [
TextField(
                  controller: SnameController,
                  decoration: const InputDecoration(
prefixIcon: Icon(Icons.person),
hintText: "Seller Name",
labelText: "Seller Name",
labelStyle: TextStyle(
fontSize: 17,
                        color: Colors.black,
fontWeight: FontWeight.w400,
                      ),
                      border: OutlineInputBorder()),
keyboardType: TextInputType.name,
textInputAction: TextInputAction.done,
                ),
Padding(
                  padding: const EdgeInsets.only(top: 14.0),
                  child: TextField(
                    controller: PnoController,
                    decoration: const InputDecoration(
prefixIcon: Icon(Icons.phone_android_outlined),
hintText: "Phone Numer",
labelText: "phone Number",
labelStyle: TextStyle(
fontSize: 17,
                          color: Colors.black,
fontWeight: FontWeight.w400,
                        ),
                        border: OutlineInputBorder()),
keyboardType: TextInputType.number,
maxLength: 10,
textInputAction: TextInputAction.done,
                  ),
                ),
Padding(
                  padding: const EdgeInsets.only(bottom: 15.0),
                  child: TextField(
                    controller: Gstcontroller,
                    decoration: const InputDecoration(
prefixIcon: Icon(Icons.content_paste_sharp),
hintText: "GST Number",
labelText: "GST Number",
labelStyle: TextStyle(
fontSize: 17,
                          color: Colors.black,
fontWeight: FontWeight.w400,
                        ),
                        border: OutlineInputBorder()),
keyboardType: TextInputType.name,
textInputAction: TextInputAction.done,
                  ),
                ),
TextFormField(
                  controller: emailcontroller,
                  decoration: const InputDecoration(
prefixIcon: Icon(Icons.email_rounded),
hintText: "email address",
labelText: "email address",
labelStyle: TextStyle(
fontSize: 17,
                        color: Colors.black,
fontWeight: FontWeight.w400,
                      ),
                      border: OutlineInputBorder()),
keyboardType: TextInputType.text,
textInputAction: TextInputAction.done,
                ),
Padding(
                  padding: EdgeInsets.only(top: 14.0),
                  child: TextFormField(
                    controller: AddrController,
                    decoration: const InputDecoration(
prefixIcon: Icon(Icons.location_on),
hintText: "Address",
labelText: "Address",
labelStyle: TextStyle(
fontSize: 17,
                          color: Colors.black,
fontWeight: FontWeight.w400,
                        ),
                        border: OutlineInputBorder()),
keyboardType: TextInputType.name,
maxLines: 3,
textInputAction: TextInputAction.done,
                  ),
                ),
Padding(
                  padding: const EdgeInsets.only(top: 14.0),
                  child: Container(
                    decoration: BoxDecoration(
                        border: Border.all(width: 1),
borderRadius: BorderRadius.circular(10)),
                    child: Column(children: [
                      const Align(
                          alignment: Alignment.centerLeft,
                          child: Text(
"  Location",
                            style: TextStyle(fontSize: 15),
                          )),
Padding(
                        padding: const EdgeInsets.all(8.0),
                        child: CSCPicker(
                          layout: Layout.vertical,
defaultCountry: CscCountry.India,
disableCountry: true,
onCountryChanged: (Country) {},
onStateChanged: (state) {
setState(() {
this.state = state;
                            });
                          },
onCityChanged: (city) {
setState(() {
this.city = city;
                            });
                          },
stateDropdownLabel: "State",
cityDropdownLabel: "City",
                        ),
                      ),
                    ]),
                  ),
                ),

                            color: Colors.black,
fontWeight: FontWeight.w400,
                          ),
                          border: OutlineInputBorder()),
keyboardType: TextInputType.number,
maxLength: 6),
                ),
Padding(
                  padding: const EdgeInsets.only(top: 14.0, right: 60),
                  child: TextField(
                      controller: PpriceController,
                      decoration: const InputDecoration(
                          // prefixIcon: Icon(Icons.shopping_cart),
                          icon: Icon(Icons.currency_rupee_sharp),
hintText: "Product Price",
labelText: "Product Price",
labelStyle: TextStyle(
fontSize: 17,
                            color: Colors.black,
fontWeight: FontWeight.w400,
                          ),
                          border: OutlineInputBorder()),
keyboardType: TextInputType.number,
maxLength: 6),
                ),
ElevatedButton(
onPressed: () {
senddata();
```

```
                    },
                    child: const Text("Upload"))
              ],
            ),
          ),
        ));
  }

getcam() async {
    // ignore: deprecated_member_use
    var img = await image.getImage(source: ImageSource.camera);
setState(() {
      file = File(img!.path);
    });
  }

