

# **Grocery Management System**

## **A PROJECT REPORT**

**Submitted By**

**Amit Jaiswal**  
(2100290140022)  
**Himanshu Pundir**  
(2100290140070)  
**Shivam Sharma**  
(2100290140129)

**Submitted in partial fulfilment of the  
Requirements for the Degree of**

## **MASTER OF COMPUTER APPLICATION**

**Under the Supervision of  
Dr. Shashank Bhardwaj  
(Associate Professor)**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS  
KIET Group of Institutions, Ghaziabad  
Uttar Pradesh-201206**

**( 2 JUNE 2023 )**

# **CERTIFICATE**

Certified that **Amit Jaiswal (2100290140022), Himanshu Pundir (2100290140070), Shivam Sharma (2100290140129)** has/ have carried out the project work having “**Grocery Management System**” for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Date:**

**Amit Jaiswal (2100290140022)**

**Himanshu Pundir (2100290140070)**

**Shivam Sharma (2100290140129)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

**Date:**

**Dr. Shashank Bhardwaj**  
(Associate Professor)  
Department of Computer Applications  
KIET Group of Institutions, Ghaziabad

**Signature of Internal Examiner**

**Signature of External Examiner**

**Dr. Arun Tripathi**  
Head, Department of Computer Applications  
KIET Group of Institutions, Ghaziabad

## **ABSTRACT**

This is a web-oriented application that allows us to access all information about the college, staff, students, facilities etc. This application provides a virtual tour of Campus. Here we will get the latest information about the students and staff. This generic application is designed for assisting the students at an institute regarding information on the courses, exams, grades and exam-timetable. It also provides support that a faculty can also check about his daily schedule, can create exams, prepare questions for Exams, and notices to the students. Here the administrator will manage the accounts of the student and faculties, check the overall performances of students, and upload the latest information about the campus.

We identify several problems including unauthorized privilege escalation, incorrect use of cryptography, vulnerabilities to network threats, and poor software development processes. We show that only college administrator can start the system administrator can search the particular student by his/her enrollment number or student id and we are adding notification module where administrator should add the notification with start and end date.

## ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Shashank Bhardwaj** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions. Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Amit Jaiswal**

**Himanshu Pundir**

**Shivam Sharma**

# TABLE OF CONTENTS

Certificate	i
Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Abbreviations	vii
List of Figures	viii
1 Introduction	1-7
1.1 Project Objectives	1-2
1.2 Project Scope	3
1.3 Project Description	4
1.4 Hardware / Software	5
1.5 Functional Requirements	6
1.6 Non- Functional Requirements	7
2 Feasibility Study	8-17
2.1 Technical feasibility	9-10
2.2 Operational Feasibility	11-12
2.3 Behavioral Feasibility	13-15
2.4 Economical Feasibility	15-17
3 Database Design	18-39
3.1 Database Table	20-21
3.1 Flow Chart Diagram	22-23
3.2 Data Flow Diagram	24-30
3.3 Entity Relationship Diagram	30-33
3.4 Use Case Diagram	34-36

3.5	Sequence Diagram	37-38
3.6	Collaborative Diagram	39
4	Form Design	40-45
4.1	Input / Output (Screenshot)	40-42
4.1.1	Login	40
4.1.2	OTP	41
4.1.3	Map	42
4.2	Account Details Page	43
4.3	Home Page	44
4.4	Search Page	45
5	Coding	46-281
5.1	Working (Front-End)	46-48
5.2	Drawable	48-97
5.3	Raw Module	97-101
5.4	Value Module	103-109
5.5	Working (Back-End)	109-142
5.6	Test Module Class	143-174
6	Bibliography	175

## **List of Abbreviations**

<b>S No.</b>	<b>Name of Abbreviations</b>	<b>Details of Abbreviations</b>	<b>Page</b>
1	CS	Computer Science	11
2	OS	Operating System	12
3	UPS	Universal Power Supply	17
4	MS word	Microsoft Word	18
5	DFD	Data Flow Diagram	23
6	ER	Entity Relationship	26
7	UCD	Use Case Diagram	27
8	SD	Sequence Diagram	28
9	UML	Unified Modeling Language	28

## LIST OF FIGURES

<b>Fig No.</b>	<b>Name of Figure</b>	<b>Page No.</b>
3.1	Flow Chart Symbol	22
3.2	Flow Chart Diagram	23
3.3	Data Flow Symbols	24
3.4	User	27
3.4.1	Level 0 DFD	27
3.4.2	Level 1 DFD	27
3.5	Admin	27-28
3.5.1	Level 0 DFD	27
3.5.2	Level 1 DFD	28
3.6	Employee	28-29
3.6.1	Level 0 DFD	28
3.6.2	Level 1 DFD	29
3.7	Payment	29-30
3.7.1	Level 0 DFD	29
3.7.2	Level 1 DFD	30
3.8	Entity Relationship Diagram	33
3.9	Use Case Diagram	36
3.10	Sequence Diagram	38
3.11	Collaborative Diagram	39
4.1	Input / Output (Screenshot)	40-42
4.4.1	Login	40
4.4.2	OTP	41
4.4.3	Map	42
4.2	Account Details Page	43
4.3	Home Page	44
4.4	Search Page	45



# CHAPTER 1

## INTRODUCTION

This is a web-oriented application that allows us to access all information about the college, staff, students, facilities etc. This application provides a virtual tour of Campus. Here we will get the latest information about the students and staff. This generic application is designed for assisting the students at an institute regarding information on the courses, exams, grades, and exam-timetable. It also provides support that a faculty can also check about his daily schedule, can create exams, prepare questions for Exams, and notices to the students. Here the administrator will manage the accounts of the student and faculties, check the overall performances of students, and upload the latest information about the campus.

We identify several problems including unauthorized privilege escalation, incorrect use of cryptography, vulnerabilities to network threats, and poor software development processes. We show that only college administrator can start the system administrator can search the particular student by his/her enrollment number or student id and we are adding notification module where administrator should add the notification with start and end date.

### 1.1 PROJECT OBJECTIVES

- **Simplify Grocery Shopping:** The primary objective of the app is to simplify the grocery shopping process for users. It aims to replace traditional paper lists or reliance on memory with a digital platform that allows users to create, organize, and manage their grocery lists conveniently.
- **Efficient List Creation:** The app aims to provide users with a user-friendly interface for creating and managing grocery lists. Users can easily add, edit, and delete items, ensuring an organized and customized shopping experience. Barcode Scanning Integration: The app integrates barcode scanning functionality, allowing users to scan product barcodes and automatically add items to their grocery lists. This feature saves time and improves accuracy when compiling shopping lists.

- **Inventory Management:** The app provides inventory management capabilities, enabling users to keep track of their existing grocery items at home. Users can update item quantities and receive notifications when items are running low, helping to avoid unnecessary purchases and reduce food waste.
- **Price Comparison and Offers:** The app includes price comparison features that allow users to compare prices of grocery items across different stores. It also provides access to exclusive offers, discounts, and promotions, helping users find the best deals and save money.
- **Personalized Recommendations:** Utilizing machine learning algorithms, the app analyzes user preferences and past purchase history to provide personalized recommendations for grocery items. This feature enhances the user's shopping experience by suggesting relevant products and helping them discover new items.
- **Recipe Integration:** The app integrates with recipe management, allowing users to import their favorite recipes. It then automatically generates grocery lists based on the required ingredients, simplifying meal planning, and ensuring users have all the necessary items.
- **User-Friendly Interface:** The app aims to provide a user-friendly interface with intuitive navigation and visually appealing design. It strives to offer a seamless and enjoyable experience for users, making the grocery management process more convenient and efficient.
- **Enhanced Efficiency and Organization:** Overall, the objective of the app is to enhance efficiency and organization in grocery management. By providing features that streamline the shopping process, track inventory, compare prices, and offer personalized recommendations, the app aims to make grocery shopping more efficient and organized for users.

## 1.2 PROJECT SCOPE

- **Grocery List Management:** The app allows users to create, edit, and manage multiple grocery lists. Users can add items, specify quantities, and mark items as purchased or completed. They can also categorize items based on departments or custom categories for better organization.
- **Barcode Scanning:** The app includes a barcode scanning feature that enables users to scan the barcodes of grocery items using their device's camera. Scanned items are automatically added to the grocery list, reducing manual data entry and improving accuracy.
- **Inventory Management:** Users can track their existing grocery items at home through the app's inventory management functionality. They can update item quantities, set expiration dates, and receive notifications when items are running low, helping them stay organized and avoid unnecessary purchases.
- **Price Comparison and Offers:** The app provides users with the ability to compare prices of grocery items across different stores. Users can view current prices, discounts, and offers from various retailers, allowing them to make informed purchasing decisions and save money.
- **Personalized Recommendations:** The app utilizes machine learning algorithms to provide personalized recommendations for grocery items based on user preferences and past purchase history. These recommendations help users discover new products and make their shopping experience more tailored to their needs.
- **Recipe Integration:** Users can import their favorite recipes into the app and generate grocery lists automatically based on the required ingredients. This feature simplifies meal planning and ensures that all necessary items are included in the grocery lists.
- **User Profiles and Settings:** The app allows users to create individual profiles and customize their settings according to their preferences. Users can set their preferred stores, units of measurement, notification preferences, and other relevant settings.
- **User-Friendly Interface:** The app aims to provide a user-friendly interface with intuitive navigation, clear visuals, and smooth interactions. It strives to offer a seamless and enjoyable user experience, making grocery management convenient and accessible for users of all levels of technical proficiency.

### **1.3 PROJECT DESCRIPTION**

The Grocery Management System Android app is a comprehensive solution designed to simplify and enhance the grocery shopping and management experience for users. With the app, users can create, organize, and manage their grocery lists, track their inventory, compare prices, receive personalized recommendations, and integrate their favorite recipes seamlessly.

The app begins by offering users a user-friendly interface where they can create and manage multiple grocery lists. They can easily add items to their lists, specify quantities, and categorize items based on departments or custom categories for better organization. Users can also mark items as purchased or completed, providing a clear overview of their shopping progress.

To streamline the process of adding items to the grocery list, the app integrates barcode scanning functionality. Users can simply use their device's camera to scan the barcodes of grocery items, and the app will automatically add them to the list. This feature saves time and eliminates manual data entry errors.

The app goes beyond list creation by providing inventory management capabilities. Users can keep track of their existing grocery items at home by updating quantities and setting expiration dates. The app also sends notifications when inventory items are running low, ensuring users never run out of essential items.

To help users find the best prices and save money, the app includes a price comparison feature. Users can compare prices of grocery items across different stores, view discounts, and offers. This empowers users to make informed purchasing decisions and find the most cost-effective options.

The Grocery Management System app incorporates machine learning algorithms to offer personalized recommendations for grocery items. By analyzing user preferences and past purchase history, the app suggests relevant products, helping users discover new items and tailor their shopping experience to their specific needs.

Additionally, the app integrates recipe management. Users can import their favorite recipes and automatically generate grocery lists based on the required ingredients. This feature simplifies meal planning and ensures that all necessary items are included in the grocery lists, making it easier for users to prepare their meals.

## **1.4 HARDWARE / SOFTWARE**

### **Software Requirements**

- **O S** – Android OS
- **Language** – Android Studio
- **Database** – MySQL Server 2019
- **Frontend** – Kotlin, XML
- **Backend** – PHP, Microsoft MySQL Server 2019

### **Hardware Requirements**

- **Processor** - Intel Core i5 (6<sup>th</sup> GEN) i5-7009U Dual-Core(3.0 GHz) and above.
- **RAM** - 8 GB (DDR4L-3600) and above
- **Storage** - 500 GB HDD
- **Monitor** - 15” Color Monitor
- **Keyboard** - 122 Keys

## **1.5 FUNCTIONAL REQUIREMENTS**

### **Ordering System**

- Create an Account
- Log into the Website
- Navigate the DAD menu.
- Select an item from the menu
- Customized options for selected item
- Add item to current service.
- Remove items/ remove all items from current service.

### **Menu Management System**

- Add a new/update/delete client to/from the menu
- Add a new/update/delete products category to/from the menu
- Add a new/update/delete products item to/from the menu
- Add a new/update/delete option for a given products item.
- Update price for a given products item
- Update additional information for a given products item

### **Order Retrieval System**

- Retrieve new services from the database.
- Display the services in an easily readable, graphical way.
- Mark and service as having been processed and remove it from the list of active services.

## 1.6 NON – FUNCTIONAL REQUIREMENTS

- **Portability**

System running on one platform can easily be converted to run on another platform.

- **Reliability**

The ability of the system to behave consistently in a user-acceptable manner when operating within the environment for which the system was intended.

- **Availability**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs.

- **Maintainability**

A commercial database is used for maintaining the database and the application server takes care of the site.

- **Security**

Secure access of confidential data

## **CHAPTER 2**

### **FEASIBILITY STUDY**

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called feasibility study. This type of study determines if a project can and should be taken.

Once it has been determined that a project is feasible, the analyst can go ahead and prepare the project specification which finalizes project requirements. Generally, feasibility studies are undertaken with tight time constraints and normally constraints and normally culminate in a written and oral feasibility report.

The contents and recommendations of such a study will be used as a sound basis for deciding whether to proceed, postpone or cancel the project, thus, since the feasibility study may lead to the commitment of large resource, it becomes necessary that it should be conducted completely and that no fundamental errors or judgments are made.

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of project and the usefulness. Thus, when a new application is proposed normally goes through a feasibility study before it's approved of development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic, and operational feasibilities.

In the conduct of feasibility studied the following feasibility:

1. Technical Feasibility
2. Operation Feasibility
3. Behaviorally Feasibility
4. Economical Feasibility



## 1.7 TECHNICAL FEASIBILITY

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirements in terms of input, output, programs, and procedures.

Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

The technical needs of the system may vary considerably but might include:

- The facility to produce outputs in given time
- Response time under output conditions
- Ability to process a certain volume of at a particular speed
- Facility to communicate data to distant location
- Technical feasibility centers on the existing computer system (hardware, software, etc. and to extent it can support the proposed addition.
- Thus, in this aspect the main point considered are:
  - i. Can the work for project can be done with current transaction? equipment, existing Software technology and available personnel.
  - ii. If new technology is needed what is the likelihood that it can be developed?

The following are the proposed hardware and software configurations:

- A computer with the proposed configuration
- UPS is required for secured working
- Printer for printing various types of quotation

### **App Development:**

- **Android Platform:** Assess the compatibility of the app with the Android operating system, considering factors such as target Android versions and device compatibility.
- **Development Tools and Technologies:** Evaluate the availability of appropriate development tools, frameworks, and programming languages to develop the app efficiently and effectively.

- **Skillsets and Expertise:** Determine if the development team possesses the required skills and expertise to develop an Android app with the desired features. Assess the need for additional resources or training.
- **Barcode Scanning:** Evaluate the availability of barcode scanning libraries or APIs that can be integrated into the app. Ensure compatibility with different barcode formats and optimal performance across a range of devices.
- **Price Comparison:** Assess the availability of reliable and up-to-date price comparison data sources or APIs. Consider the technical aspects of retrieving and displaying price information from various stores.
- **Personalized Recommendations:** Determine the feasibility of implementing machine learning algorithms or recommendation systems to analyze user preferences and provide personalized recommendations. Assess the availability of relevant data sources and the computational resources required for recommendation generation.
- **Data Sources and APIs:** Evaluate the availability and accessibility of data sources such as product information, store prices, and offers. Assess the feasibility of integrating with these data sources through APIs or other methods.
- **Recipe Integration:** Determine the feasibility of integrating recipe management functionality and the ability to import and parse recipe data. Ensure compatibility with common recipe formats and the availability of relevant recipe databases or APIs.
- **Performance Requirements:** Define the expected performance metrics of the app, such as response time, data processing speed, and smooth user interface interactions. Assess if the technical infrastructure can meet these performance requirements.
- **Scalability:** Consider the potential growth and user base of the app. Evaluate if the technical architecture can handle increasing user loads and accommodate future enhancements and updates.
- **Data Security:** Assess the technical measures needed to ensure the security of user data, including encryption, secure authentication, and data storage practices.
- **Privacy Compliance:** Evaluate the technical feasibility of complying with data privacy regulations, such as obtaining user consent, anonymizing data, and providing privacy setting.

## 1.8 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned into information. Systems that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed. It is mainly related to human organization and political aspects.

The points to be considered are:

- What changes will be brought with the system?
- What organizational structures are distributed?
- What new skill will be required?

Do the existing staff members have these skills? If not, can they be trained in due course of time? Generally, a project will not be rejected simply because of operational in feasibility but such consideration is likely critically affecting the nature and scope of the eventual recommendations.

This feasibility study is carried out by a small group of people who are familiar with information systems techniques, who understand the parts of the business that are relevant to the project and are skilled in system analysis and design process.

The operational feasibility is mainly related to human organizational and political aspects. My project "Automation of Pathology" is operationally feasible due to following reasons: -

- No existing organizational structures are disturbed.
- Basic knowledge of operational computer will be required for the end users, and it was that personnel of library were familiar with the basic operation of the computer.

Because they had little knowledge of computer on specifics like Ms-word etc. so there was only requirement of training to operate this software. It is very easy to train them within a small duration of time.

## **User Acceptance:**

- **User Needs and Preferences:** Conduct surveys or interviews with potential users to understand their needs, preferences, and pain points related to grocery shopping and management. Assess if the app's features align with user expectations and if it provides sufficient value to encourage adoption.
- **User Interface and User Experience:** Evaluate the app's user interface design and user experience to ensure it is intuitive, visually appealing, and easy to navigate. Consider usability testing to identify any usability issues and make necessary improvements.
- **Integration with Existing Systems:** Assess the compatibility and integration requirements of the app with existing systems or processes within the organization or with external systems such as retail stores or inventory management systems.
- **Change Management:** Determine the level of organizational change required to implement the app successfully. Identify any resistance to change and develop strategies to address it effectively.
- **Technical Resources:** Evaluate the availability of technical resources required for app implementation and operation, such as hardware infrastructure, software licenses, and network connectivity. Assess if the organization has the necessary resources or if additional resources need to be acquired.
- **Personnel:** Determine if the organization has the required personnel or skills to manage and support the app's operation. Identify any gaps in skills or resource allocation and plan for necessary training or recruitment.
- **App Updates and Upgrades:** Consider the need for regular updates, bug fixes, and feature enhancements to ensure the app remains relevant and functional over time. Assess the feasibility of managing these updates and upgrades efficiently.
- **Customer Support:** Evaluate the organization's capacity to provide customer support, handle user inquiries, and address technical issues promptly. Determine if additional support resources are needed.
- **User Growth:** Consider the potential growth in the user base of the app and assess if the operational infrastructure can handle increased user loads without compromising performance and user experience.
- **Feature Enhancements:** Evaluate the feasibility of adding new features and functionalities to the app in the future, based on user feedback and market trends.

## 1.9 BEHAVIORAL FEASIBILITY

Behavioral Feasibility is the measure of how society is looking towards our project, what is the reaction of people who are going to use this in the upcoming future.

It includes how strong the reaction of user will be towards the development of new system that involves computer's use in their daily life for maintaining digital records of Grocery Management System.

This includes the following questions:-

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it specifies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

### User Adoption:

- **User Needs and Preferences:** Conduct market research and user surveys to identify the specific needs, preferences, and behaviors of the target user base. Understand their current grocery shopping habits and determine if the app aligns with their expectations and requirements.
- **User Training and Familiarity:** Assess the level of training or learning curve required for users to effectively use the app. Evaluate if the app's user interface and features are intuitive enough for users with varying levels of technical proficiency.
- **User Interface Design:** Ensure that the app's user interface is visually appealing, easy to navigate, and provides a pleasant user experience. Consider incorporating gamification elements or rewards to encourage user engagement and sustained app usage.

- **Push Notifications and Reminders:** Assess the feasibility of implementing push notifications and reminders to keep users engaged with the app and remind them of important tasks, such as creating grocery lists or taking advantage of offers.
- **Management Support:** Evaluate the support and commitment of key stakeholders, such as management or store owners, in promoting and implementing the app. Assess their willingness to invest resources and support the necessary changes in operational processes.
- **User Feedback and Involvement:** Establish channels for users to provide feedback and suggestions for app improvement. Involve users in the app development process through beta testing or user feedback sessions to ensure their needs and preferences are considered.
- **Organizational Culture:** Assess the organization's readiness and willingness to adopt new technologies and processes. Identify potential resistance to change and develop change management strategies to address concerns and promote acceptance.
- **Communication and Training:** Develop effective communication and training plans to introduce the app to users and stakeholders. Provide clear instructions, tutorials, and support materials to facilitate user onboarding and adoption.

- **Data Privacy:** Address users' concerns regarding data privacy and security. Ensure that the app complies with relevant data protection regulations and implements robust security measures to protect user data.
- **Transparency:** Communicate the app's privacy policy and data handling practices clearly to users. Gain their trust by being transparent about data collection, usage, and sharing practices.

## 1.10 ECONIMICAL FEASIBILITY

Economical is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost or benefit analysis, the procedure is to determine the benefits and saving that are expected from a proposed system and compare with cost.

It benefits out weight costs a decision taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves accuracy at each phase of the system's life cycle.

An evaluation of development cost weighed against the ultimate income of benefit derived from the development system or project among the most important information contained in feasibility study is cost benefit analysis an assessment of the economic justification for a computer-based system project.

The benefits of a project include four types:

- Cost saving benefits.
- Cost avoidances benefits.
- Improved service level benefits.
- Improved the information benefits.
- The cost of the hardware and software.
- The costs conduct a full system investigation.
- The benefits come in the form of reduced costs or fewer costly errors.

Cost-saving benefits of our projects lead to the reduction in administration & operational costs. A cost avoidance benefits our project does not require future and additional staff and also reduces any future operational cost. This project leads the quicker and enhanced administrative decision thus making improved information benefits.

### **Cost Analysis:**

- **Development Costs:** Evaluate the expenses associated with developing the app, including software development, hardware requirements, licenses, and any third-party integrations. Consider the cost of hiring developers or outsourcing development services.
- **Maintenance Costs:** Assess the ongoing costs of maintaining and updating the app, including bug fixes, feature enhancements, server hosting, and technical support.
- **Operational Costs:** Consider the costs of operating the app, such as server infrastructure, data storage, and marketing expenses.
- **Cost Savings:** Identify potential cost savings that the app can offer to users, such as reducing impulse purchases, avoiding duplicate purchases, and minimizing food waste through inventory management. Estimate the monetary value of these savings.
- **Efficiency Gains:** Determine the time savings and increased efficiency that the app can provide to users in terms of creating and managing grocery lists, comparing prices, and generating recipe-based shopping lists.
- **Revenue Generation:** Identify potential revenue streams for the app, such as in-app purchases, premium features, partnerships with retailers, or targeted advertisements. Assess the market demand and competition to estimate the app's revenue-generating potential.
- **Payback Period:** Calculate the expected payback period, which is the time required to recoup the initial investment through cost savings or revenue generation.
- **Net Present Value (NPV):** Conduct a financial analysis to determine the NPV, which assesses the profitability of the project by considering the time value of money.
- **Return on Investment (ROI):** Estimate the ROI by comparing the expected financial gains from the app against the initial investment. Assess if the ROI meets the organization's or stakeholders' expectations.
- **Market Size and Potential:** Assess the size of the target market for the app and estimate the potential number of users. Consider demographic factors, such as smartphone penetration and grocery shopping habits, to evaluate the market opportunity.



- **Competitive Landscape:** Analyze the competition in the grocery management app market. Identify competitors' pricing models, features, and market positioning. Determine if the app can differentiate itself and capture a significant market share.
- **Financial Risks:** Identify potential financial risks, such as underestimated development costs, lack of revenue generation, or high ongoing operational costs. Develop risk mitigation strategies to address these risks.
- **Market Risks:** Evaluate the market demand and potential adoption challenges. Assess if there are any external factors, such as changing consumer behaviors or new market entrants, that could impact the app's success. Incurred by our system will include.

# CHAPTER 3

## DATABASE DESIGN

Database design can be defined as a collection of tasks or processes that enhance the designing, development, implementation, and maintenance of enterprise data management system. Designing a proper database reduces the maintenance cost thereby improving data consistency and the cost-effective measures are influenced in terms of disk storage space.

Therefore, there has to be a brilliant concept of designing a database. The designer should follow the constraints and decide how the elements correlate and what kind of data must be stored.

Database design for the Grocery Management System Android app involves structuring and organizing the data in a way that efficiently supports the app's functionalities and requirements.

**Here are some considerations for the database design:-**

- **Identify Entities and Attributes:** Determine the entities that need to be stored in the database, such as users, grocery items, stores, recipes, and shopping lists. Define the attributes for each entity, such as user ID, name, email, password, item name, quantity, price, store name, recipe name, ingredients, etc.
- **Establish Relationships:** Analyze the relationships between different entities. For example, a user can have multiple shopping lists, a shopping list can have multiple grocery items, and a recipe can have multiple ingredients. Define the types of relationships, such as one-to-one, one-to-many, or many-to-many, and implement them using appropriate database techniques (e.g., foreign keys).

- **Normalize the Data:** Apply normalization techniques to ensure data integrity and eliminate redundancy. Normalize the database by breaking down entities into smaller tables and reducing data duplication. Identify and eliminate any functional dependencies and anomalies that could lead to data inconsistencies.
- **Determine Data Storage:** Decide on the appropriate data types for each attribute based on the nature of the data (e.g., integers, strings, dates, Booleans). Consider storage requirements for images or other media associated with grocery items or recipes.
- **Design Database Schema:** Create a database schema that represents the entities, attributes, relationships, and constraints. Use an appropriate database management system (DBMS) for the app, such as MySQL, SQLite, or Firebase Realtime Database. Define primary keys, foreign keys, indexes, and constraints to ensure data integrity and improve query performance.
- **Optimize Query Performance:** Analyze the app's data access patterns and design the database schema and indexes to optimize query performance. Consider the most frequent and critical operations, such as retrieving shopping lists, updating item quantities, or searching for specific grocery items.
- **Implement Security Measures:** Implement security measures to protect the database and sensitive user data. This may include encrypting passwords, using parameterized queries to prevent SQL injection attacks, and ensuring proper access controls.
- **Plan for Scalability:** Anticipate the potential growth and increased data volume in the future. Design the database to handle increased user loads and additional features without sacrificing performance. Consider database scaling options, such as sharing, replication, or cloud-based solutions, to accommodate future growth.
- **Backup and Recovery:** Develop a backup and recovery plan to protect the database from data loss. Regularly back up the database and implement recovery mechanisms in case of failures or disasters.
- **Test and Refine:** Test the database design to ensure that it meets the app's requirements and performs well under different scenarios. Refine the design based on testing results and user feedback.

## 3.1 DATABASE TABLE

Based on the Grocery Management System Android app, here are some suggested database tables:

- **Users:**

user\_id (Primary Key)

name

email

password

address

phone\_number

- **Stores:**

store\_id (Primary Key)

name

address

phone\_number

- **GroceryItems:**

item\_id (Primary Key)

name

price

description

image\_url

- **ShoppingLists:**

list\_id (Primary Key)

user\_id (Foreign Key referencing Users table)

name

creation\_date

- **ShoppingListItems:**

list\_item\_id (Primary Key)

list\_id (Foreign Key referencing ShoppingLists table)

item\_id (Foreign Key referencing GroceryItems table)

quantity

- **Recipes:**

recipe\_id (Primary Key)

name

description

image\_url

- **RecipeIngredients:**

recipe\_ingredient\_id (Primary Key)

recipe\_id (Foreign Key referencing Recipes table)

item\_id (Foreign Key referencing GroceryItems table)






quantity

## 3.2 FLOW CHART DIAGRAM

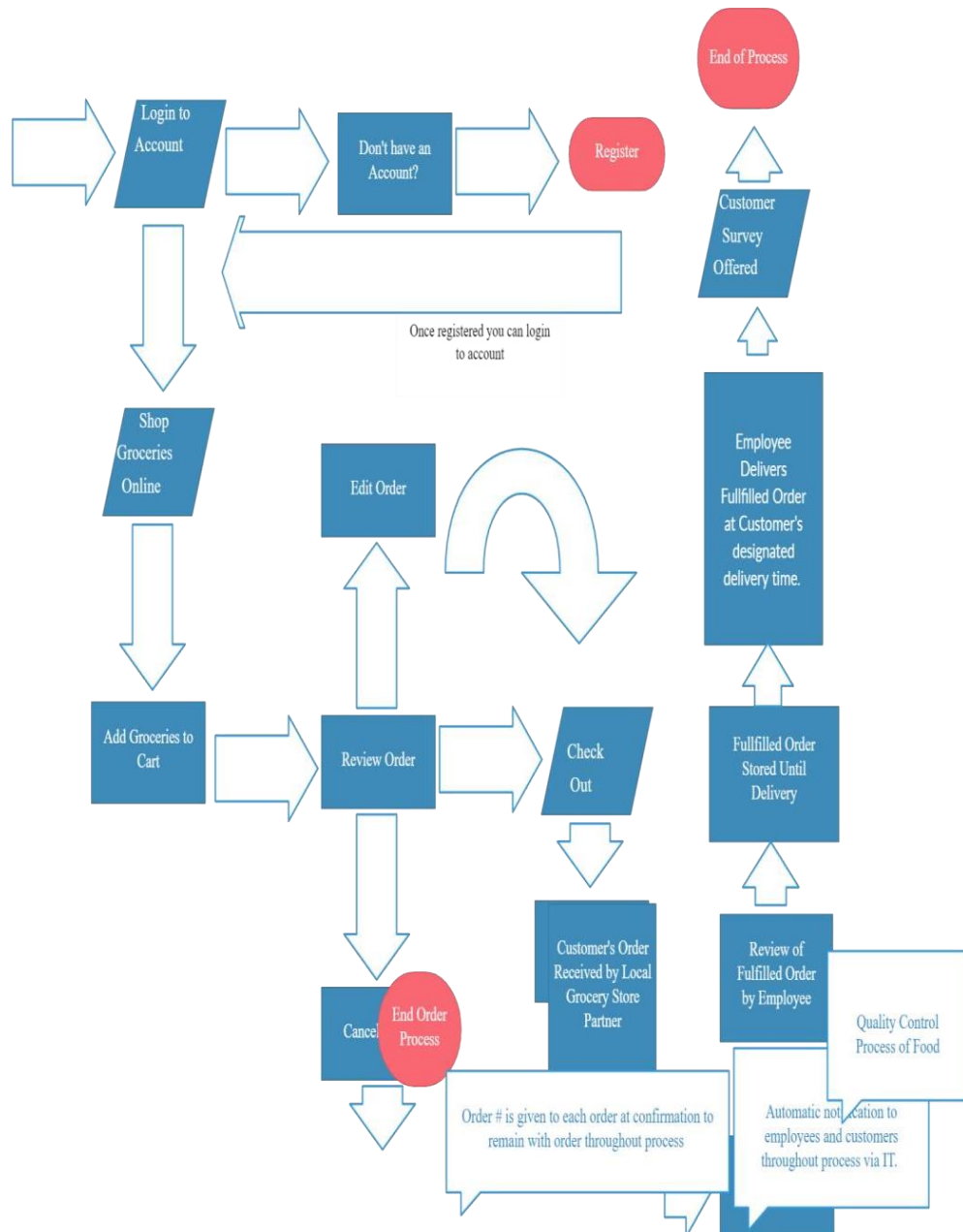
A flowchart is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted within a diagram shape.

Steps are linked by connecting lines and directional arrows. This allows anyone to view the flowchart and logically follow the process from beginning to end.

A flowchart is a powerful business tool. With proper design and construction, it communicates the steps in a process very effectively and efficiently.

Symbol	Name	Function
	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision

**Fig. 3.1 :** Flow Chart Symbols




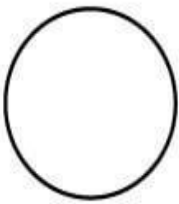
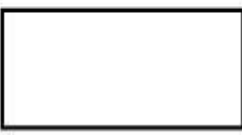
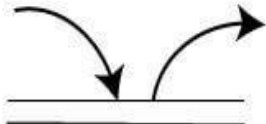
**Fig. 3.2 : Flow Chart Diagram**

### 3.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called a data flow graph or bubble chart.

Symbol	Name	Function
	Data flow	Used to Connect Processes to each other, to sources or Sinks; the arrow head indicates direction of data flow.
	Process	Performs Some transformation of Input data to yield output data.
	Source of Sink (External Entity)	A Source of System inputs or Sink of System outputs.
	Data Store	A repository of data; the arrow heads indicate net inputs and net outputs to store.

**Fig. 3.3 : Data Flow Symbols**



The DFD takes an input-process-output view of a system i.e., data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software.

Data objects represented by labeled arrows and transformation are represented by circles also called bubbles. DFD is presented in a hierarchical fashion i.e., the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater levels of details, the analyst performs an implicit functional decomposition of the system. At the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

A context-level DFD for the system the primary external entities produce information for use by the system and consume information generated by the system. The labeled arrow represents data objects or object hierarchy.

The DFD may be used to perform a system or software at any level of abstraction. Infact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily two levels in the data flow diagram, which are: 0-level DFD and 1-level DFD.

#### **RULES FOR DFD:**

- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions.
- Reads left to right and top to bottom.
- Identify all inputs and outputs.
- Identify and label each process internal to the system with Rounded circles.
- A process is required for all the data transformation and Transfers.

Therefore, never connect a data store to a data Source or the destinations. or another data store with just a Data flow arrow.

- Do not indicate hardware and ignore control information.
- Make sure the names of the processes accurately convey everything the process is done.
- There must not be an unnamed process.
- Indicate external sources and destinations of the data, with Squares.
- Number each occurrence of repeated external entities.
- Identify all data flows for each process step, except simple Record retrievals.
- Label data flows on each arrow.
- Use details flow on each arrow.

### **0-level DFD**

The Level-0 DFD, also called context diagram of the result management system is shown in fig. As the bubbles are decomposed into less and less abstract bubbles, the corresponding data flow may also need to be decomposed.

### **1-level DFD**

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.

## 1) User

### 0-level DFD

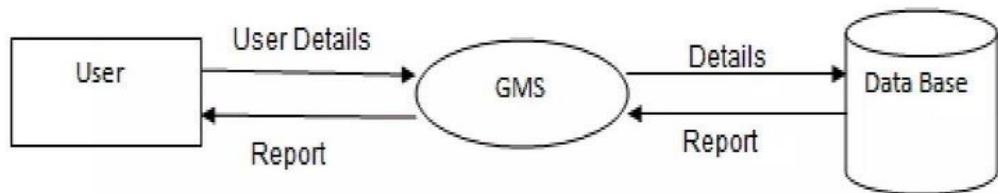


Fig. 3.4.1 : Level 0 DFD

### 1-level DFD

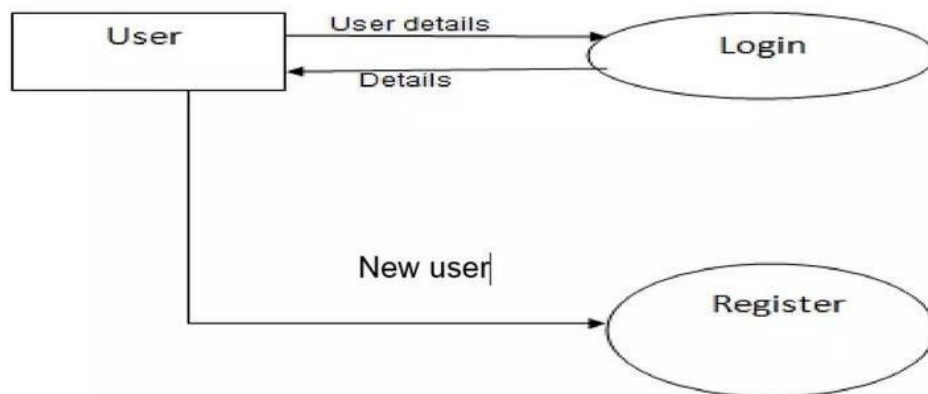


Fig. 3.4.2 : Level 1 DFD

## 2) Admin

### 0-level DFD

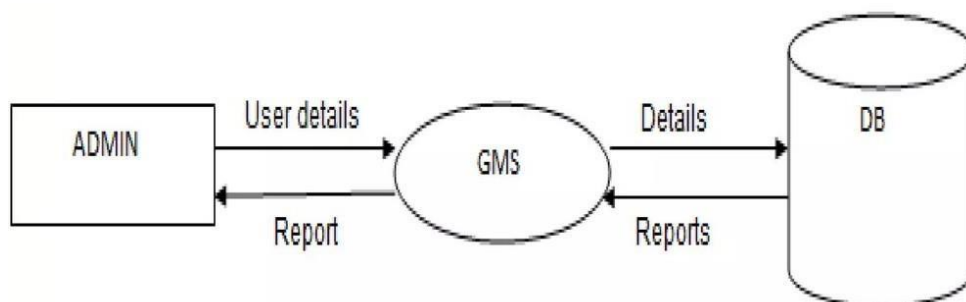


Fig. 3.5.1 : Level 0 DFD

### 1-level DFD

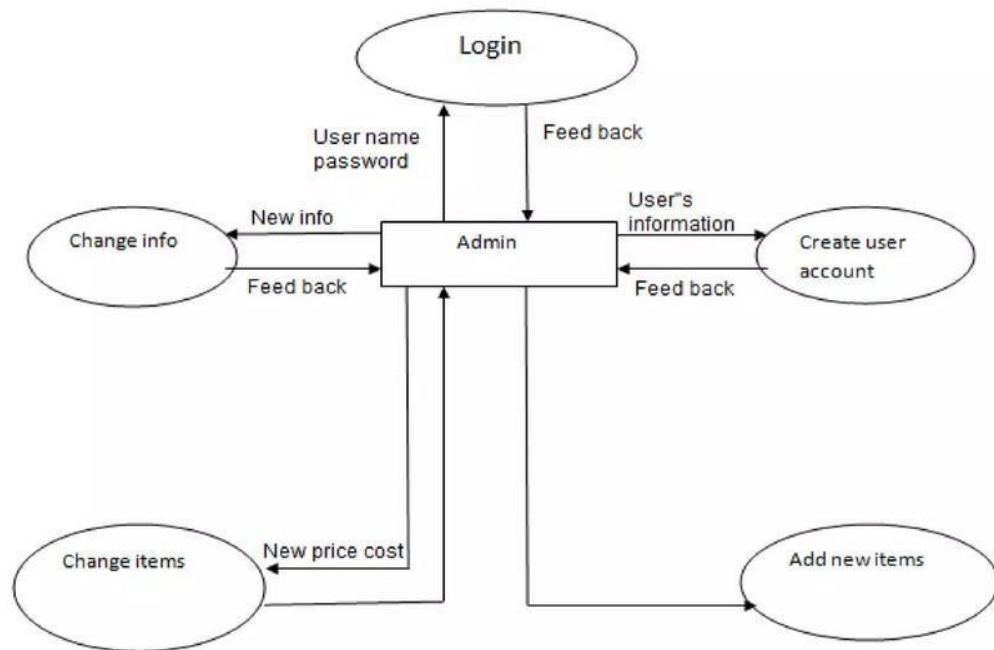


Fig. 3.5.2 : Level 1 DFD

### 3) Employee

#### 0-level DFD

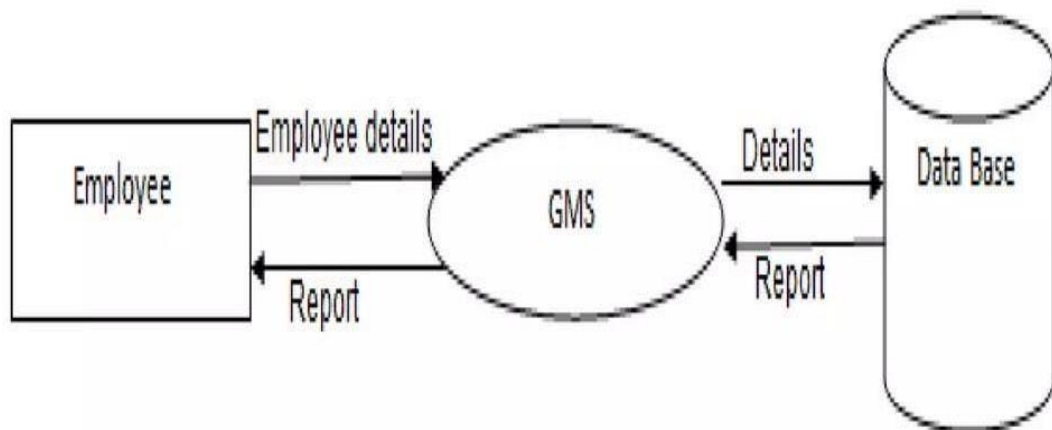


Fig. 3.6.1 : Level 0 DFD

### 1-level DFD

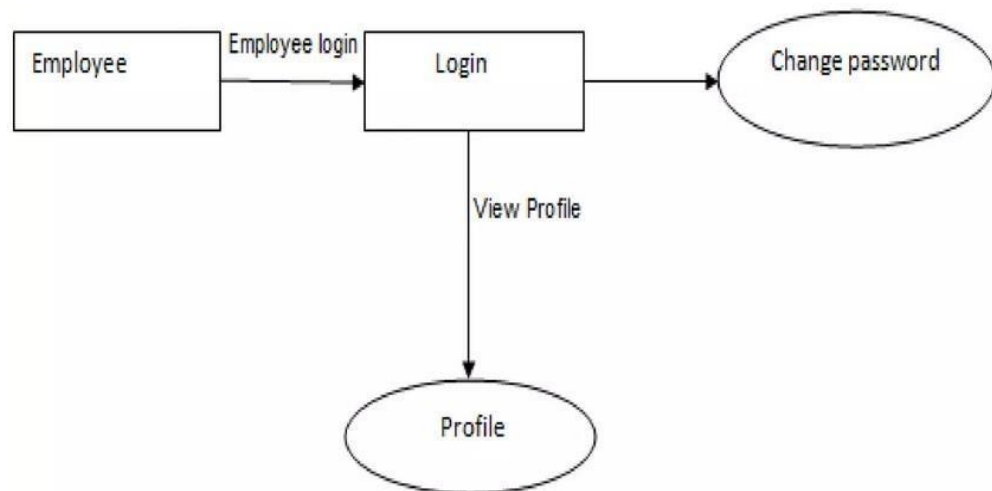


Fig. 3.6.2 : Level 1 DFD

### 4) Payment

#### 0-level DFD

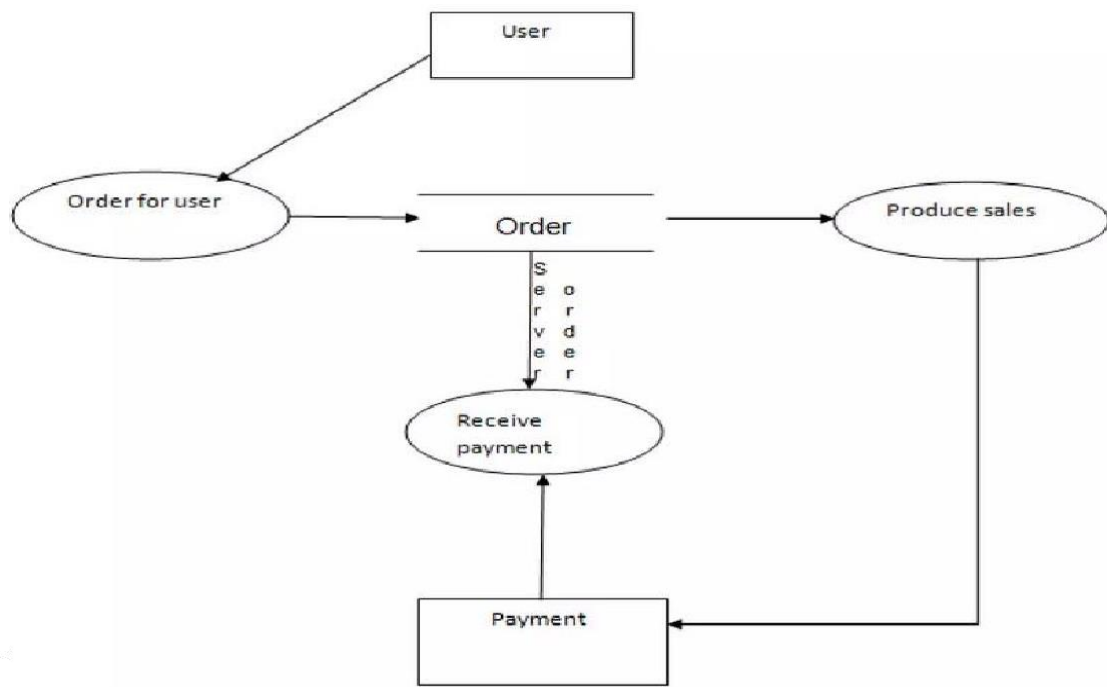
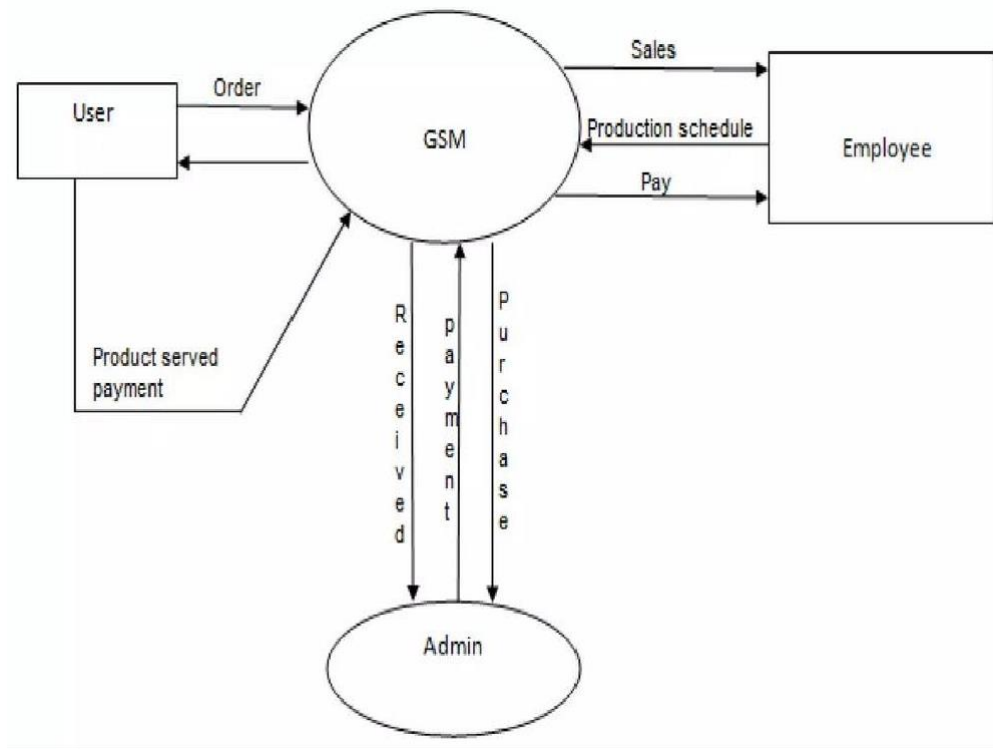


Fig. 3.7.1 : Level 0 DFD

### 1-level DFD



**Fig. 3.7.2 : Level 1 DFD**

### 3.3 ENTITY RELATIONSHIP DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects.

Since Chen wrote his paper the model has been extended and today it is commonly used for database design. For the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.

- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in a specific database management software.

Entity-Relationship model stands for an ER model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system. It develops a conceptual design for the database. It also develops a very simple and easy to design view of data. In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

### **Connectivity and Cardinality**

The basic types of connectivity for relations are: one-to-one, one-to-many, and many-to-many. A one-to-one (1:1) relationship is when at most one instance of an entity A is associated with one instance of entity B. For example, "employees in the company are each assigned their own office. For each employee there exists a unique office and for each office there exists a unique employee.

A *one-to-many* (1:N) relationship is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A. An example of a 1:N relationships is a department that has many employees each employee is assigned to one department.

A many-to-many (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A. The connectivity of a relationship describes the mapping of associated.