getgal() async {
    // ignore: deprecated_member_use
    var img = await image.getImage(source: ImageSource.gallery);
setState(()
```

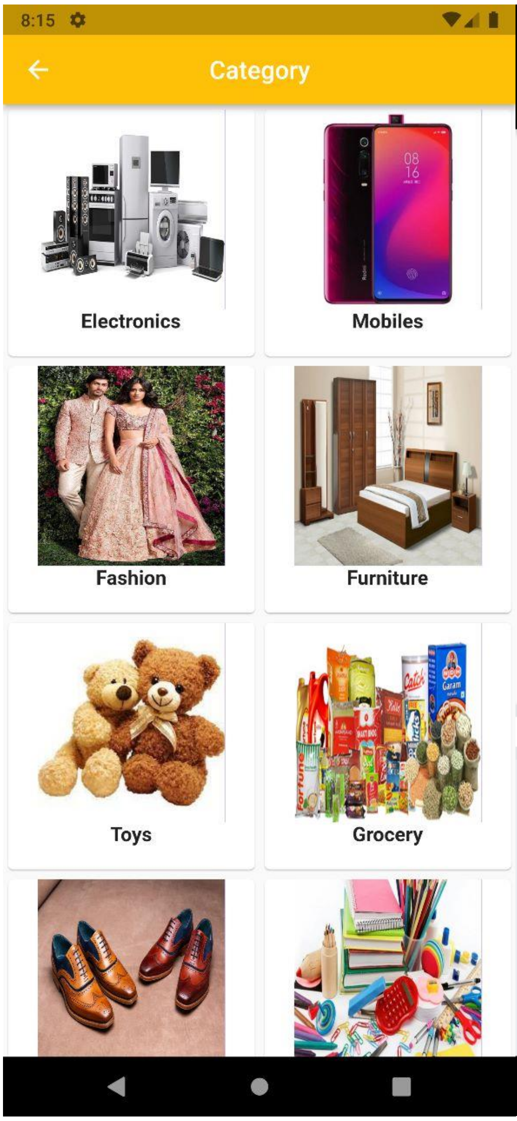
A.3 SCREENSHOT



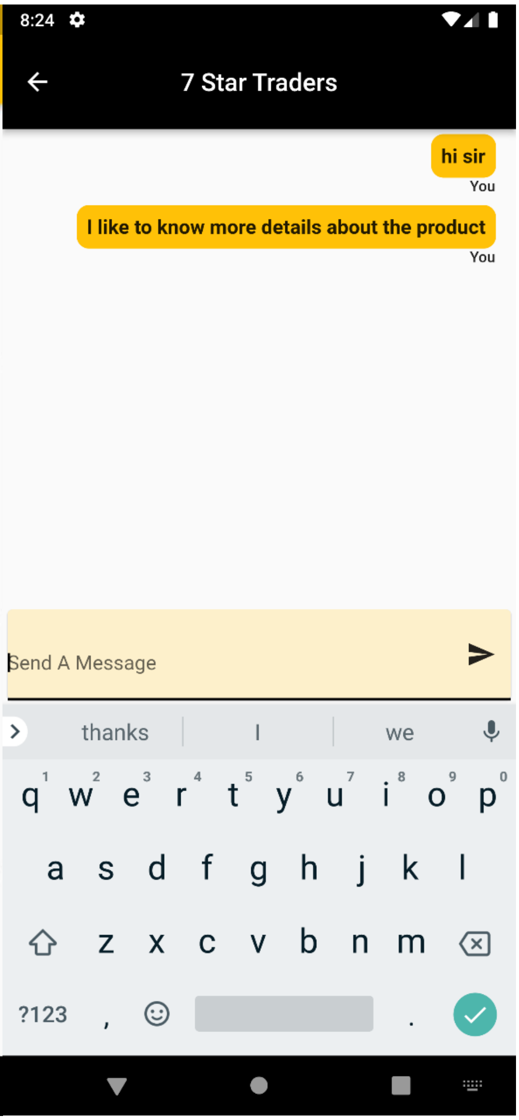


Login ScreenShot

Home Page



Category Page

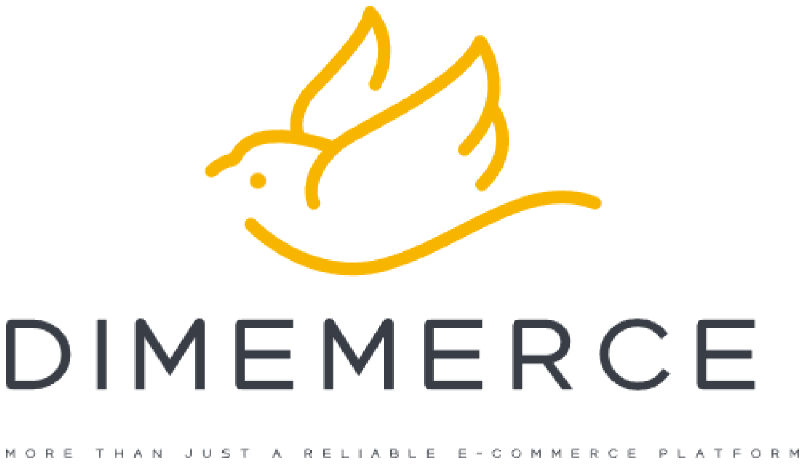


hi sir
You

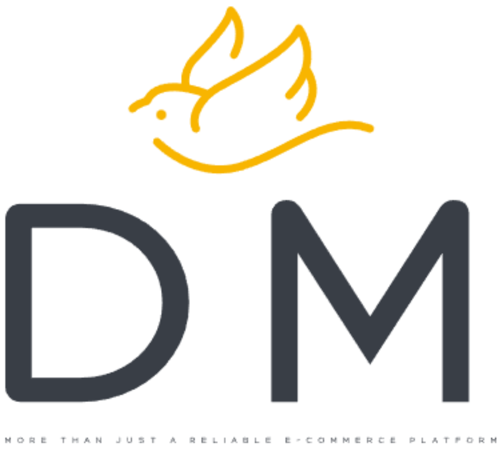
I like to know more details about the product
You

Send A Message

Chat Box



Application logo



Application Icon

PLAGIARISM REPORT