## ER Notation

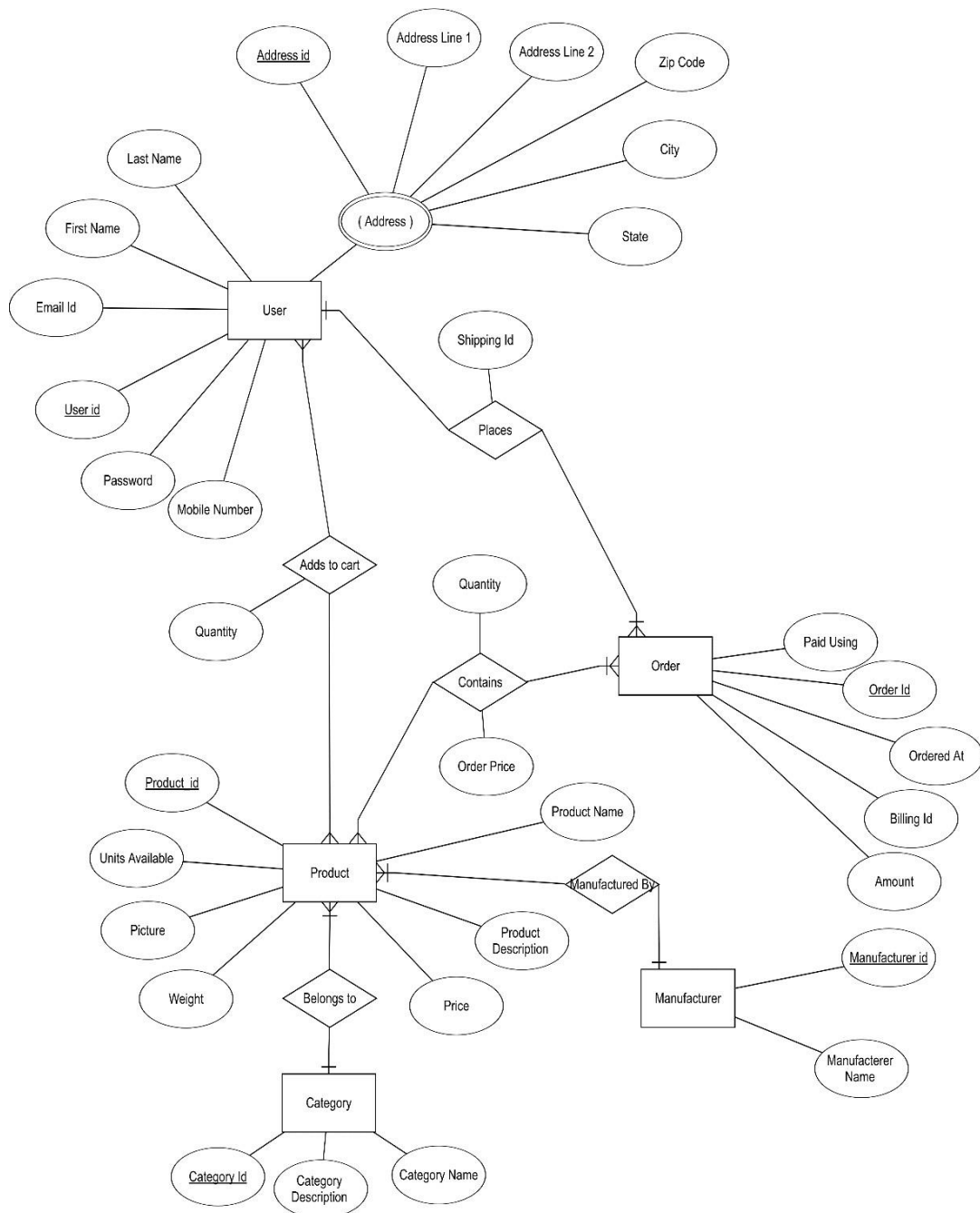
There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used, among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin.

The symbols used for the basic ER constructs are:

- **Entities:** They are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- **Relationships:** They are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.
- **Attributes:** when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
- **Cardinality:** It is of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.
- **Existence:** It is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.





**Fig. 3.8 : Entity Relationship Diagram**

### 3.4 USE CASE DIAGRAM

A use case diagram (UCD) is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. A use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

A Use case is a description of set of sequence of actions Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object.

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirements, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

- **User:**

**Register:** The user can create a new account in the app by providing their details.

**Login:** The user can log in to their account using their credentials.

**Manage Profile:** The user can view and update their profile information.

**Create Shopping List:** The user can create a new shopping list by specifying a name and adding grocery items.

**Manage Shopping List:** The user can view, edit, and delete their existing shopping lists.

**Browse Grocery Items:** The user can search and browse the available grocery items in the app's database.

**View Recipes:** The user can browse and view a collection of recipes.

**Logout:** The user can log out from their account.

- **Admin:**

**Manage Grocery Items:** The admin can add, edit, and delete grocery items from the app's database.

**Manage Stores:** The admin can add, edit, and delete stores from the app's database.

- **System:**

**Generate Shopping List:** The system generates a shopping list based on the user's input, including selected grocery items and quantities.

**Notify User:** The system sends notifications to the user for various events, such as new recipes, offers, or reminders.

- **External Systems:**

**Payment Gateway:** The app integrates with a payment gateway to facilitate secure online transactions.

**Recipe API:** The app may use an external Recipe API to fetch and display recipe information.



**Fig. 3.9 : Use Case Diagram**

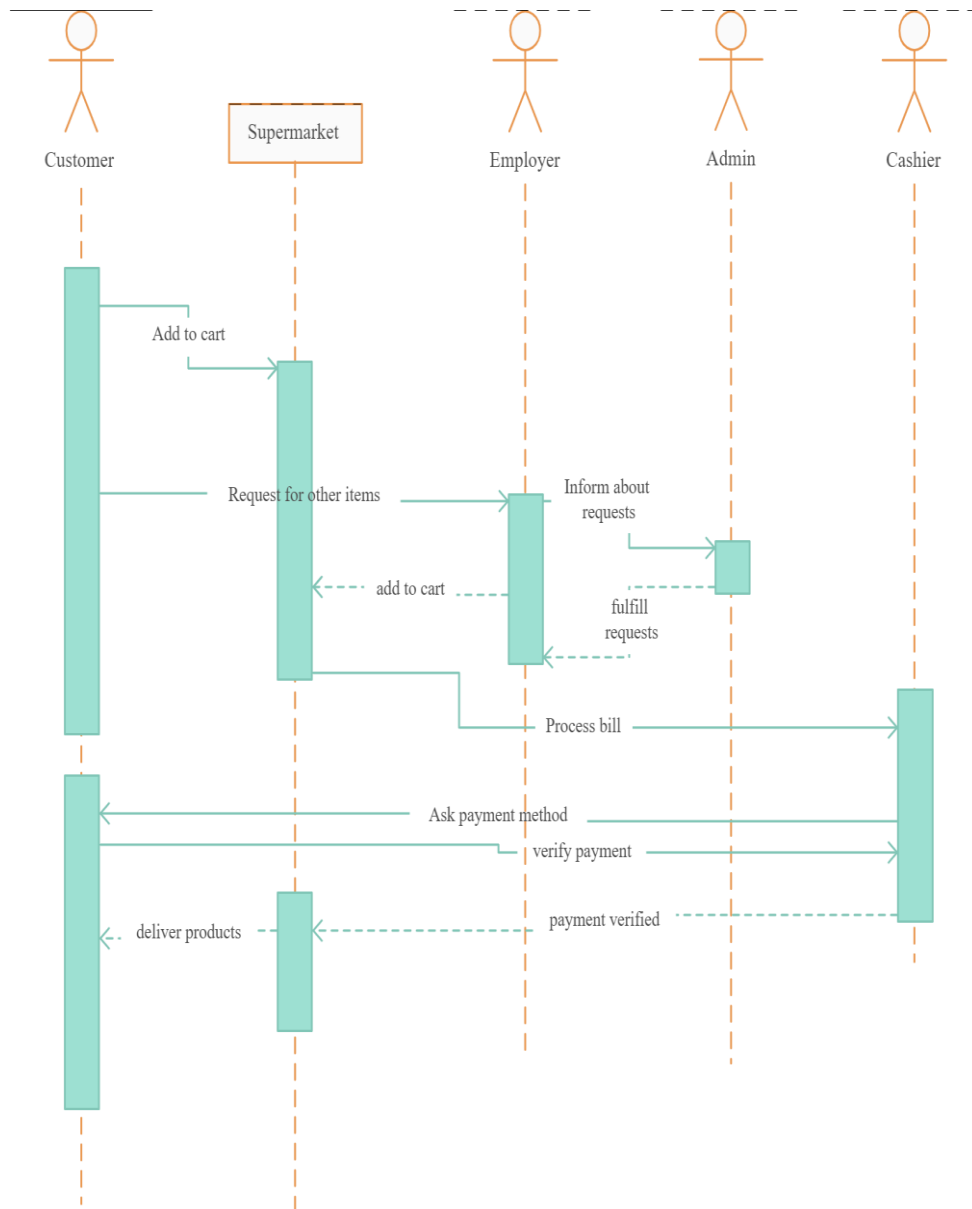
### 3.5 SEQUENCE DIAGRAM

The sequence diagram (SD) represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time.

In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

Now, the sequences will be as :-

- The user opens the app and selects the option to create a new shopping list.
- The app presents a form where the user can enter the name of the shopping list.
- The user enters the name of the shopping list and submits the form.
- The app sends a request to the server to create the shopping list.
- The server receives the request and verifies the user's authentication credentials.
- Upon successful verification, the server creates a new entry in the database for the shopping list.
- The server sends a response to the app confirming the successful creation of the shopping list.
- The app updates the user interface to reflect the newly created shopping list.
- The user can now proceed to add grocery items to the shopping list.



**Fig. 3.10 : Sequence Diagram**

### 3.6 COLLABORATIVE DIAGRAM

To create a Collaboration Diagram, you can follow these steps:

Identify the main components or objects in your system. For example, User, ShoppingList, GroceryItem, Store, Recipe, etc. Determine the interactions or messages exchanged between these objects. For example, User creates a ShoppingList, ShoppingList adds GroceryItems, User views ShoppingList, etc.

Arrange the objects on the diagram and connect them with lines representing the interactions or messages. Add labels to the lines to indicate the purpose of the communication. Consider adding additional details such as object attributes or method invocations, if necessary, to provide more clarity and context.

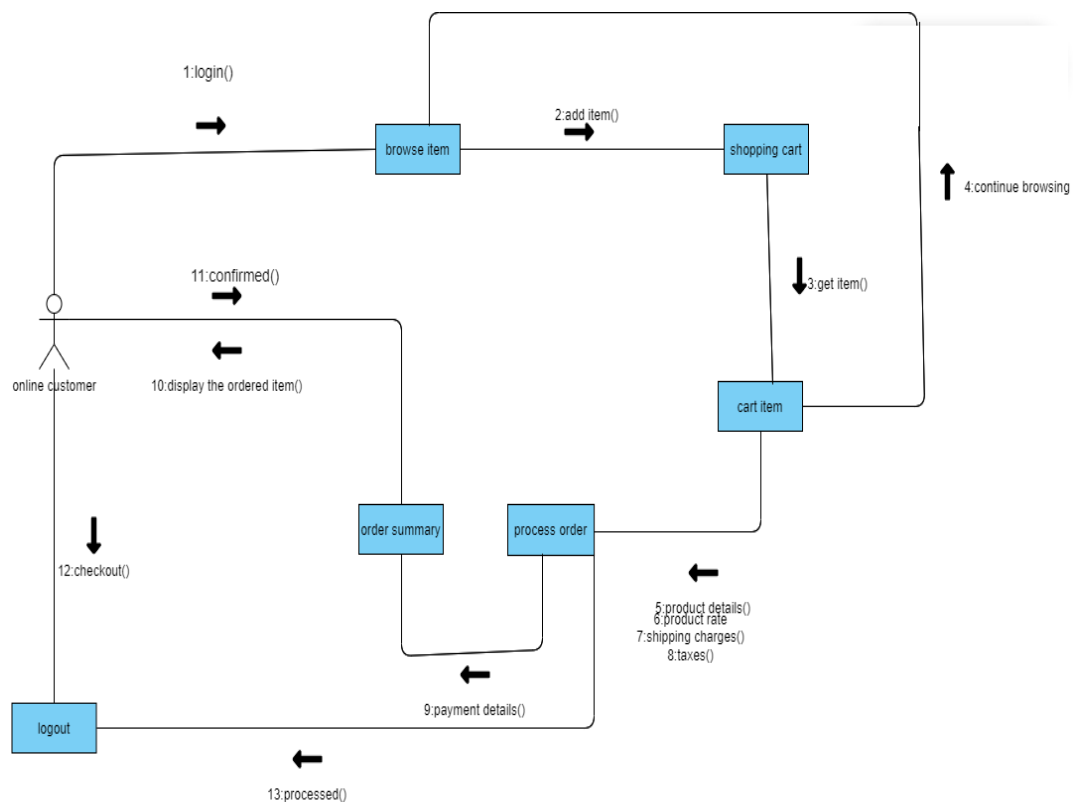


Fig: 3.11 Activity Diagram

# CHAPTER 4

## FORM DESIGN

### 4.1 INPUT / OUTPUT PAGE (SCREENSHORT)

Login:

13:53 82%

× Update Profile

Add Profile Picture

Name  
Amit

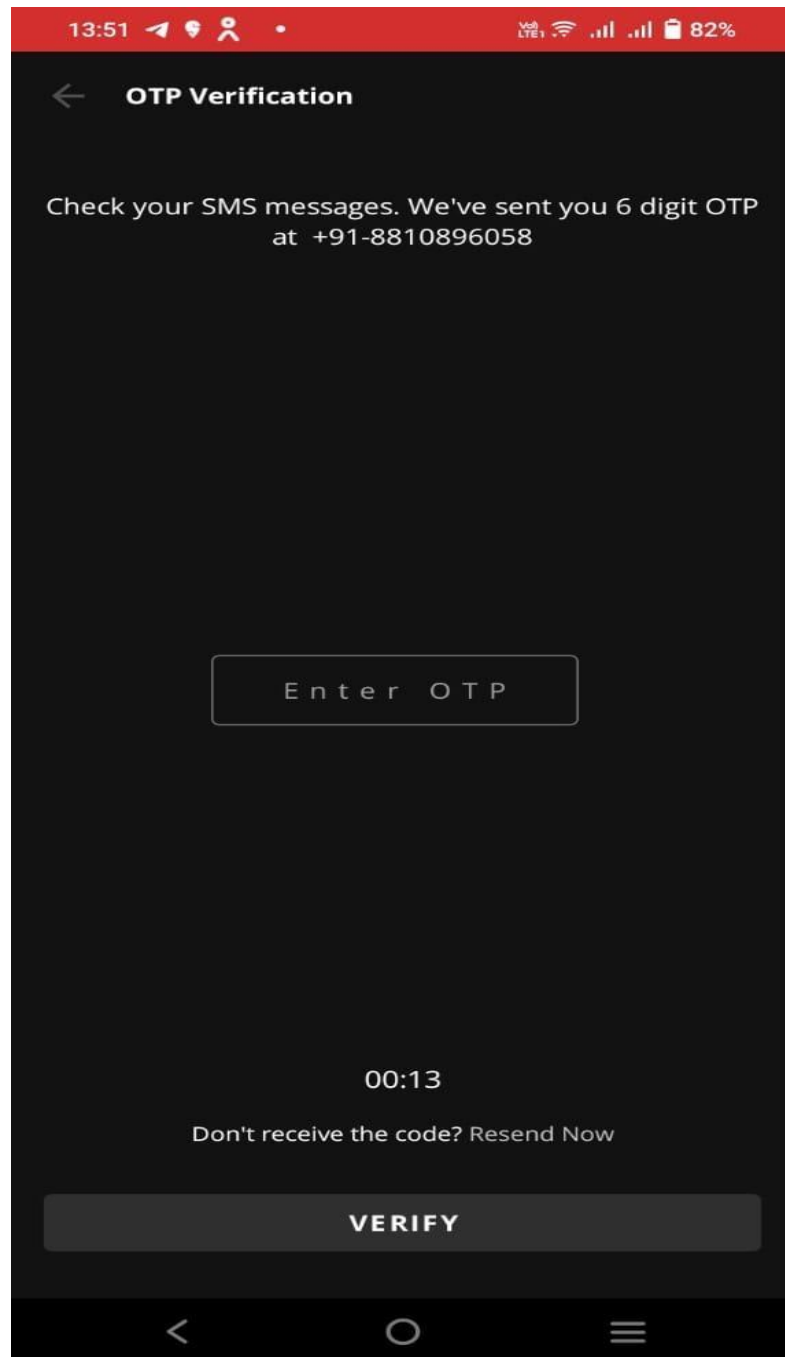
Email Address  
amitaj2128@gmail.com

SAVE CHANGES

Fig. 4.1.1 : Login Page



**OTP:**



**Fig. 4.1.2 : OTP Page**

MAP:

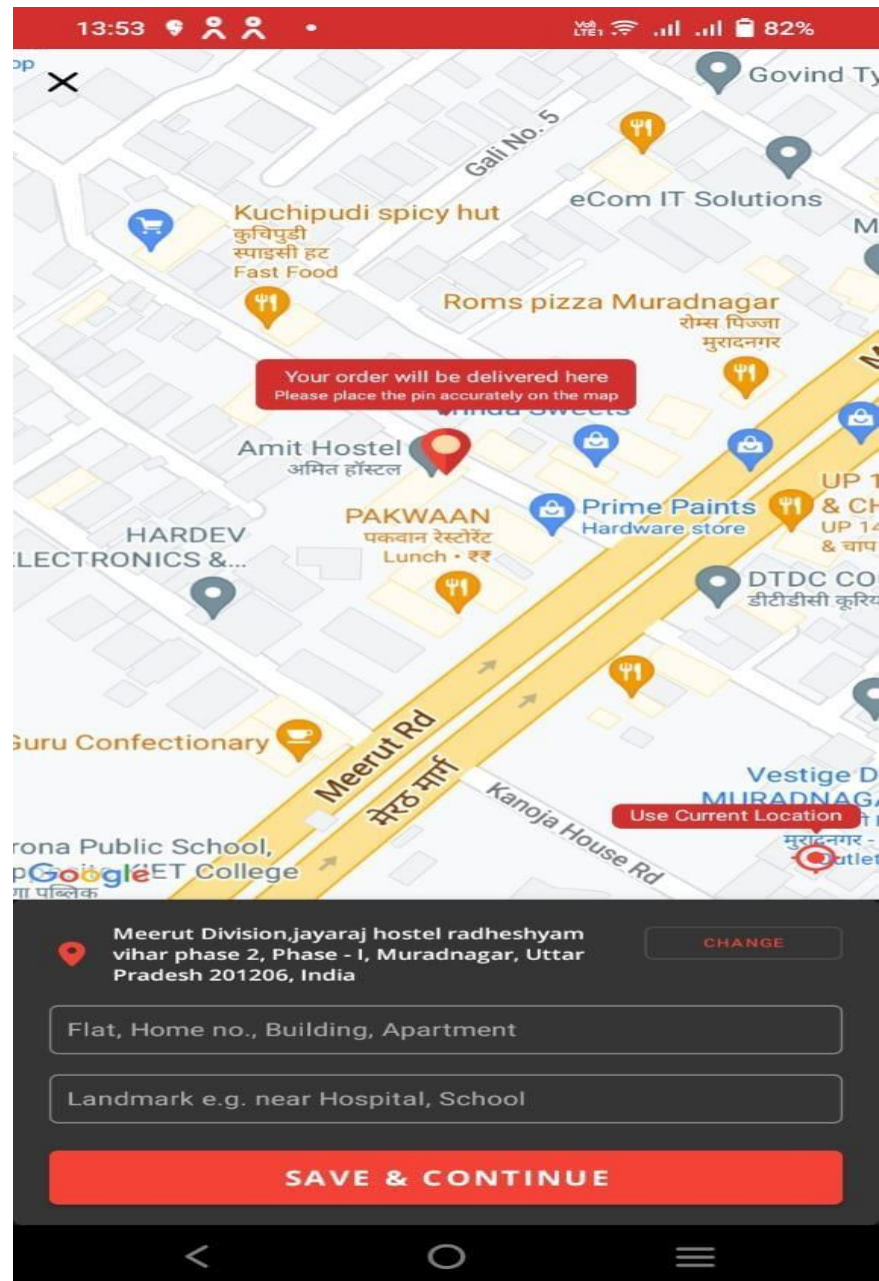
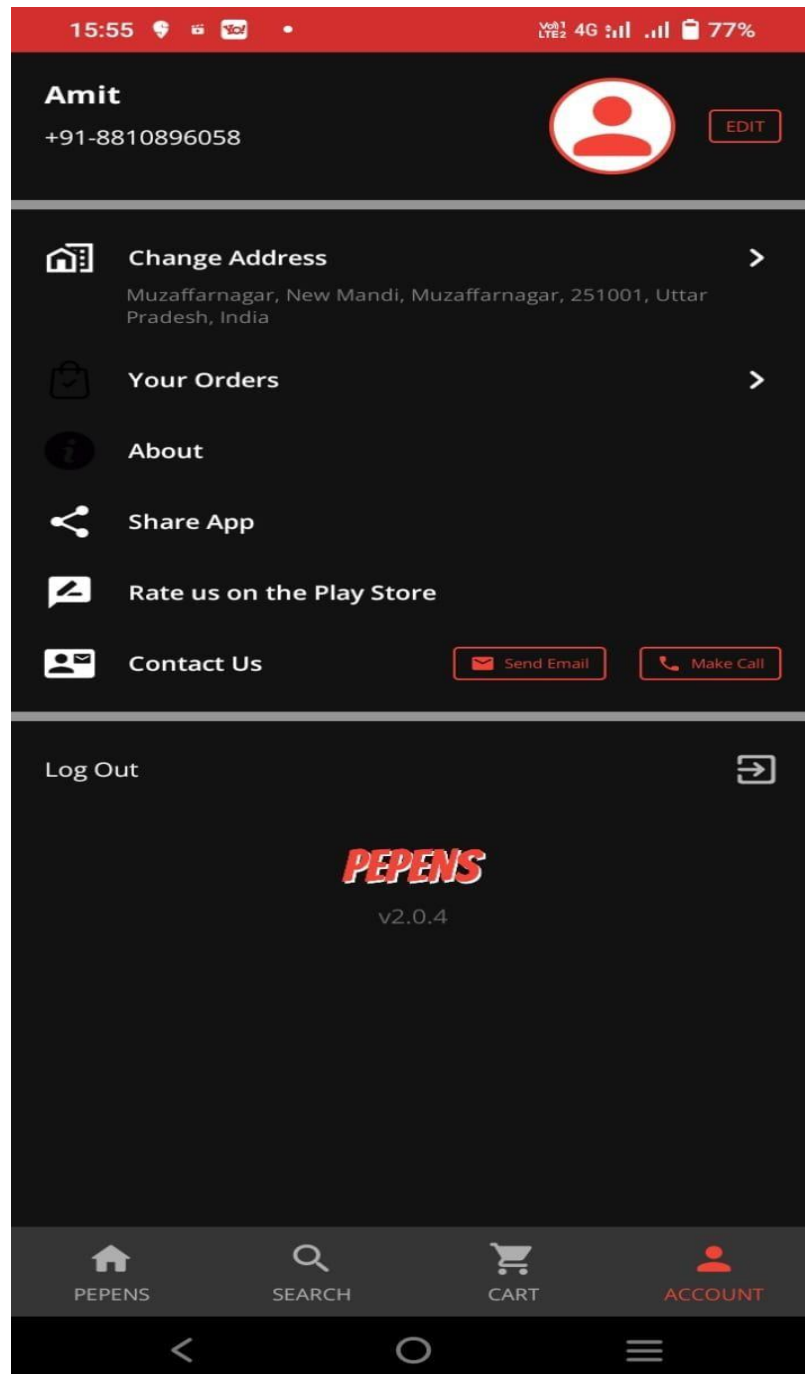


Fig. 4.1.3 : Map Page

## 4.2 ACCOUNT DETAILS PAGE



**Fig. 8.3 :** Account Details Page

### 4.3 HOME PAGE

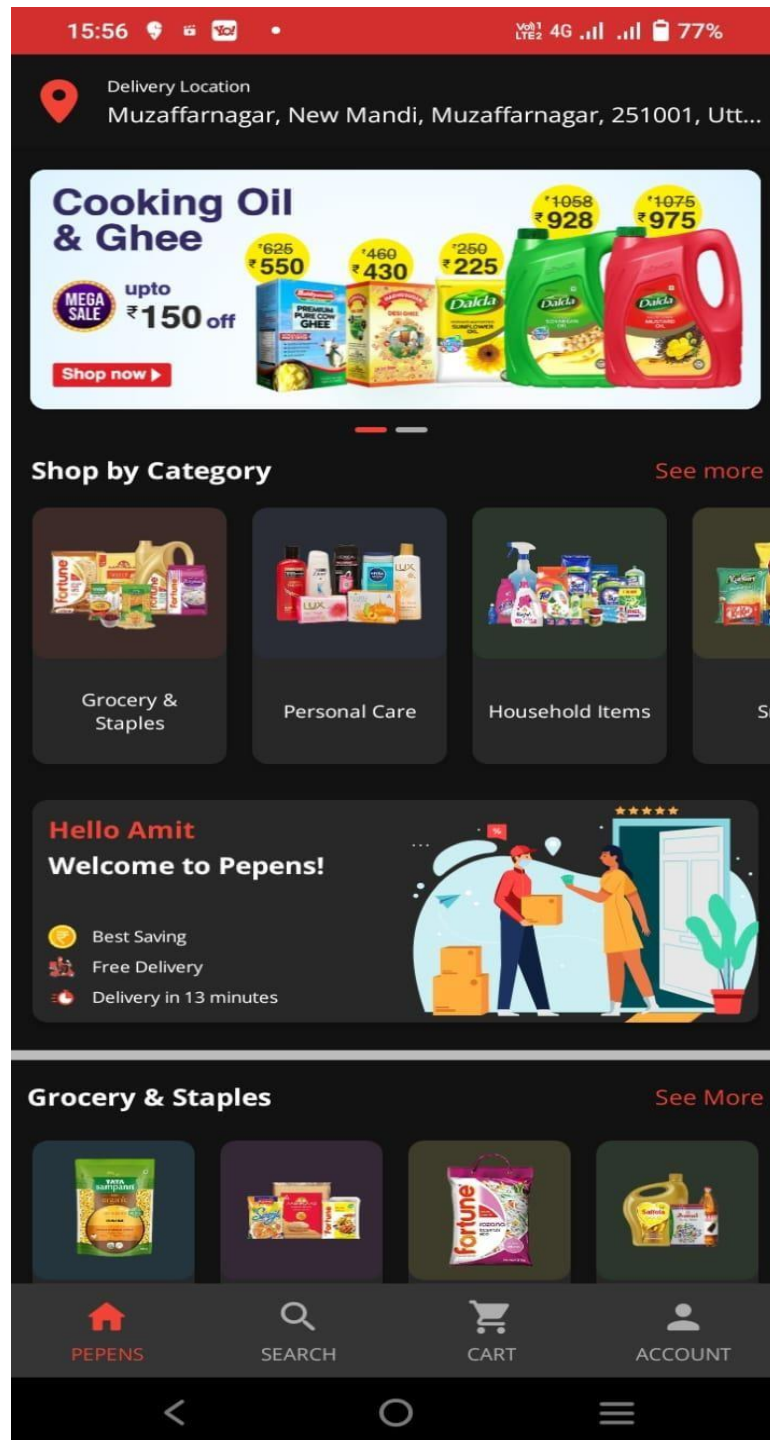


Fig. 4.3 : Home Page

## 4.4 SEARCH PAGE

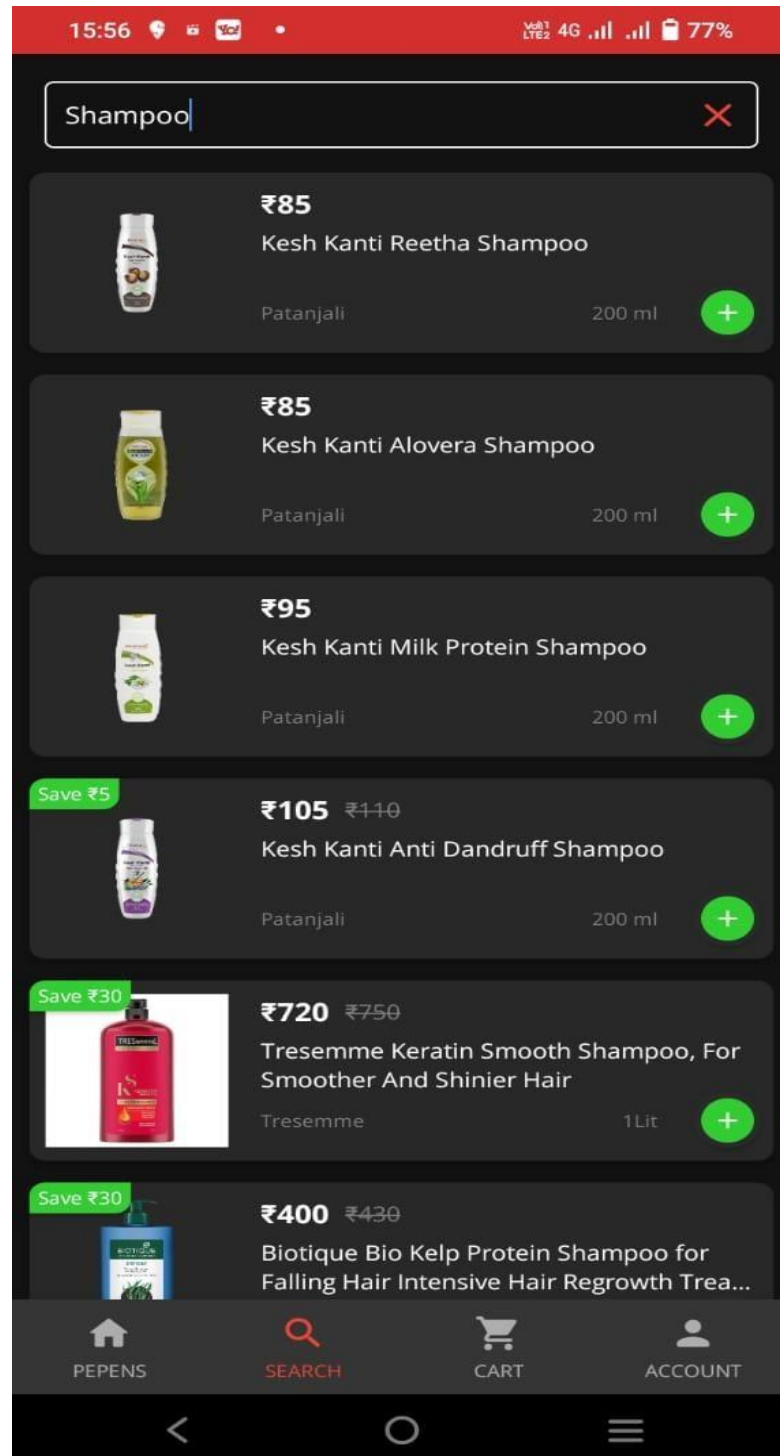


Fig. 4.4 : Search Page

# CHAPTER 5

## CODING

### 1. WORKING (FRONT-END)

#### 1.1. Scale\_down.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate
    android:duration="300"
    android:fromYDelta="50%p"
    android:toYDelta="100%p" />
  <alpha
    android:duration="300"
    android:fromAlpha="1"
    android:interpolator="@android:anim/decelerate_interpolator"
    android:toAlpha="0" />
</set>
```

#### 1.2. Scale\_up.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <translate
    android:duration="300"
    android:fromYDelta="100%p"
    android:toYDelta="0%p" />
  <alpha
    android:duration="300"
    android:fromAlpha="0.2"
    android:interpolator="@android:anim/decelerate_interpolator"
    android:toAlpha="1" />
</set>
```

### 1.3. Scale\_add.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <translate
        android:duration="150"
        android:fromYDelta="-100%p"
        android:toYDelta="0%p" />
    <alpha
        android:duration="150"
        android:fromAlpha="0.3"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:toAlpha="1.0" />
</set>
```

### 1.4. Scale\_in.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <translate
        android:duration="150"
        android:fromYDelta="100%p"
        android:toYDelta="0%p" />
    <alpha
        android:duration="150"
        android:fromAlpha="0.3"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:toAlpha="1.0" />
</set>
```

### 1.5. Scale\_out.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true">
    <translate
        android:duration="150"
        android:fromYDelta="0%p"
        android:toYDelta="-100%p" />
```

```

<alpha
  android:duration="150"
  android:fromAlpha="1.0"
  android:interpolator="@android:anim/accelerate_interpolator"
  android:toAlpha="0.3" />
</set>

```

## 1.6. Scale\_remove.xml

```

<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
  android:fillAfter="true">
  <translate
    android:duration="150"
    android:fromYDelta="0%p"
    android:toYDelta="100%p" />
  <alpha
    android:duration="150"
    android:fromAlpha="1.0"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:toAlpha="0.3" />
</set>

```

## 2. DRAWABLE

### 2.1. a 1\_background.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/firstColor" />
  <corners
    android:bottomLeftRadius="@dimen/eight_dp"
    android:topRightRadius="@dimen/eight_dp" />
</shape>

```

### 2.2. a 2\_background.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">

```



```

<solid android:color="@color/secondColor" />
<corners android:topRightRadius="@dimen/eight_dp"
    android:bottomLeftRadius="@dimen/eight_dp"/>
</shape>

```

### 2.3. a 3\_background.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/thirdColor" />
    <corners
        android:bottomLeftRadius="@dimen/eight_dp"
        android:topRightRadius="@dimen/eight_dp" />
</shape>

```

### 2.4. a 4\_background.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/forthColor" />
    <corners
        android:bottomLeftRadius="@dimen/eight_dp"
        android:topRightRadius="@dimen/eight_dp" />
</shape>

```

### 2.5. a 5\_background.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">

    <solid android:color="@color/fifthColor" />
    <corners
        android:bottomLeftRadius="@dimen/eight_dp"
        android:topRightRadius="@dimen/eight_dp" />

</shape>

```

## 2.6. a\_6\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/sixthColor" />
  <corners
    android:bottomLeftRadius="@dimen/eight_dp"
    android:topRightRadius="@dimen/eight_dp" />
</shape>
```

## 2.7. a\_7\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="#dedddd" />
  <corners
    android:bottomLeftRadius="@dimen/eight_dp"
    android:topRightRadius="@dimen/eight_dp" />
</shape>
```

## 2.8. Accuracy\_bg.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="#99F06917" />
  <corners android:radius="@dimen/two_dp" />
</shape>
```

## 2.9. B1\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/firstColor" />
  <corners
    android:bottomRightRadius="@dimen/eight_dp"
    android:topLeftRadius="@dimen/eight_dp" />
</shape>
```

## 2.10. B2\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/secondColor" />
  <corners
    android:bottomRightRadius="@dimen/eight_dp"
    android:topLeftRadius="@dimen/eight_dp" />
</shape>
```

## 2.11. B3\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/thirdColor" />
  <corners
    android:bottomRightRadius="@dimen/eight_dp"
    android:topLeftRadius="@dimen/eight_dp" />
</shape>
```

## 2.12. B4\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="@color/forthColor" />
  <corners
    android:bottomRightRadius="@dimen/eight_dp"
    android:topLeftRadius="@dimen/eight_dp" />
</shape>
```

## 2.13. C1.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <stroke
    android:width="2dp"
    android:color="@color/grey" />
  <corners android:radius="1dp" />
</shape>
```

## 2.14. C1\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/first1Color" />
    <corners android:radius="@dimen/thirty_dp" />
</shape>
```

## 2.15. C2\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/second1Color" />
    <corners android:radius="@dimen/thirty_dp" />
</shape>
```

## 2.16. Cart\_icon.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="16dp"
    android:height="16dp"
    android:viewportWidth="200.391"
    android:viewportHeight="187.007">
    <path
        android:fillColor="#000000"
        android:pathData="M157.35,165.809m-21.199,0a21.199,21.199 0,1 1,42.398
0a21.199,21.199 0,1 1,-42.398 0" />
    <path
        android:fillColor="#000000"
        android:pathData="M72.555,165.809m-21.199,0a21.199,21.199 0,1 1,42.398
0a21.199,21.199 0,1 1,-42.398 0" />
    <path
        android:fillColor="#ff3a52"
        android:pathData="M199.327,26.868a4.239,4.239 0,0 0,-3.18 -
1.433l43.517,25.435a4.239,4.239 0,0 0,-4.155 5.088l16.959,84.795a4.592,4.592 0,0 0,4.714
3.392l127.192,-16.959a4.239,4.239 0,0 0,3.646 -3.68l8.479,-67.836A4.237,4.237 0,0 0,199.327
26.868Z" />
    <path
        android:fillColor="#fafafa"
        android:pathData="M72.554,101.907a4.24,4.24 0,0 1,-4.18 -3.544l-8.479,-50.877a4.24,4.24
0,0 1,8.361 -1.391l8.479,50.877a4.24,4.24 0,0 1,-3.485 4.876A4.076,4.076 0,0 1,72.554
101.907Z" />
    <path
        android:fillColor="#fafafa"
```

```

        android:pathData="M93.753,97.667a4.24,4.24 0,0 1,-4.24 -3.858l-4.24,-46.637a4.24,4.24 0,0
1,3.824 -4.618l0.034,0a4.24,4.24 0,0 1,4.6 3.841h0l4.24,46.637a4.241,4.241 0,0 1,-3.841 4.6h0Z"
/>
    <path
        android:fillColor="#fafafa"
        android:pathData="M114.952,93.427a4.24,4.24 0,0 1,-4.24 -4.24v-42.4a4.24,4.24 0,1 1,8.48
0v42.4A4.239,4.239 0,0 1,114.952 93.427Z" />
    <path
        android:fillColor="#fafafa"
        android:pathData="M136.15,89.188h-0.475a4.24,4.24 0,0 1,-3.77 -4.663c0,-0.014 0,-0.029
0,-0.043l4.24,-38.158a4.3,4.3 0,0 1,4.707 -3.773,4.241 4.241,0 0,1 3.748,4.68h0l-
4.24,38.158A4.24,4.24 0,0 1,136.15 89.188Z" />
    <path
        android:fillColor="#fafafa"
        android:pathData="M157.35,84.947a4.239,4.239 0,0 1,-4.113 -5.266l8.479,-
33.918a4.24,4.24 0,1 1,8.272 1.861c-0.014,0.064 -0.03,0.127 -0.048,0.191l-
8.479,33.918A4.239,4.239 0,0 1,157.35 84.947Z" />
    <path
        android:fillColor="#000000"
        android:pathData="M174.3,135.672L76.172,135.672a29.823,29.823 0,0 1,-29.18 -
23.853L26.272,8.48L4.251,8.48a4.24,4.24 0,1 1,0 -8.48L29.758,0a4.248,4.248 0,0 1,4.166
3.392L55.333,110.149a21.3,21.3 0,0 0,20.839 17.044L174.3,127.193a4.24,4.24 0,1 1,0 8.48Z" />
</vector>

```

## 2.17. Discount\_background.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@android:color/holo_red_dark" />
    <stroke
        android:width="1dp"
        android:color="@android:color/white" />
    <corners android:radius="@dimen/twenty_dp" />
</shape>

```

## 2.18. Dot\_background.xml

```

<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:width="@dimen/eight_dp"

```

```

        android:height="@dimen/eight_dp">
        <shape android:shape="rectangle">
            <solid android:color="#CCAAAAAA" />
            <corners android:radius="@dimen/four_dp" />
        </shape>
    </item>
</layer-list>

```

## 2.19. Dot\_selected.xml

```

<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:width="@dimen/sixteen_dp"
        android:height="@dimen/eight_dp">
        <shape android:shape="rectangle">
            <solid android:color="#CCF06917" />
            <corners android:radius="@dimen/four_dp" />
        </shape>
    </item>
</layer-list>

```

## 2.20. Dot\_blue\_\_view.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">

    <stroke android:width="1.5dp"
        android:color="#1A237E"
        android:dashWidth="@dimen/four_dp"
        android:dashGap="@dimen/two_dp" />

    <solid android:color="#0D1A237E" />

    <corners android:radius="@dimen/four_dp" />

</shape>

```

### 2.21. Egg\_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
  <item>
    <shape>
      <stroke
        android:width="0.7dp"
        android:color="@color/yellow" />
      <solid android:color="@color/colorBG" />
      <corners android:radius="@dimen/two_dp" />
    </shape>
  </item>
  <item
    android:bottom="@dimen/two_dp"
    android:end="@dimen/two_dp"
    android:start="@dimen/two_dp"
    android:top="@dimen/two_dp">
    <shape>
      <solid android:color="@color/yellow" />
      <corners android:radius="@dimen/eight_dp" />
    </shape>
  </item>
</layer-list>
```

### 2.22. Fav\_view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <stroke android:color="@color/pink"
    android:width="0.8dp"/>
  <corners android:radius="@dimen/eight_dp" />
</shape>
```

### 2.23. Ic\_discount.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
  android:width="12dp"
  android:height="12dp"
  android:tint="@color/white"
  android:viewportWidth="512.003"
  android:viewportHeight="512.003">
  <path
```

```

        android:fillColor="#FF000000"
        android:pathData="M477.958,262.633c-2.06,-4.215 -2.06,-9.049 0,-13.263l19.096,-
39.065c10.632,-21.751 2.208,-47.676 -19.178,-59.023l-38.41,-20.38c-4.144,-2.198 -6.985,-6.11 -
7.796,-10.729l-7.512,-42.829c-4.183,-23.846 -26.241,-39.87 -50.208,-36.479l-43.053,6.09c-
4.647,0.656 -9.242,-0.838 -12.613,-4.099l-31.251,-30.232c-17.401,-16.834 -44.661,-16.835 -
62.061,0l193.72,42.859c-3.372,3.262 -7.967,4.753 -12.613,4.099l-43.053,-6.09c-23.975,-3.393 -
46.025,12.633 -50.208,36.479l-7.512,42.827c-0.811,4.62 -3.652,8.531 -7.795,10.73l-
38.41,20.38c-21.386,11.346 -29.81,37.273 -19.178,59.024l19.095,39.064c2.06,4.215 2.06,9.049
0,13.263l-19.096,39.064c-10.632,21.751 -2.208,47.676 19.178,59.023l38.41,20.38c4.144,2.198
6.985,6.11 7.796,10.729l7.512,42.829c3.808,21.708 22.422,36.932 43.815,36.93c2.107,0 4.245,-
0.148 6.394,-0.452l43.053,-6.09c4.643,-0.659 9.241,0.838
12.613,4.099l31.251,30.232c8.702,8.418 19.864,12.626 31.03,12.625c11.163,-0.001 22.332,-
4.209 31.03,-12.625l31.252,-30.232c3.372,-3.261 7.968,-4.751 12.613,-
4.099l43.053,6.09c23.978,3.392 46.025,-12.633 50.208,-36.479l7.513,-42.827c0.811,-4.62
3.652,-8.531 7.795,-10.73l38.41,-20.38c21.386,-11.346 29.81,-37.273 19.178,-
59.024l477.958,262.633zM196.941,123.116c29.852,0 54.139,24.287 54.139,54.139s-
24.287,54.139 -54.139,54.139s-54.139,-24.287 -54.139,-54.139S167.089,123.116
196.941,123.116zM168.997,363.886c-2.883,2.883 -6.662,4.325 -10.44,4.325s-7.558,-1.441 -
10.44,-4.325c-5.766,-5.766 -5.766,-15.115 0,-20.881l194.889,-194.889c5.765,-5.766 15.115,-
5.766 20.881,0c5.766,5.766 5.766,15.115 0,20.881L168.997,363.886zM315.061,388.888c-
29.852,0 -54.139,-24.287 -54.139,-54.139s24.287,-54.139 54.139,-54.139c29.852,0
54.139,24.287 54.139,54.139S344.913,388.888 315.061,388.888z" />
    <path
        android:fillColor="#FF000000"
        android:pathData="M315.061,310.141c-13.569,0 -24.609,11.039 -
24.609,24.608s11.039,24.608 24.609,24.608c13.569,0 24.608,-11.039 24.608,-
24.608S328.63,310.141 315.061,310.141z" />
    <path
        android:fillColor="#FF000000"
        android:pathData="M196.941,152.646c-13.569,0 -24.608,11.039 -24.608,24.608c0,13.569
11.039,24.609 24.608,24.609c13.569,0 24.609,-11.039 24.609,-24.609C221.549,163.686
210.51,152.646 196.941,152.646z" />
</vector>

```

## 2.24. Ic\_discount.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:tint="#FFFFFF"
    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="@android:color/white"

```



```

        android:pathData="M12,4L5,4C3.34,4 2,5.34 2,7v8c0,1.66 1.34,3 3,3l-1,1v1h1l2,-2.03L9,18v-
5L4,13L4,5.98L13,6v2h2L15,7c0,-1.66 -1.34,-3 -3,-3zM5,14c0.55,0 1,0.45 1,1s-0.45,1 -1,1 -1,-0.45
-1,-1 0.45,-1 1,-1zM20.57,9.66c-0.14,-0.4 -0.52,-0.66 -0.97,-0.66h-7.19c-0.46,0 -0.83,0.26 -
0.98,0.66L10,13.77l0.01,5.51c0,0.38 0.31,0.72 0.69,0.72h0.62c0.38,0 0.68,-0.38 0.68,-
0.76L12,18h8v1.24c0,0.38 0.31,0.76 0.69,0.76h0.61c0.38,0 0.69,-0.34 0.69,-0.72l0.01,-1.37v-
4.14l-1.43,-4.11zM12.41,10h7.19l1.03,3h-9.25l1.03,-3zM12,16c-0.55,0 -1,-0.45 -1,-1s0.45,-1 1,-1
1,0.45 1,1 -0.45,1 -1,1zM20,16c-0.55,0 -1,-0.45 -1,-1s0.45,-1 1,-1 1,0.45 1,1 -0.45,1 -1,1z" />
</vector>

```

## 2.25. Ic\_empty\_cart.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="227.5dp"
    android:height="167.8dp"
    android:viewportWidth="227.5"
    android:viewportHeight="167.8">
    <path
        android:pathData="M70.8,19.6c-2,0 -3.5,-1.6 -3.5,-3.5c0,-2 1.6,-3.5 3.5,-3.5c2,0
3.5,1.6 3.5,3.5C74.4,18 72.8,19.6 70.8,19.6zM70.8,13.9c-1.2,0 -2.1,1 -2.1,2.1s1,2.1
2.1,2.1c1.2,0 2.1,-1 2.1,-2.1S72,13.9 70.8,13.9z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M185.3,168.7c-2,0 -3.5,-1.6 -3.5,-3.5c0,-2 1.6,-3.5 3.5,-3.5c2,0
3.5,1.6 3.5,3.5C188.9,167.1 187.3,168.7 185.3,168.7zM185.3,163c-1.2,0 -2.1,1 -
2.1,2.1c0,1.2 1,2.1 2.1,2.1c1.2,0 2.1,-1 2.1,-2.1C187.5,164 186.5,163 185.3,163z"
        android:fillColor="FFFFFF"/>
    <path
        android:pathData="M191.6,21.7c-2,0 -3.5,-1.6 -3.5,-3.5c0,-2 1.6,-3.5 3.5,-3.5c2,0
3.5,1.6 3.5,3.5C195.2,20.1 193.6,21.7 191.6,21.7zM191.6,16c-1.2,0 -2.1,1 -2.1,2.1c0,1.2
1,2.1 2.1,2.1c1.2,0 2.1,-1 2.1,-2.1C193.8,17 192.8,16 191.6,16z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M219,151.3c-2,0 -3.5,-1.6 -3.5,-3.5c0,-2 1.6,-3.5 3.5,-3.5c2,0
3.5,1.6 3.5,3.5S220.9,151.3 219,151.3zM219,145.6c-1.2,0 -2.1,1 -2.1,2.1c0,1.2 1,2.1
2.1,2.1c1.2,0 2.1,-1 2.1,-2.1S220.1,145.6 219,145.6z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M10.9,89.3c-2,0 -3.5,-1.6 -3.5,-3.5c0,-2 1.6,-3.5 3.5,-3.5c2,0
3.5,1.6 3.5,3.5C14.5,87.7 12.9,89.3 10.9,89.3zM10.9,83.6c-1.2,0 -2.1,1 -2.1,2.1c0,1.2

```

```

1,2.1 2.1,2.1c1.2,0 2.1,-1 2.1,-2.1C13.1,84.6 12.1,83.6 10.9,83.6z"
    android:fillColor="#6F8DD8"/>
<path
    android:pathData="M125.2,7h-1.5v5.5c0,-0.6 -0.5,-1.1 -1.1,-1.1s-1.1,0.5 -
1.1,1.1v7h120c-0.6,0 -1.1,0.5 -1.1,1.1s0.5,1.1 1.1,1.1h1.5v1.5c0,0.6 0.5,1.1 1.1,1.1s1.1,-
0.5 1.1,-1.1v9.1h1.5c0.6,0 1.1,-0.5 1.1,-1.1C126.2,7.5 125.8,7 125.2,7z"
    android:fillColor="#6F8DD8"/>
<path
    android:pathData="M15.9,135.1h-1.5v-1.5c0,-0.6 -0.5,-1.1 -1.1,-1.1s-1.1,0.5 -
1.1,1.1v1.5h-1.5c-0.6,0 -1.1,0.5 -1.1,1.1s0.5,1.1 1.1,1.1h1.5v1.5c0,0.6 0.5,1.1
1.1,1.1s1.1,-0.5 1.1,-1.1v-1.5h1.5c0.6,0 1.1,-0.5 1.1,-1.1S16.5,135.1 15.9,135.1z"
    android:fillColor="#6F8DD8"/>
<path
    android:pathData="M221.1,97.6h-1.5v-1.5c0,-0.6 -0.5,-1.1 -1.1,-1.1s-1.1,0.5 -
1.1,1.1v1.5h-1.5c-0.6,0 -1.1,0.5 -1.1,1.1s0.5,1.1 1.1,1.1h1.5v1.5c0,0.6 0.5,1.1
1.1,1.1s1.1,-0.5 1.1,-1.1v-1.5h1.5c0.6,0 1.1,-0.5 1.1,-1.1C222.2,98.1 221.7,97.6
221.1,97.6z"
    android:fillColor="#B9D3EE"/>
<path
    android:pathData="M133.2,35.1c-16.9,5.1 -26.7,-4.4 -42.9,-
7C83,27 72.3,25.9 65,28.5c-7.7,2.9 -7.3,8.1 -7.7,15.8C56.7,61.8 35.6,54.1
31.6,71C29,81.6 30.3,90 38,97c4,4 7,8.1 7.3,13.9c0.4,11.4 -11.4,28.6 3.7,36.3c8.4,4.4
19.8,-0.4 28.9,4c4.8,2.2 7.7,6.2 12.8,8.1c8.8,2.6 16.1,0.4 24.2,-2.6c7.7,-2.9 15.4,-5.9
23.1,-6.6c5.9,-0.4 11.7,0.7 17.2,-0.7c6.2,-1.5 8.1,-5.5 12.8,-8.4c5.9,-3.3 13.2,0 19.4,-
2.2c15.4,-4.8 2.9,-19.8 3.3,-30.4c0.4,-11.7 9.3,-20.5 8.6,-
33.3c-1.1,-14.3 -10.8,-17.6 -22.2,-22.7c-4,-1.8 -7,-3.7 -9.5,-7.3c-2.9,-3.7 -4.4,-9.9 -8.1,-
12.5C153.7,28.1 141.9,30.5 133.2,35.1"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M174.6,47.7c1.8,0 1.8,-2.8 0,-2.8C172.8,44.8 172.8,47.7
174.6,47.7L174.6,47.7z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M199.7,103.1c2.6,2.6 6.6,-1.3 4.1,-4C201.2,96.5 197.2,100.5

```

```

199.7,103.1L199.7,103.1z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M36.3,120.3c3.7,0 3.7,-5.7 0,-5.7C32.6,114.6 32.6,120.3
36.3,120.3L36.3,120.3z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M184.2,46.3c3.7,0 3.7,-5.7 0,-5.7S180.5,46.3
184.2,46.3L184.2,46.3z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M30.6,111.3c4.8,0 4.8,-7.4 0,-7.4S25.8,111.3
30.6,111.3L30.6,111.3z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M199.7,116.5c3.3,3.4 8.6,-1.7 5.3,-5.1C201.7,107.9 196.4,113.1
199.7,116.5L199.7,116.5z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M48,51.1c4.8,0 4.8,-7.4 0,-7.4C43.2,43.7 43.2,51.1
48,51.1L48,51.1z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M176,41.7c5.5,0 5.5,-8.5 0,-8.5C170.6,33.1 170.6,41.7
176,41.7L176,41.7z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>

```

```

<path
  android:pathData="M27.5,117.9c2.7,0 2.7,-4.3 0,-4.3S24.8,117.9
27.5,117.9L27.5,117.9z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M208.9,108.9c1.9,2 5,-1 3,-3C210,104 207,106.9
208.9,108.9L208.9,108.9z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M39.1,55.2c2.7,0 2.7,-4.3 0,-4.3C36.4,50.9 36.4,55.2
39.1,55.2L39.1,55.2z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M39.8,47.4c1.8,0 1.8,-2.8 0,-2.8C38,44.6 38,47.4
39.8,47.4L39.8,47.4z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M125.5,32.4c3.7,0 3.7,-5.7 0,-5.7C121.8,26.8 121.8,32.4
125.5,32.4L125.5,32.4z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M74.4,161.5c3.7,0 3.7,-5.7 0,-5.7C70.8,155.8 70.8,161.5
74.4,161.5L74.4,161.5z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M176.9,150.9c3.7,0 3.7,-5.7 0,-5.7S173.2,150.9
176.9,150.9L176.9,150.9z"
  android:strokeAlpha="0.34"

```

```

        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M117,28.2c1.8,0 1.8,-2.8 0,-2.8C115.1,25.3 115.1,28.2
117,28.2L117,28.2z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M66.1,156.6c1.8,0 1.8,-2.8 0,-2.8S64.2,156.6
66.1,156.6L66.1,156.6z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M168.4,154.5c1.8,0 1.8,-2.8 0,-2.8S166.5,154.5
168.4,154.5L168.4,154.5z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M111.2,67.2h24.9v3.1h-24.9C111.2,70.3 111.2,67.2 111.2,67.2z"
        android:fillColor="#3700B3"/>
    <path
        android:pathData="M111.2,75h24.9v3.1h-24.9C111.2,78.1 111.2,75 111.2,75z"
        android:fillColor="#3700B3"/>
    <path
        android:pathData="M80,87.4h26.5v3.1H80V87.4z"
        android:fillColor="#3700B3"/>
    <path
        android:pathData="M80,95.2h20.3v3.1H80V95.2z"
        android:fillColor="#3700B3"/>
    <path
        android:pathData="M106.5,62.5H80v18.7h26.5V62.5zM103.4,78.1H83.2V65.6h20.3L10
3.4,78.1L103.4,78.1z"
        android:fillColor="#3700B3"/>
    <path
        android:pathData="M143.9,91.5V57.8h4.8l-0.3,-1.8c-0.4,-2.2 -1.4,-4.3 -3,-5.9c-2,-2.1
-4.8,-3.2 -7.7,-3.2H83.2c-6,0 -10.9,4.9 -10.9,10.9v77.9h42.6c7.9,5.5 18.2,6.3
26.8,2s14.3,-12.9 14.8,-22.5C156.9,105.6 152.1,96.5 143.9,91.5zM137.7,50c3.1,0 5.9,1.8

```

```

7.2,4.7H93.6c-0.5,-1.7 -1.4,-3.3 -2.7,-4.6L90.8,50C90.8,50 137.7,50
137.7,50zM75.4,132.6V57.8c0,-3.2 1.9,-6 4.8,-7.2s6.3,-0.5 8.5,1.7c1.1,1.1 1.9,2.6
2.1,4.2l0.2,1.3h49.8v32c-11.7,-5.3 -25.4,-1.5 -32.7,9.1c-7.3,10.5 -6,24.7
3,33.8L75.4,132.6L75.4,132.6zM143.9,132.6c-0.5,0.4 -1,0.7 -1.5,1c-0.3,0.2 -0.6,0.3 -
0.9,0.5c-1,0.6 -2.1,1.1 -3.3,1.6c-7.5,2.9 -16,1.7 -22.4,-3.1c-0.4,-0.3 -0.7,-0.5 -1,-0.8c-7.5,-
6.3 -10.3,-16.7 -6.9,-25.9s12.1,-15.4 21.9,-15.4c3.8,0 7.5,0.9 10.9,2.7h0.1c7.1,3.8
11.7,10.9 12.3,18.8C153.8,120.1 150.3,127.8

143.9,132.6z"
    android:fillColor="#3700B3"/>
<path
    android:pathData="M138.7,113.9l8.8,-8.8l-8.8,-8.8l-8.8,8.8l-8.8,-8.8l-8.8,8.8l8.8,8.8l-
8.8,8.8l8.8,8.8l8.8,-8.8l8.8,8.8l8.8,-8.8L138.7,113.9zM138.7,127.2l-8.8,-8.8l-8.8,8.8l-4.4,-
4.4l8.8,-8.8l-8.8,-8.8l4.4,-4.4l8.8,8.8l8.8,-8.8l4.4,4.4l-8.8,8.8l8.8,8.8L138.7,127.2z"
    android:fillColor="#3700B3"/>
<path
    android:pathData="M80,103h20.3v3.1H80V103z"
    android:fillColor="#3700B3"/>
<path
    android:pathData="M63.2,69.2v4.1l9,46.7c0.7,3.9 4.1,6.7 8.1,6.7h68.5c3.9,0 7.3,-2.8
8.1,-6.7l9,-46.7v-4.1H63.2z"
    android:fillColor="#EB423F"/>
<path
    android:pathData="M169.9,73.3H59.1c-2.3,0 -4.1,-1.8 -4.1,-4.1s1.8,-4.1 4.1,-
4.1h110.8c2.3,0 4.1,1.8 4.1,4.1s172.2,73.3 169.9,73.3z"
    android:fillColor="#C7312E"/>
<path
    android:pathData="M59.9,95l89,66c1.6,-1.6 4.2,-1.6 5.8,0l0.4,0.4c1.6,1.6 1.6,4.2
0,5.8l-29.1,29.1c-1.6,1.6 -4.2,1.6 -5.8,0l-0.4,-0.4C58.3,99.2 58.3,96.6 59.9,95z"
    android:fillColor="#4F5659"/>
<path
    android:pathData="M133.8,66.3l0.4,-0.4c1.6,-1.6 4.2,-1.6 5.8,0l169,95c1.6,1.6
1.6,4.2 0,5.8l-0.4,0.4c-1.6,1.6 -4.2,1.6 -5.8,0l-29.1,-29.1C132.2,70.5 132.2,67.9
133.8,66.3z"
    android:fillColor="#4F5659"/>
<path
    android:pathData="M91.9,67.2m-6.2,0a6.2,6.2 0,1 1,12.4 0a6.2,6.2 0,1 1,-12.4 0"
    android:fillColor="#E9EEF2"/>
<path
    android:pathData="M137.1,67.2m-6.2,0a6.2,6.2 0,1 1,12.4 0a6.2,6.2 0,1 1,-12.4 0"

```

```

        android:fillColor="#E9EEF2"/>
    <path
        android:pathData="M89.9,114.3c-2.3,0 -4.1,-1.8 -4.1,-4.1v89.7c0,-2.3 1.8,-4.1 4.1,-
4.1s4.1,1.8 4.1,4.1v20.5c94,112.5 92.1,114.3 89.9,114.3z"
        android:fillColor="#E9EEF2"/>
    <path
        android:pathData="M106.3,114.3c-2.3,0 -4.1,-1.8 -4.1,-4.1v89.7c0,-2.3 1.8,-4.1 4.1,-
4.1s4.1,1.8 4.1,4.1v20.5c110.4,112.5 108.6,114.3 106.3,114.3z"
        android:fillColor="#E9EEF2"/>
    <path
        android:pathData="M122.7,114.3c-2.3,0 -4.1,-1.8 -4.1,-
4.1v89.7c0,-2.3 1.8,-4.1 4.1,-4.1c2.3,0 4.1,1.8 4.1,4.1v20.5c126.8,112.5 125,114.3
122.7,114.3z"
        android:fillColor="#E9EEF2"/>
    <path
        android:pathData="M139.1,114.3c-2.3,0 -4.1,-1.8 -4.1,-4.1v89.7c0,-2.3 1.8,-4.1 4.1,-
4.1c2.3,0 4.1,1.8 4.1,4.1v20.5c143.2,112.5 141.4,114.3 139.1,114.3z"
        android:fillColor="#E9EEF2"/>
</vector>

```

## 2.26. Ic\_feed.xml

```

<vector android:height="24dp" android:viewportHeight="512"
    android:viewportWidth="512" android:width="24dp"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#FF000000" android:pathData="m422,467c0,24.852 20.148,45
45,45s45,-20.148 45,-45v-226h-90zM422,467"/>
    <path android:fillColor="#FF000000" android:pathData="m75,106h242v60h-242zM75,106"/>
    <path android:fillColor="#FF000000" android:pathData="m256,376h61v30h-61zM256,376"/>
    <path android:fillColor="#FF000000" android:pathData="m60,512h347.383c-9.52,-12.574 -
15.383,-28.051 -15.383,-45v-467h-392v451c0,33 27,61 60,61zM347,436h-121v-
90h121zM347,316h-121v-30h121zM347,256h-121v-30h121zM45,76h302v120h-
302zM45,226h151v30h-151zM45,286h151v30h-151zM45,346h151v30h-151zM45,406h151v30h-
151zM45,406"/>
</vector>

```

## 2.27. Ic\_filter.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="18dp"
    android:height="18dp"
    android:viewportWidth="512.001"
    android:viewportHeight="512.001">
    <path
        android:fillColor="@color/colorIcon"
        android:pathData="M214.71,306.106v197.626c0,6.139 6.46,10.132 11.951,7.386l61.5,-
30.75c5.595,-2.798 9.13,-8.517 9.13,-14.772v-159.49c0,-5.702 1.967,-11.229 5.568,-
15.649l473.4,81.157c3.602,-4.42 5.568,-9.947 5.568,-15.649V16.516C478.968,7.395 471.573,0
462.452,0H49.548c-9.122,0 -16.516,7.395 -16.516,16.516v48.991c0,5.702 1.967,11.229
5.568,15.649l170.541,209.3C212.743,294.877 214.71,300.404
214.71,306.106z" />
    <path
        android:fillColor="#4E4E4E"
        android:pathData="M404.355,66.065H198.194c-4.561,0 -8.258,-3.697 -8.258,-
8.258V0H49.548c-9.122,0 -16.516,7.395 -16.516,16.516v48.991c0,5.702 1.967,11.229
5.568,15.649l170.541,209.3c3.602,4.42 5.569,9.948 5.569,15.65v197.625c0,6.139 6.46,10.132
11.951,7.386l17.105,-8.552V305.548c0,-4.561 3.697,-8.258 8.258,-8.258h11.74c5.474,0 10.593,-
2.712 13.667,-7.242L415.346,86.804C421.33,77.985 415.013,66.065 404.355,66.065z" />
    <path
        android:fillColor="@color/colorIcon"
        android:pathData="M123.502,66.065H33.08c0.126,5.502 2.037,10.817
5.52,15.091l170.541,209.3c1.681,2.064 2.968,4.383 3.903,6.835h6.065c5.821,0 9.814,-5.86
7.686,-
11.278l146.561,81.78C142.836,72.299 133.688,66.065 123.502,66.065z" />
</vector>
```

## 2.28. Ic\_gallery.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:aapt="http://schemas.android.com/aapt"
    android:width="512dp"
    android:height="512dp"
    android:viewportWidth="512"
    android:viewportHeight="512">
    <path
        android:pathData="M456.025,70.524l55.506,280.949c2.606,13.189 -5.984,26.007 -
19.172,28.613L84.584,460.648c-13.189,2.606 -26.007,-5.984 -28.613,-
```



```

19.172L0.466,160.527c-2.606,-13.189 5.984,-26.007 19.172,-28.613l407.774,-
80.562C440.601,48.746 453.419,57.335 456.025,70.524z">
  <aapt:attr name="android:fillColor">
    <gradient
      android:startY="261.16428"
      android:startX="255.90147"
      android:endY="-282.51138"
      android:endX="266.9418"
      android:type="linear">
      <item android:offset="0" android:color="#FF97AAAA"/>
      <item android:offset="1" android:color="#FFE8EFEE"/>
    </gradient>
  </aapt:attr>
</path>
<path
  android:pathData="M492.357,380.089L84.586,460.644c-13.188,2.612 -26.004,-
5.979 -28.617,-19.167l-3.657,-18.516c4.005,1.253 8.382,1.555
12.794,0.685L456.021,70.519l55.515,280.953C514.136,364.66 505.545,377.476
492.357,380.089z">
  <aapt:attr name="android:fillColor">
    <gradient
      android:startY="374.89453"
      android:startX="281.0373"
      android:endY="441.06195"
      android:endX="294.96838"
      android:type="linear">
      <item android:offset="0" android:color="#00363F3E"/>
      <item android:offset="1" android:color="#FF303033"/>
    </gradient>
  </aapt:attr>
</path>
<path
  android:pathData="M493.635,127.339v286.38c0,13.444 -10.911,24.355 -
24.355,24.355H53.624c-13.444,0 -24.355,-10.911 -
24.355,-24.355v-286.38c0,-13.444 10.911,-24.355 24.355,-
24.355H469.28C482.724,102.984 493.635,113.894 493.635,127.339z">
  <aapt:attr name="android:fillColor">
    <gradient
      android:startY="420.2071"

```

```

        android:startX="411.11838"
        android:endY="475.04614"
        android:endX="465.94675"
        android:type="linear">
        <item android:offset="0" android:color="#FF97AAAA"/>
        <item android:offset="1" android:color="#FFE8EFEE"/>
    </gradient>
</aapt:attr>
</path>
<path
    android:pathData="M371.032,438.074H53.624c-13.451,0 -24.355,-10.905 -24.355,-
24.355v-286.38c0,-13.451 10.905,-24.355 24.355,-24.355H469.28c13.451,0
24.355,10.905 24.355,24.355v194.51c0,3.727 -1.467,7.305 -4.082,9.959l-
95.005,96.423C388.343,434.528 379.872,438.074 371.032,438.074z">
    <aapt:attr name="android:fillColor">
        <gradient
            android:startY="92.42278"
            android:startX="261.46005"
            android:endY="447.53787"
            android:endX="261.46005"
            android:type="linear">
            <item android:offset="0" android:color="#FFFFFFFF"/>
            <item android:offset="1" android:color="#FFE8EFEE"/>
        </gradient>
    </aapt:attr>
</path>
<path
    android:pathData="M458.32,149.114v214.398c-16.218,16.462 -35.42,35.941 -
48.364,49.084H85.237c-11.401,0 -20.653,-9.252 -20.653,-20.653V149.114c0,-11.401
9.252,-20.653 20.653,-20.653h352.43C449.068,128.461 458.32,137.713
458.32,149.114z">
    <aapt:attr name="android:fillColor">
        <gradient
            android:startY="207.54105"
            android:startX="261.46005"
            android:endY="466.0878"
            android:endX="261.46005"
            android:type="linear">
            <item android:offset="0" android:color="#FF3FA9F5"/>
            <item android:offset="1" android:color="#FF0071E2"/>

```

```

    </gradient>
  </aapt:attr>
</path>
<path
  android:pathData="M493.635,394.235v19.484c0,13.444 -10.911,24.355 -
24.355,24.355H53.624c-13.444,0 -24.355,-10.911 -
24.355,-24.355v-19.484H493.635z">
  <aapt:attr name="android:fillColor">
    <gradient
      android:startY="410.3954"
      android:startX="261.46005"
      android:endY="442.49237"
      android:endX="261.46005"
      android:type="linear">
      <item android:offset="0" android:color="#00C2CECE"/>
      <item android:offset="0.179" android:color="#2DAFBCBC"/>
      <item android:offset="1" android:color="#FF5B6A6A"/>
    </gradient>
  </aapt:attr>
</path>
<path
  android:pathData="M64.705,394.143c0.064,0.605 0.158,1.2
0.273,1.788c0.021,0.109 0.043,0.218 0.066,0.326c0.123,0.578 0.269,1.148
0.439,1.708c0.03,0.098 0.063,0.195 0.095,0.293c0.183,0.571 0.385,1.133
0.615,1.681c0.021,0.05 0.045,0.097 0.066,0.147c0.226,0.525 0.475,1.037
0.741,1.538c0.039,0.075 0.077,0.151 0.117,0.226c0.273,0.498 0.571,0.98
0.882,1.452c0.07,0.107 0.14,0.213 0.212,0.318c0.31,0.451 0.638,0.89
0.982,1.313c0.086,0.106 0.175,0.21 0.262,0.315c0.354,0.419 0.721,0.828
1.106,1.217c0.081,0.081 0.165,0.159 0.247,0.239c0.364,0.356 0.74,0.699
1.13,1.027c0.075,0.063 0.145,0.129 0.221,0.191c3.562,2.918 8.113,4.672
13.074,4.672h324.719c10.13,-10.285 24.091,-24.449 37.457,-38.014L337.691,273.928c-
8.48,-7.779 -21.501,-7.779 -29.981,0l-48.684,44.66l-76.543,-76.499c-9.07,-10.86 -
26.039,-10.575 -34.686,0.868L64.584,353.074v38.867c0,0.655 0.038,1.3
0.098,1.94C64.69,393.968 64.695,394.056 64.705,394.143z">
  <aapt:attr name="android:fillColor">
    <gradient
      android:startY="382.49265"
      android:startX="250.97064"
      android:endY="240.76022"
      android:endX="109.227554"

```

```

        android:type="linear">
        <item android:offset="1E-4" android:color="#00008BF2"/>
        <item android:offset="1" android:color="#FF0046E2"/>
        </gradient>
    </aapt:attr>
</path>
<path
    android:pathData="M447.418,374.586c-13.35,13.536 -27.281,27.676 -37.405,37.95l-
98.748,-0.081l-154.413,-0.128l91.132,-83.609l59.726,-54.789c8.48,-7.779 21.501,-7.779
29.981,0l447.418,374.586z">
    <aapt:attr name="android:fillColor">
        <gradient
            android:startY="239.67004"
            android:startX="302.14505"
            android:endY="421.4144"
            android:endX="302.14505"

            android:type="linear">
            <item android:offset="0" android:color="#FFC2D600"/>
            <item android:offset="1" android:color="#FF7BD101"/>
            </gradient>
        </aapt:attr>
</path>
<path
    android:pathData="M337.691,273.928c-8.48,-7.779 -21.501,-7.779 -29.981,0l-
59.726,54.789l63.281,83.737l98.748,0.081c10.123,-10.274 24.054,-24.414 37.405,-
37.95l337.691,273.928z">
    <aapt:attr name="android:fillColor">
        <gradient
            android:startY="397.48904"
            android:startX="383.32816"
            android:endY="256.40732"
            android:endX="242.24643"

            android:type="linear">
            <item android:offset="1E-4" android:color="#001ACEB8"/>
            <item android:offset="1" android:color="#FF04959C"/>
            </gradient>
        </aapt:attr>
</path>
<path

```

```

        android:pathData="M357.506,292.114c-3.761,3.053 -9.682,2.345 -12.457,-2.137l-
0.012,-0.012c-3.285,-5.329 -11.028,-5.329 -14.314,0v0.012c-3.285,5.329 -11.04,5.329 -
14.325,0v-0.012c-1.637,-2.67 -4.4,-3.994 -7.163,-3.994s-5.526,1.324 -7.163,3.994c-
2.983,4.841 -9.647,5.282 -13.293,1.335l18.934,-17.367c8.475,-7.79 21.5,-7.79
29.974,0L357.506,292.114z">
        <aapt:attr name="android:fillColor">
            <gradient
                android:startY="265.97488"
                android:startX="323.15317"
                android:endY="291.949"
                android:endX="323.15317"
                android:type="linear">
                <item android:offset="0" android:color="#FFFFFFF"/>
                <item android:offset="1" android:color="#FFE8EFEE"/>
            </gradient>
        </aapt:attr>
    </path>
    <path
        android:pathData="M357.506,292.114c-3.761,3.053 -9.682,2.345 -12.457,-2.137l-
0.012,-0.012c-3.285,-5.329 -11.028,-5.329 -14.314,0v0.012c-3.285,5.329 -11.04,5.329 -
14.325,0v-0.012c-1.637,-2.67 -4.4,-3.994 -7.163,-3.994s-5.526,1.324 -7.163,3.994c-
2.983,4.841 -9.647,5.282 -13.293,1.335l18.934,-17.367c8.475,-7.79 21.5,-7.79
29.974,0L357.506,292.114z">
        <aapt:attr name="android:fillColor">
            <gradient
                android:startY="289.4013"
                android:startX="320.8506"
                android:endY="262.59512"
                android:endX="294.04443"
                android:type="linear">
                <item android:offset="0" android:color="#00C2CECE"/>
                <item android:offset="0.179" android:color="#2DAFBCBC"/>
                <item android:offset="1" android:color="#FF5B6A6A"/>
            </gradient>
        </aapt:attr>
    </path>
    <path
        android:pathData="M163.096,271.861c-0.14,0 -0.277,0.007 -0.416,0.011c0.155,-
1.267 0.236,-2.557 0.236,-3.866c0,-17.447 -14.144,-31.59 -31.59,-31.59c-12.994,0 -
24.155,7.846 -29.006,19.059c-1.13,-0.453 -2.363,-0.705 -3.654,-0.705c-5.423,0 -

```

```

9.819,4.396 -9.819,9.819c0,1.652 0.41,3.209 1.132,4.575c-6.953,2.599 -11.904,9.301 -
11.904,17.161c0,10.116 8.202,18.318 18.318,18.318h66.704c9.052,0 16.39,-7.338
16.39,-16.39S172.148,271.861 163.096,271.861z"
    android:fillColor="#008BF2"/>
<path
    android:pathData="M311.371,412.594l-229.455,-0.267c-9.821,-1.59 -17.332,-10.112
-17.332,-20.386v-38.867l83.214,-110.116c8.873,-11.741 26.512,-11.741
35.385,0.001l64.803,85.759l63.281,83.737L311.371,412.594z">
    <aapt:attr name="android:fillColor">
        <gradient
            android:startY="199.03944"
            android:startX="187.98361"
            android:endY="423.56912"
            android:endX="187.98361"
            android:type="linear">
                <item android:offset="0" android:color="#FFC2D600"/>
                <item android:offset="1" android:color="#FF7BD101"/>
            </gradient>
        </aapt:attr>
    </path>
<path
    android:pathData="M200.214,265.506c-3.668,3.877 -10.274,3.402 -13.246,-1.405v-
0.023c-1.649,-2.658 -4.4,-3.982 -7.163,-3.982s-5.515,1.324 -7.163,3.994c-1.648,2.658 -
4.4,3.994 -7.163,3.994s-5.515,-1.335 -7.163,-3.994c-1.648,-2.658 -4.4,-3.994 -7.163,-
3.994s-5.515,1.335 -7.163,3.994c-2.96,4.806 -9.542,5.282 -13.222,1.417l17.03,-
22.545c8.869,-11.748 26.515,-11.748 35.385,0L200.214,265.506z">
    <aapt:attr name="android:fillColor">
        <gradient
            android:startY="234.62456"
            android:startX="165.49715"
            android:endY="262.1134"
            android:endX="165.49715"
            android:type="linear">
                <item android:offset="0" android:color="#FFFFFFF"/>
                <item android:offset="1" android:color="#FFE8EFEE"/>
            </gradient>
        </aapt:attr>
    </path>
<path
    android:pathData="M492.387,327.665c0,0 -0.72,2.038 -3.553,4.871c-1.052,1.052 -
93.729,95.129 -93.729,95.129c-6.561,6.659 -15.52,10.41 -24.869,10.41l0,0c8.07,0

```

```

14.613,-6.542 14.613,-14.613V352.02c0,-13.451 10.905,-24.355 24.355,-
24.355L492.387,327.665L492.387,327.665z">
  <aapt:attr name="android:fillColor">
    <gradient
      android:startY="339.15762"
      android:startX="387.62762"
      android:endY="387.16458"
      android:endX="435.62387"
      android:type="linear">
      <item android:offset="0" android:color="#FFFFFFF"/>
      <item android:offset="1" android:color="#FFE8EFEE"/>
    </gradient>
  </aapt:attr>
</path>
<path
  android:pathData="M337.233,206.65c-0.14,0 -0.277,0.007 -0.416,0.011c0.155,-1.267 0.236,-
2.557 0.236,-3.866c0,-17.447 -14.144,-31.59 -31.59,-31.59c-12.994,0 -24.155,7.846 -
29.006,19.059c-1.13,-0.453 -2.363,-0.705 -3.654,-0.705c-5.423,0 -9.819,4.396 -
9.819,9.819c0,1.652 0.41,3.209 1.132,4.575c-6.953,2.599 -11.904,9.301 -11.904,17.161c0,10.116
8.202,18.318 18.318,18.318h66.704c9.052,0 16.39,-7.338 16.39,-16.39S346.286,206.65
337.233,206.65z"
  android:fillColor="#008BF2"/>
<path
  android:pathData="M458.32,221.675l-50.558,-50.558c-9.039,-9.494 -21.79,-15.422 -35.934,-
15.422c-27.405,0 -49.621,22.217 -49.621,49.621c0,14.144 5.927,26.895
15.422,35.934L458.32,361.941L458.32,221.675L458.32,221.675z">
  <aapt:attr name="android:fillColor">
    <gradient
      android:startY="308.15164"
      android:startX="474.6924"
      android:endY="138.22624"
      android:endX="304.75644"
      android:type="linear">
      <item android:offset="1E-4" android:color="#00008BF2"/>
      <item android:offset="1" android:color="#FF0046E2"/>
    </gradient>
  </aapt:attr>
</path>
<path
  android:pathData="M371.829,205.312m-49.621,0a49.621,49.621 0,1 1,99.242
0a49.621,49.621 0,1 1,-99.242 0">

```

```

<aapt:attr name="android:fillColor">
  <gradient
    android:startY="119.30362"
    android:startX="371.84055"
    android:endY="295.83182"
    android:endX="371.84055"
    android:type="linear">
    <item android:offset="0" android:color="#FFFCF95"/>
    <item android:offset="0.427" android:color="#FFFC954"/>
    <item android:offset="1" android:color="#FFFC200"/>
  </gradient>
</aapt:attr>
</path>
</vector>

```

## 2.29. Ic\_home.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
  android:width="24dp"
  android:height="24dp"
  android:tint="#FFFFFF"
  android:viewportWidth="24"
  android:viewportHeight="24">
  <path
    android:fillColor="@android:color/white"
    android:pathData="M10,19v-5h4v5c0,0.55 0.45,1 1,1h3c0.55,0 1,-0.45 1,-1v-7h1.7c0.46,0 0.68,-0.57 0.33,-0.87L12.67,3.6c-0.38,-0.34 -0.96,-0.34 -1.34,0l-8.36,7.53c-0.34,0.3 -0.13,0.87 0.33,0.87H5v7c0,0.55 0.45,1 1,1h3c0.55,0 1,-0.45 1,-1z" />
</vector>

```

## 2.30. Ic\_home\_basket.xml

```

<vector android:height="24dp" android:viewportHeight="512"
  android:viewportWidth="512" android:width="24dp"
  xmlns:android="http://schemas.android.com/apk/res/android">
  <path android:fillColor="#78B9EB" android:pathData="M369.152,182.945l-7.494,22.316c-8.508,25.338 -36.196,39.115 -61.544,30.607l-18.176,-6.107l-127.011,-42.623c-25.348,-8.508 -39.115,-36.206 -30.617,-61.544l7.494,-22.316c5.155,-15.36 17.358,-26.466 31.807,-30.938c9.398,-2.919 19.749,-3.033 29.737,0.321l145.197,48.741c363.884,129.909

```



```

377.66,157.597 369.152,182.945z"/>
    <path android:fillColor="#5AAAE7" android:pathData="M291.896,221.18l-9.957,8.581l-
127.011,-42.623c-25.348,-8.508 -39.115,-36.206 -30.617,-61.544l7.494,-22.316c5.155,-15.36
17.358,-26.466 31.807,-30.938l291.896,221.18z"/>
    <path android:fillColor="#C70024"

android:pathData="M265.616,233.032l-84.719,73.023l-12.794,11.023l-156.127,-
181.124l12.804,-11.034l84.719,-73.023z"/>
    <path android:fillColor="#FF0C38" android:pathData="M265.616,233.032l-84.719,73.023l-
156.117,-181.135l84.719,-73.023z"/>
    <path android:fillColor="#A5001E" android:pathData="M118.421,62.972l-96.808,83.436c-
4.896,4.213 -12.359,3.664 -16.582,-1.232l-2.194,-2.546c-4.223,-4.896 -3.664,-12.359 1.232,-
16.582l7.659,-6.604l89.139,-76.832c4.854,-4.182 12.203,-3.685 16.447,1.076c0.052,0.052
0.104,0.104 0.145,0.155l2.184,2.546c123.865,51.287 123.317,58.749 118.421,62.972z"/>
    <path android:fillColor="#C70024" android:pathData="M117.313,43.689l-89.149,76.832c-
4.844,4.182 -12.193,3.674 -16.437,-1.076l89.139,-76.832c105.721,38.431 113.07,38.928
117.313,43.689z"/>
    <path android:fillColor="#A5001E"

android:pathData="M129.029,173.516m-45.563,0a45.563,45.563 0,1 1,91.126 0a45.563,45.563
0,1 1,-91.126 0"/>
    <path android:fillColor="#FFCD00" android:pathData="M161.17,171.02l-7.807,-6.142l2.577,-
10.589c1.014,-4.165 -1.541,-8.365 -5.706,-9.379c-4.167,-1.015 -8.364,1.541 -9.379,5.706l-
1.738,7.14c-2.939,-0.709 -6.418,-1.252 -10.212,-1.252c-9.98,0 -23.814,3.744 -33.043,21.582c-
1.754,3.391 -0.783,7.556 2.293,9.82c0.977,0.719 9.883,7.026 21.503,7.026c7.855,0 18.384,-
2.902 27.177,-15.434l4.735,3.725c1.423,1.12 3.114,1.662 4.794,1.662c2.299,0 4.575,-1.016
6.106,-2.963c165.122,178.551 164.54,173.67 161.17,171.02zM119.658,179.404c-1.958,0 -
3.832,-0.349 -5.501,-0.841c3.971,-4.384 8.846,-6.534 14.748,-6.534c1.359,0 2.67,0.115
3.898,0.299c128.972,177.03 124.572,179.404 119.658,179.404z"/>
    <path android:fillColor="#FF728B" android:pathData="M473.853,68.748l367.18,23.831c-
9.128,-3.843 -19.643,0.442 -23.486,9.57l-14.83,35.227c-0.109,0.053 -0.113,0.055 -0.004,0.002l-
43.622,34.848c-1.488,1.188 -2.888,2.485 -4.17,3.867c-3.153,3.37 -5.699,7.284 -7.501,11.565l-
35.641,84.658c-5.676,13.483 0.65,29.009 14.133,34.686l25.131,10.58l-
0.001,0.002l123.506,51.995c13.484,5.676 29.017,-0.652 34.693,-14.136l35.669,-84.724c2.522,-
5.993 3.517,-12.518 2.893,-18.99l-5.354,-55.523l14.828,-35.222c487.266,83.105 482.981,72.59
473.853,68.748z"/>
    <path android:fillColor="#FF3F62" android:pathData="M400.694,300.831l-123.506,-
51.995l45.917,-109.068c2.542,-6.037 6.555,-11.343 11.673,-15.432l43.627,-
34.851l90.187,37.969l5.354,55.526c0.624,6.472 -0.37,12.998 -2.893,18.99l-
35.669,84.724c429.711,300.178 414.178,306.508 400.694,300.831z"/>
    <path android:fillColor="#FF728B" android:pathData="M468.593,127.456l-42.838,35.439c-
1.394,1.156 -2.697,2.395 -3.901,3.72c-3.042,3.301 -5.494,7.119 -7.248,11.285l-
46.055,109.396l252.06,238.253c-13.483,-5.676 -19.809,-21.203 -14.133,-34.686l35.641,-
84.658c1.802,-4.281 4.348,-8.195 7.501,-11.565c1.282,-1.383 2.682,-2.68 4.17,-3.867l43.622,-

```

```

34.848L468.593,127.456z"/>
  <path android:fillColor="#FFA5B5"

android:pathData="M421.853,166.615c-3.042,3.301 -5.494,7.119 -7.248,11.285l-21.856,51.916l-
140.901,-59.319l21.719,-51.588c1.802,-4.281 4.348,-8.195 7.501,-11.565l421.853,166.615z"/>
  <path android:fillColor="#FF0C38" android:pathData="M473.951,182.976l-5.362,-55.52l-
42.83,35.44c-1.397,1.159 -2.701,2.391 -3.902,3.716c-3.043,3.302 -5.496,7.121 -7.256,11.292l-
46.049,109.395l32.138,13.528c13.487,5.682 29.023,-0.652 34.695,-14.128l35.668,-
84.729C473.578,195.977 474.572,189.456 473.951,182.976z"/>
  <path android:fillColor="#FF3F62" android:pathData="M344.639,172.997c0.358,8.991 -
8.499,15.723 -16.942,12.609c-0.193,-0.071 -0.385,-0.147 -0.577,-0.228c-0.191,-0.081 -0.38,-
0.165 -0.565,-0.254c-8.128,-3.862 -9.504,-14.902 -

2.822,-20.93l14.836,-13.384c1.988,-1.794 5.166,-0.455 5.273,2.22l344.639,172.997z"/>
  <path android:fillColor="#FF3F62" android:pathData="M357.238,80.568l-0.455,1.108l-
43.627,34.85c-1.49,1.19 -2.888,2.484 -4.171,3.871c-3.147,3.364 -5.693,7.287 -
7.494,11.562l256.393,240.07l-4.337,-1.822c-13.476,-5.672 -19.811,-21.198 -14.128,-
34.685l35.637,-84.657c1.801,-4.275 4.347,-8.187 7.504,-11.562c1.283,-1.387 2.681,-2.681
4.171,-3.871l43.617,-34.84c0,-0.01 0.01,-0.01 0.01,-0.01l357.238,80.568z"/>
  <path android:fillColor="#FF728B" android:pathData="M309.875,119.476c-0.311,0.3 -
0.6,0.611 -0.89,0.921c-3.147,3.364 -5.693,7.287 -7.494,11.562l-21.084,50.552l-28.557,-
12.017l21.715,-51.587c1.801,-4.275 4.347,-8.187 7.504,-11.562l309.875,119.476z"/>
  <path android:fillColor="#FF3F62"

android:pathData="M468.593,127.461l328.864,68.627l14.83,-35.227c3.843,-9.128 14.358,-
13.413 23.486,-9.57l106.672,44.917c9.128,3.843 13.413,14.358
9.57,23.486L468.593,127.461z"/>
  <path android:fillColor="#FF0C38" android:pathData="M378.53,28.609l-21.291,51.96l-28.371,-
11.945l14.822,-35.223c3.85,-9.129 14.367,-13.414 23.496,-9.574l378.53,28.609z"/>
  <path android:fillColor="#7ED63E" android:pathData="M484.312,238.124l-73.975,227.422c-
3.157,9.698 -12.193,16.271 -22.398,16.271H125.273c-10.164,0 -19.18,-6.521 -22.367,-
16.168l27.688,238.124H484.312z"/>
  <path android:fillColor="#A4E276" android:pathData="M484.312,238.124l-73.975,227.422c-
3.157,9.698 -12.193,16.271 -22.398,16.271H169.584c-10.164,0 -19.18,-6.521 -22.367,-
16.168l72.009,238.124H484.312z"/>
  <path android:fillColor="#5EAC24" android:pathData="M490.588,233.561c-1.46,-2.01 -3.796,-
3.2 -6.281,-3.2H72.424c-4.287,0 -7.763,3.476 -7.763,7.763s3.476,7.763 7.763,7.763h401.195l-
27.982,86.029h-45.888l12.546,-58.945c0.892,-4.194 -1.783,-8.317 -5.976,-9.209c-4.192,-0.894 -
8.317,1.782 -9.209,5.976l-13.234,62.178h-52.298l5.301,-59.878c0.378,-4.271 -2.777,-8.039 -
7.049,-8.417c-4.26,-0.38 -8.039,2.777 -8.417,7.049l-5.422,61.246h-52.229v-60.562c0,-4.287 -
3.476,-7.763 -7.763,-7.763c-4.287,0 -

7.763,3.476 -7.763,7.763v60.562h-50.5l-5.422,-61.246c-0.378,-4.271 -4.156,-7.421 -8.417,-
7.049c-4.271,0.378 -7.426,4.146 -7.049,8.417l5.301,59.878h-9.497c-4.287,0 -7.763,3.476 -
7.763,7.763c0,4.287 3.476,7.763 7.763,7.763h10.871l4.889,55.227H144.91l-11.755,-
55.227h8.446c4.287,0 7.763,-3.476 7.763,-7.763c0,-4.287 -3.476,-7.763 -7.763,-7.763h-11.75l-

```

13.234,-62.178c-0.893,-4.193 -5.012,-6.874 -9.209,-5.976c-4.193,0.892 -6.869,5.015 -  
5.976,9.209l12.546,58.945H66.872l-28.438,-86.029h2.939c4.287,0 7.763,-3.476 7.763,-7.763s-  
3.476,-7.763 -7.763,-7.763h-13.68c-2.492,0 -4.832,1.197 -6.292,3.216c-1.459,2.02 -1.861,4.617 -  
1.079,6.983L95.533,468.09c4.249,12.853 16.199,21.49

29.736,21.49h28.526c0.006,0 0.011,0.001 0.018,0.001c0.009,0 0.02,-0.001 0.029,-  
0.001h49.351c0.003,0 0.007,0 0.01,0c0.004,0 0.007,0 0.011,0h107.298c0.004,0 0.007,0  
0.011,0c0.003,0 0.007,0 0.01,0h49.351c0.009,0 0.02,0.001 0.029,0.001c0.006,0 0.011,-0.001  
0.018,-0.001h28.004c13.606,0 25.575,-8.693 29.784,-21.632l23.045,-70.854c1.326,-4.077 -  
0.905,-8.457 -4.982,-9.783c-4.078,-1.325 -8.456,0.905 -9.783,4.982l-  
3.375,10.378H384.69l11.755,-55.227h44.143l-4.983,15.32c-1.326,4.077 0.905,8.457  
4.982,9.783c0.798,0.259 1.606,0.383 2.402,0.383c3.272,0 6.314,-2.085 7.381,-5.364l41.319,-  
127.039C492.458,238.161 492.049,235.573 490.588,233.561zM314.617,347.442l-4.889,55.227h-  
45.966v-55.227H314.617zM318.995,474.053l4.944,-55.858h41.572l-  
11.89,55.858H318.995zM303.409,474.053h-39.647v-  
55.858h44.591L303.409,474.053zM210.318,474.053l-4.944,-  
55.858h42.864v55.858H210.318zM199.11,347.442h49.127v55.227H204L199.11,347.442zM189.  
788,418.195l4.944,55.858h-34.628l-11.889,-  
55.858H189.788zM117.282,347.442l11.755,55.227H90.26l-18.255,-  
55.227H117.282zM110.274,463.216l-14.882,-  
45.022h36.95l11.89,55.858H125.27C118.443,474.053 112.417,469.698  
110.274,463.216zM417.575,418.195l-14.619,44.949c-2.122,6.525 -8.157,10.908 -  
15.019,10.908h-18.439l11.89,-55.858H417.575zM368.817,402.669h-43.502l4.889,-  
55.227h50.369L368.817,402.669z"/>

<path android:fillColor="#4E901E" android:pathData="M512,238.124c0,9.14 -3.737,17.451 -  
9.761,23.465c-6.024,6.024 -14.335,9.761 -23.475,9.761H33.236c-18.279,0 -33.236,-14.946 -  
33.236,-33.225c0,-9.14 3.737,-17.451 9.761,-23.475c6.024,-6.024 14.335,-9.761 23.475,-  
9.761h445.528c17.658,0 32.221,13.963 33.184,31.383C511.979,236.882 512,237.503  
512,238.124z"/>

<path android:fillColor="#5EAC24" android:pathData="M511.948,236.271H66.42c-17.658,0 -  
32.211,-13.963 -33.173,-31.383h445.518C496.422,204.888 510.985,218.851  
511.948,236.271z"/>

<path android:fillColor="#47821C" android:pathData="M278.543,68.251v157.794c0,5.848 -  
2.256,11.189 -5.941,15.205c-4.13,4.502 -10.05,7.339 -16.602,7.339c-12.4,0 -22.543,-10.144 -  
22.543,-22.543V68.251c0,-12.39 10.144,-22.533 22.543,-22.533c6.2,0 11.841,2.536  
15.919,6.624c0.228,0.228 0.455,0.466 0.683,0.714C276.287,57.062 278.543,62.414  
278.543,68.251z"/>

<path android:fillColor="#4E901E" android:pathData="M278.543,68.251v157.794c0,5.848 -  
2.256,11.189 -

5.941,15.205c-3.695,-4.016 -5.952,-9.357 -5.952,-15.205V68.251c0,-5.838 2.256,-11.189 5.952,-  
15.195C276.287,57.062 278.543,62.414 278.543,68.251z"/>

<path android:fillColor="#A4E276" android:pathData="M255.999,226.666m-  
6.693,0a6.693,6.693 0,1 1,13.386 0a6.693,6.693 0,1 1,-13.386 0"/>

</vector>

### 2.31. Ic\_home\_food.xml

```
<vector android:height="24dp" android:viewportHeight="64"
    android:viewportWidth="64" android:width="24dp"
xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#f9e109" android:pathData="m26,18.664h-4v-8a2,2 0,0 1,0.891 -
1.664l9,-6 2.218,3.328 -8.109,5.406z"/>
    <path android:fillColor="#fcbc04" android:pathData="m34.11,6.33 l-8.11,5.4v6.93h-4v-
8a1.994,1.994 0,0 1,0.89 -1.66l1.01,-0.68a5.35,5.35 0,0 0,4.82 -0.75l4.19,-3.05z"/>
    <path android:fillColor="#f8664f" android:pathData="m35,61h-22l-4,-35h30z"/>
    <path android:fillColor="#84d2f4" android:pathData="m11,22h26a2,2 0,0 1,2 2v2a0,0 0,0 1,0
0h-30a0,0 0,0 1,0 0v-2a2,2 0,0 1,2 -2z"/>
    <path android:fillColor="#84d2f4" android:pathData="m16,18h16a2,2 0,0 1,2 2v2a0,0 0,0 1,0
0h-20a0,0 0,0 1,0 0v-2a2,2 0,0 1,2 -2z"/>
    <path android:fillColor="#cc5144" android:pathData="m38.66,29h-22.66a34.147,34.147 0,0
0,21.53 9.9l-2.53,22.1h-22l-4,-35h30z"/>
    <path android:fillColor="#57b7eb" android:pathData="m39,24v2h-30v-2a2.006,2.006 0,0 1,2 -
2,2.006 2.006,0 0,0 2,2z"/>
    <path android:fillColor="#57b7eb" android:pathData="m34,20v2h-20v-2a2.006,2.006 0,0 1,2 -
2,2.006 2.006,0 0,0 2,2z"/>
    <path android:fillColor="#e59730" android:pathData="m45,23 l-2,25 -1,11 5,1 3,-36 -3,-1.6z"/>
    <path android:fillColor="#fcbc04" android:pathData="m39,28 l2,24h6l-2,-25 -3,-3z"/>
    <path android:fillColor="#e59730" android:pathData="m47,52h-6l-2,-24 1.18,-1.57
1.82,14.57h1l1.09,-6.36 0.91,-0.64 0.61,0.61z"/>
    <path android:fillColor="#f9e109" android:pathData="m37,31 l-2.963,-1.018 -1.037,3.018 5,16
4.667,-2z"/>
    <path android:fillColor="#fcbc04" android:pathData="m42.67,47 l-4.67,2 -5,-16 1.02,-2.95
1.83,5.5a17.733,17.733 0,0 0,5.66 8.18z"/>
    <path android:fillColor="#f9e109" android:pathData="m55,24 l-4.079,19h4.079l3,-14z"/>
    <path android:fillColor="#fcbc04" android:pathData="m54.848,33.325 l1.6,-6.918 -1.448,-
2.407 -4.079,19h4.079l1.363,-6.359a3,3 0,0 1,-1.515 -3.316z"/>
    <path android:fillColor="#f9e109" android:pathData="m27,30 l7,19 4,-1 -7,-18z"/>
    <path android:fillColor="#fcbc04" android:pathData="m27,30 l7,19 2,-0.5 -7.142,-18.5z"/>
    <path android:fillColor="#f9e109" android:pathData="m49,27 l-3,24 4,-1 4,-21z"/>
    <path android:fillColor="#fcbc04" android:pathData="m51.72,40.97 l-1.72,9.03 -4,1 3,-24
1.94,0.78 -0.8,11.23a1.856,1.856 0,0 0,1.58 1.96z"/>
    <path android:fillColor="#f9e109" android:pathData="m48,52h-4l1,-18 4,1z"/>
    <path android:fillColor="#fcbc04" android:pathData="m46.732,42.567 l0.226,-8.077 -1.958,-
0.49 -1,18h4l0.393,-6.674a3,3 0,0 1,-1.661 -2.759z"/>
    <path android:fillColor="#84d2f4" android:pathData="m53,39c0,3.866 -4.029,7 -9,7s-9,-3.134 -
9,-7h-5l3,22h22l3,-22z"/>
    <path android:fillColor="#57b7eb" android:pathData="m55.55,57 l-0.55,4h-22l-3,-
22h5a6.112,6.112 0,0 0,2.07 4.46,15.991 15.991,0 0,0 15.81,13.54z"/>
    <path android:fillColor="#f9e109" android:pathData="M44,54m-3,0a3,3 0,1 1,6 0a3,3 0,1 1,-6
0"/>
```

```

<path android:fillColor="#fcbc04" android:pathData="m46.49,55.67a3,3 0,0 1,-5.49 -1.67,2.947
2.947,0 0,1 0.54,-1.71 15.951,15.951 0,0 4.95 3.38z"/>
<path android:fillColor="#f89c8d" android:pathData="m30,51h-18a4,4 0,0 1,-4 -4,4 0,0 1 4,-
4h18a4,4 0,0 1 4 4,4 0,0 1 -4,4z"/>
<path android:fillColor="#f8664f" android:pathData="m32,49a3.931,3.931 0,0 0,1.64 -
0.36,3.865 3.865,0 0,1 -0.81,1.19 4.025,4.025 0,0 1,-2.83 1.17h-18a4,4 0,0 1,-2.83 -6.83,3.865
3.865,0 0,1 1.19,-0.81 3.931,3.931 0,0 1,1.64 -0.36h18a4,4 0,0 1,2.64 1,3.833 3.833,0 0,1 1.1,1.6
2.116,2.116 0,0 1,-1.85 1.09,2.163 2.163,0 0,1 -0.95,-0.22 2.076,2.076 0,0 0,-1.88 0l-
2.12,1.06a2.076,2.076 0,0 1,-1.88 0l-2.12,-1.06a2.076,2.076 0,0 0,-1.88 0l-2.12,1.06a2.076,2.076
0,0 1,-1.88 0l-2.12,-1.06a2.076,2.076 0,0 0,-1.88 0l-1.61,0.81a3.1,3.1 0,0 1,-0.58 0.21,3.984
3.984,0 0,0 3.13,1.51z"/>
<path android:fillColor="#fcbc04" android:pathData="m6,55h30a0,0 0,0 1,0 0v2a4,4 0,0 1,-4
4h-22a4,4 0,0 1,-4 -4v-2a0,0 0,0 1,0 0z"/>
<path android:fillColor="#e59730" android:pathData="m23.8,55 l-0.8,1 -0.8,-1z"/>
<path android:fillColor="#e59730" android:pathData="m23.8,55 l-0.8,1 -0.8,-1z"/>
<path android:fillColor="#cc5144" android:pathData="m32,55h-22a2,2 0,0 1,-2 -2,2 2,0 0,1 2,-
2h22a2,2 0,0 1,2 2,2 2,0 0,1 -2,2z"/>
<path android:fillColor="#f7f754" android:pathData="m24,56 l-4,-5h8z"/>
<path android:fillColor="#f9e109" android:pathData="m23.657,52.072 l-0.857,-1.072h-2.8l4,5
2.265,-2.831a2.974,2.974 0,0 1,-2.608 -1.097z"/>
<path android:fillColor="#c4f236" android:pathData="m7,43a3.234,3.234 0,0 0,1.789
2.894l0.764,0.382a3.232,3.232 0,0 0,2.894 0l1.609,-0.8a2.112,2.112 0,0 1,1.888
0l2.112,1.056a2.112,2.112 0,0 0,1.888 0l2.112,-1.056a2.112,2.112 0,0 1,1.888
0l2.112,1.056a2.112,2.112 0,0 0,1.888 0l2.112,-1.056a2.112,2.112 0,0 1,1.888 0,2.112 2.112,0
0,0 3.056,-1.892v-0.584z"/>
<path android:fillColor="#fcbc04" android:pathData="m26,33h-10a10,10 0,0 0,-10
10h30a10,10 0,0 0,-10 -10z"/>
<path android:fillColor="#e59730" android:pathData="m36,43h-30a10,10 0,0 1,7.32 -9.63,9
9,0 0,0 8.68,6.63h13.54a9.908,9.908 0,0 1,0.46 3z"/>
<path android:fillColor="#e59730" android:pathData="m36,57a4,4 0,0 1,-4 4h-22a4,4 0,0 1,-4 -
4v-2h2.13a3.992,3.992 0,0 0,3.87 3h21a4.941,4.941 0,0 0,3 -1z"/>
<path android:fillColor="#a1d51c" android:pathData="m35,43v0.58a1.424,1.424 0,0 1,-0.05
0.42h-26.25a3.058,3.058 0,0 0,1.09 0.89l0.76,0.39a3.284,3.284 0,0 0,2.9 0l1.61,-
0.81a2.076,2.076 0,0 1,1.88 0l2.12,1.06a2.076,2.076 0,0 0,1.88 0l2.12,-1.06a2.076,2.076 0,0
1,1.88 0l2.12,1.06a2.076,2.076 0,0 0,1.88 0l2.12,-1.06a2.076,2.076 0,0 1,1.88 0,2.1 2.1,0 0,0
1.86,0.01 2.119,2.119 0,0 1,-1.91 1.21,2.163 2.163,0 0,1 -0.95,-0.22 2.076,2.076 0,0 0,-1.88 0l-
2.12,1.06a2.076,2.076 0,0 1,-1.88 0l-2.12,-1.06a2.076,2.076 0,0 0,-1.88 0l-2.12,1.06a2.076,2.076
0,0 1,-1.88 0l-2.12,-1.06a2.076,2.076 0,0 0,-1.88 0l-1.61,0.81a3.284,3.284 0,0 1,-2.9 0l-0.76,-
0.39a3.21,3.21 0,0 1,-1.79 -2.89z"/>
<path android:fillColor="#f9e109" android:pathData="M23,37m-1,0a1,1 0,1 1,2 0a1,1 0,1 1,-2
0"/>
<path android:fillColor="#f9e109" android:pathData="M19,36m-1,0a1,1 0,1 1,2 0a1,1 0,1 1,-2
0"/>

```

```

    <path android:fillColor="#f9e109" android:pathData="M16,38m-1,0a1,1 0,1 1,2 0a1,1 0,1 1,-2
0"/>
    <path android:fillColor="#f9e109" android:pathData="M31,38m-1,0a1,1 0,1 1,2 0a1,1 0,1 1,-2
0"/>
    <path android:fillColor="#fcbc04" android:pathData="M12,37m-1,0a1,1 0,1 1,2 0a1,1 0,1 1,-2
0"/>
    <path android:fillColor="#f9e109" android:pathData="M27,36m-1,0a1,1 0,1 1,2 0a1,1 0,1 1,-2
0"/>
    <path android:fillColor="#fcbc04" android:pathData="m16.91,38.42a1,1 0,0 1,-1.91 -0.42,1.022
1.022,0 0,1 0.44,-0.83h0.01a8.693,8.693 0,0 0,1.46 1.25z"/>
</vector>

```

## 2.32. Ic\_home\_store.xml

```

<vector android:height="24dp" android:viewportHeight="511.999"
    android:viewportWidth="511.999" android:width="24dp"
xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#D6E5F6" android:pathData="M441.838,150.941L401.594,16.794c-
2.553,-8.512 -10.387,-14.339 -19.273,-14.339H221.344H60.366c-8.886,0 -16.72,5.829 -
19.273,14.339L0.849,150.941C0.286,152.818 0,154.766 0,156.724c0,30.799 16.067,57.901
40.244,73.418v196.183c0,11.113 9.009,20.122 20.122,20.122h160.977h88.631c-5.579,-6.058 -
10.411,-12.812 -14.344,-20.122c-3.402,-6.325 -6.116,-13.067 -8.077,-20.122c-2.334,-8.388 -
3.609,-17.212 -3.609,-26.335c0,-47.35 33.627,-86.987 78.255,-
96.299c6.498,-1.356 13.227,-2.077 20.122,-2.077s13.624,0.72 20.122,2.077v53.427c24.179,-
15.516 40.244,-42.619 40.244,-73.418C442.687,154.766 442.401,152.818 441.838,150.941z"/>
    <path android:fillColor="#FF6243" android:pathData="M441.838,150.941L401.594,16.794c-
2.553,-8.512 -10.387,-14.339 -19.273,-14.339H221.344H60.366c-8.886,0 -16.72,5.829 -
19.273,14.339L0.849,150.941C0.286,152.818 0,154.766 0,156.724c0,30.799 16.067,57.901
40.244,73.418c6.218,3.991 12.965,7.217 20.122,9.537c6.416,2.081 13.156,3.425
20.122,3.957c2.216,0.17 4.448,0.284 6.707,0.284c20.492,0 39.346,-7.122 54.248,-18.998c4.688,-
3.737 8.996,-7.932 12.826,-12.541c3.831,4.609 8.137,8.804 12.827,12.541c14.901,11.877
33.754,18.998 54.247,18.998s39.346,-7.122 54.248,-18.998c4.688,-3.737 8.996,-7.932 12.826,-
12.541c3.831,4.609 8.137,8.804 12.827,12.541c14.901,11.877 33.754,18.998
54.247,18.998c2.259,0 4.491,-0.114 6.707,-0.284c6.966,-0.533 13.706,-1.877 20.122,-
3.957c7.157,-2.319 13.904,-5.547 20.122,-9.537c24.179,-15.516 40.244,-42.619 40.244,-
73.418C442.687,154.766 442.401,152.818 441.838,150.941z"/>
    <path android:fillColor="#BDFDFD" android:pathData="M402.443,325.33c-6.274,-2.321 -
13.051,-3.595 -20.122,-3.595c-7.07,0 -13.848,1.272 -20.122,3.595c-22.165,8.203 -38.011,29.553 -
38.011,54.536c0,9.476 2.294,18.421 6.334,26.335c4.046,7.924 9.844,14.809
16.899,20.122c9.726,7.326 21.814,11.675 34.901,11.675c6.313,0 12.387,-1.025 18.084,-
2.895c23.224,-7.624 40.046,-29.495 40.046,-55.237C440.451,354.885 424.608,333.535

```

```

402.443,325.33z"/>
    <path android:fillColor="#57A4FF" android:pathData="M464.604,433.691c10.158,-15.478
16.091,-33.965 16.091,-53.823c0,-47.35 -33.627,-86.987 -78.252,-96.299c-6.498,-1.356 -13.227,-
2.077 -20.122,-2.077c-6.895,0 -13.624,0.72 -20.122,2.077c-44.628,9.313 -78.255,48.949 -
78.255,96.299c0,9.122 1.274,17.946 3.609,26.335c1.963,7.055 4.675,13.797
8.077,20.122c3.935,7.31 8.765,14.064 14.344,20.122c17.992,19.533 43.759,31.797
72.346,31.797c19.859,0 38.349,-5.935 53.828,-16.095l41.501,41.501c3.929,3.929 9.079,5.894
14.229,5.894c5.15,0 10.3,-1.965 14.228,-5.894c7.858,-7.858 7.858,-20.598 0,-
28.457l464.604,433.691zM400.405,435.103c-5.696,1.871 -11.77,2.895 -18.084,2.895c-13.087,0 -
25.176,-4.349 -34.901,-11.675c-7.055,-5.314 -12.853,-12.198 -16.899,-20.122c-4.041,-7.912 -
6.334,-16.857 -6.334,-26.335c0,-24.984 15.844,-46.333 38.011,-54.536c6.274,-2.323 13.053,-
3.595 20.122,-3.595c7.071,0 13.848,1.273 20.122,3.595c22.165,8.204 38.008,29.554
38.008,54.536c440.451,405.608 423.629,427.481 400.405,435.103z"/>
    <path android:fillColor="#C9DDF4" android:pathData="M167.097,224.922c-4.688,-3.737 -
8.996,-7.932 -12.827,-12.541c-3.83,4.609 -8.137,8.804 -12.826,12.541c-14.902,11.876 -
33.756,18.998 -54.248,18.998c-2.259,0 -4.491,-0.114 -6.707,-0.284c-6.966,-0.534 -13.706,-1.877
-20.122,-3.957c-7.157,-2.321 -13.904,-5.547 -20.122,-9.537v196.183c0,11.113 9.009,20.122
20.122,20.122h160.977V243.92C200.851,243.92 181.998,236.798 167.097,224.922z"/>
    <path android:fillColor="#FFOC38"
android:pathData="M80.489,243.636c2.216,0.17 4.448,0.284 6.707,0.284c20.492,0 39.346,-
7.122 54.248,-18.998c4.688,-3.737 8.996,-7.932 12.826,-12.541c3.831,4.609 8.137,8.804
12.827,12.541c14.901,11.877 33.754,18.998 54.247,18.998V2.455H60.366c-8.886,0 -16.72,5.829
-19.273,14.339L0.849,150.941C0.286,152.818 0,154.766 0,156.724c0,30.799 16.067,57.901
40.244,73.418c6.218,3.991 12.965,7.217 20.122,9.537C66.783,241.758 73.522,243.102
80.489,243.636z"/>
</vector>

```

### 2.33. `ic_instruction.xml`

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:tint="@color/colorIcon"
    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="@android:color/white"
        android:pathData="M21.99,4c0,-1.1 -0.89,-2 -1.99,-2l4,2c-1.1,0 -2,0.9 -2,2v12c0,1.1 0.9,2
2,2h14l4,4 -0.01,-18zM17,14l7,14c-0.55,0 -1,-0.45 -1,-1s0.45,-1 1,-1h10c0.55,0 1,0.45 1,1s-0.45,1
-1,1zM17,11l7,11c-0.55,0 -1,-0.45 -1,-1s0.45,-1 1,-1h10c0.55,0 1,0.45 1,1s-0.45,1 -
1,1zM17,8l7,8c-0.55,0 -1,-0.45 -1,-1s0.45,-1 1,-1h10c0.55,0 1,0.45 1,1s-0.45,1 -1,1z" />
</vector>

```

### 2.34. Ic\_launcher\_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path
        android:fillColor="#3DDC84"
        android:pathData="M0,0h108v108h-108z" />
    <path
        android:fillColor="#00000000"
        android:pathData="M9,0L9,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,0L19,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M29,0L29,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M39,0L39,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M49,0L49,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M59,0L59,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M69,0L69,108"
        android:strokeWidth="0.8"
```



```

        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M79,0L79,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M89,0L89,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M99,0L99,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,9L108,9"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,19L108,19"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,29L108,29"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,39L108,39"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,49L108,49"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"

```

```

        android:pathData="M0,59L108,59"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,69L108,69"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,79L108,79"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,89L108,89"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,99L108,99"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,29L89,29"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,39L89,39"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,49L89,49"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,59L89,59"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />

```

```

<path
    android:fillColor="#00000000"
    android:pathData="M19,69L89,69"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,79L89,79"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M29,19L29,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M39,19L39,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M49,19L49,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M59,19L59,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M69,19L69,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />

```

### 2.35. Ic\_location\_on.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="20dp"
    android:height="20dp"
    android:tint="#FFFFFF"

```

```

    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="@android:color/white"
        android:pathData="M12,2C8.13,2 5,5.13 5,9c0,4.17 4.42,9.92 6.24,12.11 0.4,0.48 1.13,0.48
1.53,0C14.58,18.92 19,13.17 19,9c0,-3.87 -3.13,-7 -7,-7z"
M12,11.5c-1.38,0 -2.5,-1.12 -2.5,-
2.5s1.12,-2.5 2.5,-2.5 2.5,1.12 2.5,2.5 -1.12,2.5 -2.5,2.5z" />
</vector>

```

## 2.36. Ic\_log\_out.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="20dp"
    android:height="20dp"
    android:tint="@color/colorIcon"
    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="@android:color/white"
        android:pathData="M10.79,16.29c0.39,0.39 1.02,0.39 1.41,0l3.59,-3.59c0.39,-0.39 0.39,-
1.02 0,-1.41L12.2,7.7c-0.39,-0.39 -1.02,-0.39 -1.41,0 -0.39,0.39 -0.39,1.02 0,1.41L12.67,11H4c-
0.55,0 -1,0.45 -1,1s0.45,1 1,1h8.67l-1.88,1.88c-0.39,0.39 -0.38,1.03 0,1.41z"
M19,3H5c-1.11,0 -
2,0.9 -2,2v3c0,0.55 0.45,1 1,1s1,-0.45 1,-1V6c0,-0.55 0.45,-1 1,-1h12c0.55,0 1,0.45 1,1v12c0,0.55
-0.45,1 -1,1H6c-0.55,0 -1,-0.45 -1,-1v-2c0,-0.55 -0.45,-1 -1,-1s-1,0.45 -1,1v3c0,1.1 0.9,2
2,2h14c1.1,0 2,-0.9 2,-2V5c0,-1.1 -0.9,-2 -2,-2z" />
</vector>

```

## 2.37. Ic\_maker.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="32dp"
    android:height="32dp"
    android:viewportWidth="32"
    android:viewportHeight="32">
    <path
        android:fillColor="#ED6A23"
        android:fillType="nonZero"
        android:pathData="M16.0391,31.7578C15.5117,30.9609 14.9297,30.1992
14.4609,29.375C12.9453,26.6992 10.9141,24.5078 8.4922,22.6016C4.5078,19.4609
2.8203,15.3125 3.8125,10.3438C4.8438,5.2305 8.1055,2.0195

```

```

13.1758,0.6758C14.1016,0.4258 15.0625,0.4375 16.0117,0.3125C16.0117,2.625
16.0313,4.9141 16.0313,7.2305C13.6641,7.3242 11.8984,8.3984
10.957,10.5703C10.0391,12.6758 10.3984,14.6641 11.8945,16.4141C12.9688,17.668
14.4258,18.1641 16.0313,18.3047C16.0391,22.793 16.0391,27.2734
16.0391,31.7578ZM16.0391,31.7578"
    android:strokeColor="#00000000" />
<path
    android:fillColor="#FFFFFF"
    android:fillType="nonZero"
    android:pathData="M16.0391,18.3125C14.4297,18.168 12.9766,17.6758
11.8984,16.418C10.4063,14.668 10.043,12.6797 10.9609,10.5742C11.9063,8.4063
13.668,7.332 16.0391,7.2383C16.0391,10.9258 16.0391,14.6172
16.0391,18.3125ZM16.0391,18.3125"
    android:strokeColor="#00000000" />
<path
    android:fillColor="#DC5626"
    android:fillType="nonZero"
    android:pathData="M15.9883,31.7266C15.9883,27.3438 15.9922,22.668
15.9922,18.293C20.125,17.168 21.707,15.4883 21.4297,12.1445C21.2188,9.5195
19.043,7.5195 16.0117,7.2188C16.0117,4.9375 16.0117,2.6563
16.0117,0.3125C18.8047,0.3555 21.3008,1.1875 23.4922,2.8828C30.0859,7.9688
30,17.8867 23.3008,22.8438C21.1016,24.4688 19.3047,26.4258
17.9453,28.8047C17.3867,29.7891 16.0391,31.75 15.9883,31.7266ZM15.9883,31.7266"
    android:strokeColor="#00000000" />
<path
    android:fillColor="#FFFFFF"
    android:fillType="nonZero"
    android:pathData="M22.0938,12.8359C22.0938,16.2891 19.2969,19.0859
15.8438,19.0859C12.3906,19.0859 9.5938,16.2891 9.5938,12.8359C9.5938,9.3867
12.3906,6.5859 15.8438,6.5859C19.2969,6.5859 22.0938,9.3867
22.0938,12.8359ZM22.0938,12.8359"
    android:strokeColor="#00000000" />
</vector>

```

## 2.38. Ic\_male.xml

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="508dp"
    android:height="508dp"
    android:viewportWidth="508"
    android:viewportHeight="508">
    <path
        android:pathData="M254,254m-254,0a254,254 0,1 1,508 0a254,254 0,1 1,-508 0"
        android:fillColor="#90DFAA"/>
    <path
        android:pathData="M255.2,363.2c-0.4,0 -0.8,0.4 -1.6,0.4c-0.4,0 -0.8,-0.4 -1.6,-0.4H255.2z"
        android:fillColor="#E6E9EE"/>
    <path
        android:pathData="M458.4,404c-46,62.8 -120.8,104 -204.8,104S95.2,467.2 48.8,404c36,-38.4
84.8,-58.8 125.6,-69.2c-3.6,29.2 11.6,68.4 12,67.2c15.2,-35.2 66.8,-38.4 66.8,-38.4s51.6,2.8
67.2,38.4c0.4,0.8 15.6,-38 12,-67.2C373.6,345.2 422.4,365.6 458.4,404z"
        android:fillColor="#E6E9EE"/>
    <path
        android:pathData="M316.8,308L316.8,308c0,5.2 -3.2,32.8 -61.6,55.2H252c-58.4,-22.4 -61.6,-
50 -61.6,-55.2l0,0c0.4,-10.4 2.8,-20.8 7.2,-30.4c16,18 35.2,30 56,30c20.4,0 40,-11.6 56,-
30C314,287.2 316.8,297.6 316.8,308z"
        android:fillColor="#FFD05B"/>
    <path
        android:pathData="M288.4,372.4L275.6,398h-44l-12.8,-25.6c17.6,-7.6 34.8,-8.8 34.8,-
8.8S271.2,364.8 288.4,372.4z"
        android:fillColor="#F1543F"/>
    <path
        android:pathData="M218,505.6c11.6,1.6 23.6,2.4 36,2.4c12,0 24,-0.8 36,-2.4l-14,-107.2h-
44L218,505.6z"
        android:fillColor="#FF7058"/>
    <path
        android:pathData="M316.8,307.2c0,0 2.8,32 -63.2,56.4c0,0 51.6,2.8 67.2,38.4C321.2,403.6
351.2,326 316.8,307.2z"
        android:fillColor="#FFFFFF"/>
    <path
        android:pathData="M190.4,307.2c-34,18.8 -4.4,96 -3.6,94.8c15.2,-35.2 67.2,-38.4 67.2,-
38.4C187.6,339.2 190.4,307.2 190.4,307.2z"
        android:fillColor="#FFFFFF"/>
    <path
        android:pathData="M312.8,285.6c-16.8,18 -36.8,29.6 -59.2,29.6s-42.4,-11.6 -59.2,-29.6c0.8,-
2.8 2,-5.6 3.2,-8c16,18 35.2,30 56,30s40,-11.6 56,-30C310.8,280 312,282.8 312.8,285.6z"
        android:fillColor="#F9B54C"/>
    <path
        android:pathData="M362.8,224.4c-8.4,14 -21.2,22.4 -30.8,20.8c-19.2,35.6 -47.2,62 -78.4,62s-
```

```

59.2,-26.8 -78.4,-62c-9.6,1.2 -22.4,-6.8 -30.8,-20.8c-10,-16.4 -10.4,-34.4 -0.8,-40.4c2.4,-1.2 4.8,-2
7.6,-1.6c6.4,16.4 15.2,26.4 15.2,26.4c-9.2,-50.8 28.4,-56.4 22,-105.2c0,0 23.6,52.4 91.2,15.6l-
5.2,10c94.4,-21.2 62.8,90 62,92.8c10.8,-13.6

17.6,-27.2 21.6,-39.6c1.6,0 3.6,0.8 4.8,1.6C373.2,189.6 372.8,208 362.8,224.4z"
    android:fillColor="#FFD05B"/>
<path
    android:pathData="M308,50.8c7.6,-0.8 20,6 20,6c-34,-38.8 -89.6,-14 -89.6,-14c18.8,-16 35.6,-
14.4 35.6,-14.4c-79.6,-12 -93.2,35.6 -93.2,35.6c-3.6,-5.6 -3.6,-13.6 -3.2,-17.6C172,56 178,75.2
178,75.2c-5.6,-14 -25.2,-11.6 -25.2,-11.6c16.8,2.8 19.6,13.2 19.6,13.2c-42,15.6 -34.8,59.2 -
34.8,59.2l10,-12c-12.4,47.6 19.2,84.4 19.2,84.4c-9.2,-50.8 28.4,-56.4 22,-105.2c0,0 23.6,52.4
91.2,15.6l-5.2,10c95.6,-21.6 62,93.2 62,93.2c34,-43.2 28.8,-87.6 28.8,-87.6l4,16C380.4,78.4
308,50.8 308,50.8z"
    android:fillColor="#324A5E"/>
</vector>

```

## 2.39. Ic\_no\_food.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="512dp"
    android:height="512dp"
    android:viewportWidth="512"
    android:viewportHeight="512">
    <path
        android:pathData="M157.9,101.5c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0
7.5,3.5 7.5,7.5C165.7,98.1 162.2,101.5 157.9,101.5zM157.9,89.2c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.4
2.2,4.5 4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5C162.4,91.4 160.5,89.2 157.9,89.2z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M404.9,423.1c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0
7.5,3.5 7.5,7.5C412.7,419.7 409.2,423.1 404.9,423.1zM404.9,410.9c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6
2.2,4.5 4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5C409.6,413 407.5,410.9 404.9,410.9z"
        android:fillColor="#FFFFFF"/>
    <path
        android:pathData="M418.5,106c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0 7.5,3.5
7.5,7.5C426.2,102.6 422.8,106 418.5,106zM418.5,93.8c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6 2.2,4.5
4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5C423.2,95.9 421.1,93.8 418.5,93.8z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M477.6,385.6c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0
7.5,3.5 7.5,7.5S481.7,385.6 477.6,385.6zM477.6,373.3c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6 2.2,4.5
4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5S480,373.3 477.6,373.3z"
        android:fillColor="#B9D3EE"/>

```

```

<path
  android:pathData="M28.7,251.9c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0 7.5,3.5
7.5,7.5C36.5,248.4 33,251.9

28.7,251.9zM28.7,239.6c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6 2.2,4.5 4.5,4.5c2.6,0 4.5,-2.2 4.5,-
4.5C33.4,241.7 31.3,239.6 28.7,239.6z"
  android:fillColor="#6F8DD8"/>
<path
  android:pathData="M275.2,74.3H272v-3.2c0,-1.3 -1.1,-2.4 -2.4,-2.4s-2.4,1.1 -
2.4,2.4v3.2H264c-1.3,0 -2.4,1.1 -2.4,2.4s1.1,2.4 2.4,2.4h3.2v3.2c0,1.3 1.1,2.4 2.4,2.4s2.4,-1.1
2.4,-2.4v-3.5h3.2c1.3,0 2.4,-1.1 2.4,-2.4C277.4,75.4 276.5,74.3 275.2,74.3z"
  android:fillColor="#6F8DD8"/>
<path
  android:pathData="M39.5,350.7h-3.2v-3.2c0,-1.3 -1.1,-2.4 -2.4,-2.4c-1.3,0 -2.4,1.1 -
2.4,2.4v3.2h-3.2c-1.3,0 -2.4,1.1 -2.4,2.4s1.1,2.4 2.4,2.4h3.2v3.2c0,1.3 1.1,2.4 2.4,2.4s2.4,-1.1 2.4,-
2.4v-3.2h3.2c1.3,0 2.4,-1.1 2.4,-2.4S40.8,350.7 39.5,350.7z"
  android:fillColor="#6F8DD8"/>
<path
  android:pathData="M482.1,269.8h-3.2v-3.2c0,-1.3 -1.1,-2.4 -2.4,-2.4s-2.4,1.1 -2.4,2.4v3.2h-
3.2c-1.3,0 -2.4,1.1 -2.4,2.4s1.1,2.4 2.4,2.4h3.2v3.2c0,1.3 1.1,2.4 2.4,2.4s2.4,-1.1 2.4,-2.4v-
3.2h3.2c1.3,0 2.4,-1.1 2.4,-2.4S483.4,269.8 482.1,269.8z"
  android:fillColor="#B9D3EE"/>
<path
  android:pathData="M292.5,135c-36.5,11 -57.6,-9.5 -92.5,-15.1c-15.7,-2.4 -38.8,-4.7 -
54.6,0.9c-16.6,6.3 -15.7,17.5 -16.6,34.1c-1.3,37.7 -46.8,21.1 -55.4,57.6c-5.6,22.9 -2.8,41
13.8,56.1c8.6,8.6 15.1,17.5 15.7,30c0.9,24.6 -24.6,61.7 8,78.3c18.1,9.5 42.7,-0.9
62.3,8.6c10.4,4.7 16.6,13.4 27.6,17.5c19,5.6 34.7,0.9 52.2,-5.6c16.6,-6.3 33.2,-12.7 49.8,-
14.2c12.7,-0.9 25.2,1.5 37.1,-1.5c13.4,-3.2 17.5,-11.9 27.6,-18.1c12.7,-7.1 28.5,0 41.8,-4.7c33.2,-
10.4 6.3,-42.7 7.1,-65.6c0.9,-25.2 20.1,-44.2 18.6,-71.8c-2.4,-30.8 -23.3,-38 -47.9,-49c-8.6,-3.9 -
15.1,-8 -20.5,-15.7c-6.3,-8 -9.5,-21.4 -17.5,-27C336.7,119.9 311.3,125 292.5,135"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M381.8,162.1c3.9,0 3.9,-6 0,-6C377.9,155.9 377.9,162.1
381.8,162.1L381.8,162.1z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M436,281.6c5.6,5.6 14.2,-2.8 8.8,-8.6C439.2,267.4 430.6,276
436,281.6L436,281.6z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>

```



```

<path
  android:pathData="M83.5,318.7c8,0 8,-12.3 0,-12.3C75.5,306.4 75.5,318.7
83.5,318.7L83.5,318.7z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M402.5,159.1c8,0 8,-12.3 0,-12.3S394.5,159.1 402.5,159.1L402.5,159.1z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M71.2,299.3c10.4,0 10.4,-16 0,-16S60.8,299.3 71.2,299.3L71.2,299.3z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M436,310.5c7.1,7.3 18.6,-3.7 11.4,-11C440.3,292 428.8,303.2
436,310.5L436,310.5z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M108.7,169.5c10.4,0 10.4,-16 0,-16S98.4,169.5
108.7,169.5L108.7,169.5z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M384.8,149.2c11.9,0 11.9,-18.3 0,-18.3C373.2,130.6 373.2,149.2
384.8,149.2L384.8,149.2z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M64.5,313.6c5.8,0 5.8,-9.3 0,-9.3C58.7,304.3 58.7,313.6
64.5,313.6L64.5,313.6z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"
  android:fillAlpha="0.34"/>
<path
  android:pathData="M455.8,294.2c4.1,4.3 10.8,-2.2 6.5,-6.5C458.2,283.6 451.7,289.8
455.8,294.2L455.8,294.2z"
  android:strokeAlpha="0.34"
  android:fillColor="#D4D5FE"

```

```

        android:fillAlpha="0.34"/>
    <path
        android:pathData="M89.5,178.3c5.8,0 5.8,-9.3 0,-9.3S83.7,178.3 89.5,178.3L89.5,178.3z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M91,161.5c3.9,0 3.9,-6 0,-6S87.1,161.5 91,161.5L91,161.5z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M275.9,129.1c8,0 8,-12.3 0,-12.3C267.9,117.1 267.9,129.1
275.9,129.1L275.9,129.1z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M165.7,407.6c8,0 8,-12.3 0,-12.3C157.9,395.3 157.9,407.6
165.7,407.6L165.7,407.6z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M386.8,384.8c8,0 8,-12.3 0,-12.3S378.8,384.8 386.8,384.8L386.8,384.8z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M257.6,120.1c3.9,0 3.9,-6 0,-6C253.5,113.8 253.5,120.1
257.6,120.1L257.6,120.1z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M147.8,397c3.9,0 3.9,-6 0,-6C143.9,391 143.7,397 147.8,397L147.8,397z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M368.4,392.5c3.9,0 3.9,-6 0,-6C364.6,386.5 364.3,392.5
368.4,392.5L368.4,392.5z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>

```

```

<path
    android:pathData="M367.2,191.8c5.6,0 10.1,-4.6 10.1,-10.1v167c0,-5.6 -4.6,-10.1 -10.1,-
10.1h-52.3c-5.6,-17.9 -21.1,-28.6 -40.3,-29c-5.8,-0.1 -5.8,8.9 0,9c14.5,0.3 25.5,7.6 30.6,20h-45.5c-
5.6,0 -10.1,4.6 -10.1,10.1v14.1c-8,-2.1 -16.3,-3.3 -24.8,-3.3c-5.1,0 -10,0.5 -
14.9,1.3l4.7,5l5,5.3c1.7,-0.1 3.5,-0.3 5.2,-0.3c13.3,0 26.2,3.1 37.7,8.9c2,1 3.9,2 5.8,3.2c2,1.2 4,2.5
5.9,3.9c8,2.5 9 15.4,13.3 21.2,22.1c8.1,12.3 12.8,26.3 13.8,40.9h-
15.7l1.1,1.2l4.2,4.4l5.3,5.6l1.6,1.7c-0.1,-0.6 -0.1,-1.1 -0.2,-1.7h9.6c3.1,0 5.6,-2.5 5.6,-5.6c0,-18.8 -
5.5,-37 -15.8,-52.7c-8.2,-12.4 -19,-22.5 -31.3,-29.8l-0.6,-10.6h-11.9v-12.4h105.3v12.4h356l-
8,145.2l5.3,5.7l5.3,5.7l8,-145.3l367.2,191.8l367.2,191.8z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M128.9,292.7c0,58.5 53.1,58.5 95.8,58.5c18,0 34.8,-0.5 49.2,-3.3c1.8,1.5
4,2.4 6.5,2.4h333l-5.3,-5.6l-5.3,-5.6h-24.1c5,-3.1 9.3,-7.1 12.8,-12l-4,-4.2l-4,-4.2c-4.7,7.7 -
12.3,12.7 -22,15.8c-1.8,0.6 -3.6,1.1 -5.6,1.6c-1.8,0.4 -3.7,0.8 -5.6,1.2c-13.1,2.4 -28.6,2.8 -
45.4,2.8c-48.8,0 -81.5,-3 -84.3,-41.6h268h5.6h5.6h4.9l-1.1,-1.2l-4.1,-4.4l-0.1,-0.1l-1.1,-1.2l-4.2,-
4.4l-0.6,-0.7l-0.5,-0.5h-5.2h155.6c-0.3,-0.9 -0.5,-1.9 -0.5,-2.9s0.2,-1.8 0.4,-2.5h267h0.4l-0.4,-0.5l-
0.7,-0.7l-4.2,-4.4l-1.1,-1.2l-4.2,-4.4l-1.1,-1.2h140.4c2.1,-31.4 21.5,-58.2 48.7,-70.9l185,193l-4,-
4.3c-30.9,15.9 -52.1,48.1 -52.1,85.1c0,3.1 2.5,5.6 5.6,5.6h9.6c-0.2,1.2 -0.4,2.4 -0.4,3.7c0,1.4
0.2,2.7 0.5,4.1h-9.7c131.5,287.1 128.9,289.6 128.9,292.7z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M320.5,292.7c0,-3.1 -2.5,-5.6 -5.6,-5.6h-3.6l9.1,9.7c320.4,295.4
320.5,294.1 320.5,292.7z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M219.5,229.5l-4.1,-4.3l-3.3,6.2l-0.4,1c-1.2,3.8 -0.5,7.8 1.9,11c2.6,3.6
6.9,5.8 11.2,5.8h0.1l0,0c3.4,0 6.9,-1.4 9.5,-3.8l-3.9,-4.1l-3.9,-4.1c-0.6,0.4 -1.2,0.7 -1.8,0.7v5.6v-
5.6c-0.6,0 -1.7,-0.5 -2.2,-1.2c-0.2,-0.3 -0.3,-0.3 -0.5 -0.3,-0.7l1.1,-2.1l219.5,229.5z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M167,233.7c0.7,4.4 3.5,8.3 7.3,10.3c1.9,1 4.2,1.6 6.6,1.6c6.7,0 12,-4.3
13,-10.5l0.1,-28.9l-22.9,16.1l-0.8,0.7c167.5,225.9 166.3,229.7 167,233.7zM182.7,233.6c-0.3,0.7
-1.3,0.7 -1.8,0.7c-0.6,0 -1.1,-0.1 -1.3,-0.3c-0.6,-0.3 -1.3,-1.2 -1.4,-2.1c-0.1,-0.4 0,-0.6 0.1,-0.8l4.5,-
3.1l182.7,233.6z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M264.6,237.4l1.8,1.9l4.2,4.5l1.4,1.5c1.5,-0.2 2.9,-0.6 4.2,-1.3c0,0 0.1,0
0.1,-0.1c3.8,-2 6.5,-5.9 7.2,-10.2c0.7,-4 -0.5,-7.8 -3.3,-10.7l-5.3,-3.7l-5.9,-4.2l-5.9,-4.2l-6.7,-
4.7v22.6l5.9,6.3l264.6,237.4zM272.3,231.1c0.1,0.2 0.2,0.4 0.1,0.8c-0.1,0.9 -0.9,1.8 -1.4,2.1c-
0.2,0.1 -0.5,0.2 -0.9,0.2c-0.2,0 -0.3,0.1 -0.5,0.1c-0.5,0 -1.5,-0.1 -1.8,-0.7v228l2,1.4l272.3,231.1z"
    android:fillColor="#1F4D84"/>
<path

```

```

        android:pathData="M342.2,350.3l10.9,11.6c1.3,1.4 3.1,2.1 4.9,2.1c1.7,0 3.3,-0.6 4.6,-1.8c2.7,-
2.5 2.9,-6.8 0.3,-9.6l-6.5,-6.9l-4.2,-4.5l-5.2,-5.5l-27.7,-29.6l-4.9,-5.2l-7.7,-8.3l-2.4,-2.6l-2.8,-3l-1.6,-
1.7l-5.7,-6l-4.2,-4.4l-1.1,-1.2l-4.2,-4.4l-1.1,-1.2l-6.4,-6.8l-6,-6.4l-6,-6.4l-54.4,-58l-4.6,-4.9l-4.4,-4.7l-
20.7,-22c-2.6,-2.7 -6.8,-2.9 -9.6,-0.3c-2.7,2.6 -2.9,6.8 -
0.3,9.6l16.3,17.4l4.2,4.4l4.2,4.5l22.9,24.4l4.1,4.4l9.5,10.2l5.9,6.3l265,268l1.1,1.2l0.2,0.2l4,4.2l1.
1,1.2l0.9,1l3.2,3.5l1.1,1.2l1.6,1.7l3.4,3.7l1.1,1.2l4.2,4.4l1.1,1.2l4.2,4.4l1.1,1.2l12.8,13.7l4.2,4.5l4.
1,4.4l17,18.1l5.3,5.6l342.2,350.3z"
        android:fillColor="#1F4D84"/>
</vector>

```

## 2.40. Ic\_no\_grocery.xml

```

<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="512dp"
    android:height="512dp"
    android:viewportWidth="512"
    android:viewportHeight="512">
    <path
        android:pathData="M157.9,101.5c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0
7.5,3.5 7.5,7.5c165.7,98.1 162.2,101.5 157.9,101.5zM157.9,89.2c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.4
2.2,4.5 4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5c162.4,91.4 160.5,89.2 157.9,89.2z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M404.9,423.1c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0
7.5,3.5 7.5,7.5c412.7,419.7 409.2,423.1 404.9,423.1zM404.9,410.9c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6
2.2,4.5 4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5c409.6,413 407.5,410.9 404.9,410.9z"
        android:fillColor="#FFFFFF"/>
    <path
        android:pathData="M418.5,106c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0 7.5,3.5
7.5,7.5c426.2,102.6 422.8,106 418.5,106zM418.5,93.8c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6 2.2,4.5
4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5c423.2,95.9 421.1,93.8 418.5,93.8z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M477.6,385.6c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0
7.5,3.5 7.5,7.5s481.7,385.6 477.6,385.6zM477.6,373.3c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6 2.2,4.5
4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5s480,373.3 477.6,373.3z"
        android:fillColor="#B9D3EE"/>
    <path
        android:pathData="M28.7,251.9c-4.3,0 -7.5,-3.5 -7.5,-7.5c0,-4.3 3.5,-7.5 7.5,-7.5c4.3,0 7.5,3.5
7.5,7.5c36.5,248.4 33,251.9 28.7,251.9zM28.7,239.6c-2.6,0 -4.5,2.2 -4.5,4.5c0,2.6 2.2,4.5
4.5,4.5c2.6,0 4.5,-2.2 4.5,-4.5c33.4,241.7 31.3,239.6 28.7,239.6z"
        android:fillColor="#6F8DD8"/>

```

```

<path
    android:pathData="M275.2,74.3H272v-3.2c0,-1.3 -1.1,-2.4 -2.4,-2.4s-2.4,1.1 -
2.4,2.4v3.2H264c-1.3,0 -2.4,1.1 -2.4,2.4s1.1,2.4 2.4,2.4h3.2v3.2c0,1.3 1.1,2.4 2.4,2.4s2.4,-1.1
2.4,-2.4v-3.5h3.2c1.3,0 2.4,-1.1 2.4,-2.4C277.4,75.4 276.5,74.3 275.2,74.3z"
    android:fillColor="#6F8DD8"/>
<path
    android:pathData="M39.5,350.7h-3.2v-3.2c0,-1.3 -1.1,-2.4 -2.4,-
2.4c-1.3,0 -2.4,1.1 -2.4,2.4v3.2h-3.2c-1.3,0 -2.4,1.1 -2.4,2.4s1.1,2.4 2.4,2.4h3.2v3.2c0,1.3 1.1,2.4
2.4,2.4s2.4,-1.1 2.4,-2.4v-3.2h3.2c1.3,0 2.4,-1.1 2.4,-2.4S40.8,350.7 39.5,350.7z"
    android:fillColor="#6F8DD8"/>
<path
    android:pathData="M482.1,269.8h-3.2v-3.2c0,-1.3 -1.1,-2.4 -2.4,-2.4s-2.4,1.1 -2.4,2.4v3.2h-
3.2c-1.3,0 -2.4,1.1 -2.4,2.4s1.1,2.4 2.4,2.4h3.2v3.2c0,1.3 1.1,2.4 2.4,2.4s2.4,-1.1 2.4,-2.4v-
3.2h3.2c1.3,0 2.4,-1.1 2.4,-2.4S483.4,269.8 482.1,269.8z"
    android:fillColor="#B9D3EE"/>
<path
    android:pathData="M292.5,135c-36.5,11 -57.6,-9.5 -92.5,-15.1c-15.7,-2.4 -38.8,-4.7 -
54.6,0.9c-16.6,6.3 -15.7,17.5 -16.6,34.1c-1.3,37.7 -46.8,21.1 -55.4,57.6c-5.6,22.9 -2.8,41
13.8,56.1c8.6,8.6 15.1,17.5 15.7,30c0.9,24.6 -24.6,61.7 8,78.3c18.1,9.5 42.7,-0.9
62.3,8.6c10.4,4.7 16.6,13.4 27.6,17.5c19,5.6 34.7,0.9 52.2,-5.6c16.6,-6.3 33.2,-12.7 49.8,-
14.2c12.7,-0.9 25.2,1.5 37.1,-1.5c13.4,-3.2 17.5,-11.9 27.6,-18.1c12.7,-7.1 28.5,0 41.8,-4.7c33.2,-
10.4 6.3,-42.7 7.1,-65.6c0.9,-25.2 20.1,-44.2 18.6,-71.8c-2.4,-30.8 -23.3,-38 -47.9,-49c-8.6,-3.9 -
15.1,-8 -20.5,-15.7c-6.3,-8 -9.5,-21.4 -17.5,-27C336.7,119.9 311.3,125 292.5,135"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M381.8,162.1c3.9,0 3.9,-6 0,-6C377.9,155.9 377.9,162.1
381.8,162.1L381.8,162.1z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M436,281.6c5.6,5.6 14.2,-2.8 8.8,-8.6C439.2,267.4 430.6,276
436,281.6L436,281.6z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M83.5,318.7c8,0 8,-12.3 0,-12.3C75.5,306.4 75.5,318.7
83.5,318.7L83.5,318.7z"
    android:strokeAlpha="0.34"

```

```

        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
<path
    android:pathData="M402.5,159.1c8,0 8,-12.3 0,-12.3S394.5,159.1 402.5,159.1L402.5,159.1z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M71.2,299.3c10.4,0 10.4,-16 0,-16S60.8,299.3 71.2,299.3L71.2,299.3z"
    android:strokeAlpha="0.34"

    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M436,310.5c7.1,7.3 18.6,-3.7 11.4,-11C440.3,292 428.8,303.2
436,310.5L436,310.5z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M108.7,169.5c10.4,0 10.4,-16 0,-16S98.4,169.5
108.7,169.5L108.7,169.5z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M384.8,149.2c11.9,0 11.9,-18.3 0,-18.3C373.2,130.6 373.2,149.2
384.8,149.2L384.8,149.2z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M64.5,313.6c5.8,0 5.8,-9.3 0,-9.3C58.7,304.3 58.7,313.6
64.5,313.6L64.5,313.6z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"
    android:fillAlpha="0.34"/>
<path
    android:pathData="M455.8,294.2c4.1,4.3 10.8,-2.2 6.5,-6.5C458.2,283.6 451.7,289.8
455.8,294.2L455.8,294.2z"
    android:strokeAlpha="0.34"
    android:fillColor="#D4D5FE"

```

```

        android:fillAlpha="0.34"/>
    <path
        android:pathData="M89.5,178.3c5.8,0 5.8,-9.3 0,-9.3S83.7,178.3 89.5,178.3L89.5,178.3z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M91,161.5c3.9,0 3.9,-6 0,-6S87.1,161.5 91,161.5L91,161.5z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M275.9,129.1c8,0 8,-12.3 0,-12.3C267.9,117.1 267.9,129.1
275.9,129.1L275.9,129.1z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M165.7,407.6c8,0 8,-12.3 0,-12.3C157.9,395.3
157.9,407.6 165.7,407.6L165.7,407.6z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M386.8,384.8c8,0 8,-12.3 0,-12.3S378.8,384.8 386.8,384.8L386.8,384.8z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M257.6,120.1c3.9,0 3.9,-6 0,-6C253.5,113.8 253.5,120.1
257.6,120.1L257.6,120.1z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M147.8,397c3.9,0 3.9,-6 0,-6C143.9,391 143.7,397 147.8,397L147.8,397z"
        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
    <path
        android:pathData="M368.4,392.5c3.9,0 3.9,-6 0,-6C364.6,386.5 364.3,392.5
368.4,392.5L368.4,392.5z"

```

```

        android:strokeAlpha="0.34"
        android:fillColor="#D4D5FE"
        android:fillAlpha="0.34"/>
<path
    android:pathData="M207.1,357c0,2.1 1.7,3.8 3.8,3.8h91.5c2.1,0 3.8,-1.7 3.8,-3.8c0,-2.1 -1.7,-
3.8 -3.8,-3.8h-91.5C208.8,353.2 207.1,354.9 207.1,357z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M226.2,332.8c-2.1,0 -3.8,1.7 -3.8,3.8c0,2.1 1.7,3.8 3.8,3.8h61c2.1,0 3.8,-
1.7 3.8,-3.8c0,-2.1 -1.7,-3.8 -3.8,-3.8H226.2z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M166.8,300.1c0.1,0 0.1,0 0.2,0c2,0 3.7,-1.6 3.8,-3.6l5,-
99.7l11.4,7.2c1.2,0.8 2.8,0.8 4,0l14.9,-9.4l14.9,9.4c1.2,0.8 2.8,0.8 4,0l14.9,-9.4l14.9,9.4c1.2,0.8
2.8,0.8 4,0l14.9,-9.4l14.9,9.4c1.2,0.8 2.8,0.8 4,0l14.9,-9.4l14.9,9.4c1.2,0.8 2.8,0.8 4,0l11.4,-
7.2l0.7,13.8c0.1,2.1 1.9,3.7 4.3,6c2.1,-0.1 3.7,-1.9 3.6,-4l-1,-20.2c-0.1,-1.3 -0.8,-2.5 -2,-3.2c-1.2,-
0.6 -2.6,-0.6 -3.8,0.2l-14.9,9.4l-1.7,-1.1v178c0,-2.1 -1.7,-3.8 -3.8,-3.8c-2.1,0 -3.8,1.7 -
3.8,3.8v12.5l-5.6,-3.5c-1.2,-0.8 -2.8,-0.8 -4,0l-14.9,9.4l-14.9,-9.4c-1.2,-0.8 -2.8,-0.8 -4,0l-11.1,7v-
5.8v-34.1c0,-15 12.2,-27.2 27.2,-27.2s27.2,12.2 27.2,27.2v6.3c0,2.1 1.7,3.8 3.8,3.8c2.1,0 3.8,-1.7
3.8,-3.8v-6.3c0,-19.2 -15.6,-34.8 -34.8,-34.8c-19.2,0 -34.8,15.6 -34.8,34.8v16.5l-3.6,-4c-2.7,-3 -
4.1,-6.9 -4.1,-11v-10.3c2.7,-1.4 4.6,-4.2 4.6,-7.5v-10.1c0,-4.6 -3.8,-8.4 -8.4,-8.4h-35.6c-4.6,0 -
8.4,3.8 -
8.4,8.4v10.1c0,3.2 1.9,6.1 4.6,7.5v10.3c0,4 -1.5,7.9 -4.1,11l-5.1,5.8c-3.9,4.4 -6,10.1 -6,16v6.7l-
12.5,-7.9c-1.1,-0.7 -2.6,-0.8 -3.8,-0.2c-1.2,0.6 -2,1.8 -2,3.2l-5.3,106.2C163.1,298.3 164.7,300
166.8,300.1l166.8,300.1zM205,127.6c0,-0.5 0.4,-0.8 0.8,-0.8h35.6c0.5,0 0.8,0.4
0.8,0.8v10.1c0,0.5 -0.4,0.8 -0.8,0.8h-35.6c-0.5,0 -0.8,-0.4 -0.8,-0.8V127.6zM198.5,177.2l5.1,-
5.8c3.9,-4.4 6,-10.1 6,-16v-9.4h28v9.4c0,5.9 2.1,11.5 6,16l5.1,5.8c2.7,3 4.1,6.9 4.1,11v5.8l-11.1,-
7c-1.2,-0.8 -2.8,-0.8 -4,0l-14.9,9.4l208,187c-1.2,-0.8 -2.8,-0.8 -4,0l-9.7,6.1v-4.9C194.3,184.2
195.8,180.3 198.5,177.2z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M346.7,227.8c-0.1,-2.1 -1.9,-3.7 -4,-3.6c-2.1,0.1 -3.7,1.9 -
3.6,4l7.1,141.2H167.1l2.8,-55.2c0.1,-2.1 -1.5,-3.9 -3.6,-4c-2.1,-0.1 -3.9,1.5 -4,3.6l-3,59.2c-0.1,1
0.3,2 1,2.8c0.7,0.7 1.7,1.2 2.7,1.2h187.1c1,0 2,-0.4 2.7,-1.2c0.7,-0.7 1.1,-1.8 1,-2.8l346.7,227.8z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M268.6,162.7c0,6.8 8.2,11.9 19,11.9s19,-5.1 19,-11.9c0,-6.8 -8.2,-11.9 -
19,-11.9S268.6,155.9 268.6,162.7zM299.2,162.7c0,1.3 -4,4.4 -11.5,4.4s-11.5,-3.1 -11.5,-4.4c0,-
1.3 4,-4.4 11.5,-4.4S299.2,161.4 299.2,162.7z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M256.7,238.3c-20.3,0 -36.8,16.5 -36.8,36.8s16.5,36.8 36.8,36.8s36.8,-

```



```

16.5 36.8,-36.8S277,238.3 256.7,238.3L256.7,238.3zM256.7,304.4c-16.1,0 -29.3,-13.1 -29.3,-
29.3c0,-16.1 13.1,-29.3 29.3,-29.3s29.3,13.1 29.3,29.3C285.9,291.2 272.8,304.4 256.7,304.4z"
    android:fillColor="#1F4D84"/>
<path
    android:pathData="M134,379L379,169"
    android:strokeWidth="8"
    android:fillColor="#00000000"
    android:strokeColor="#1F4D84"/>
</vector>

```

### 3. RAW MODULE

#### 3.1. Map\_style.xml

```

[
  {
    "featureType": "all",
    "elementType": "labels.text.fill",
    "stylers": [
      {
        "color": "#7c93a3"
      },
      {
        "lightness": "-10"
      }
    ]
  },
  {
    "featureType": "administrative.country",
    "elementType": "geometry",
    "stylers": [
      {
        "visibility": "on"
      }
    ]
  },
  {
    "featureType": "administrative.country",
    "elementType": "geometry.stroke",
    "stylers": [
      {
        "color": "#a0a4a5"
      }
    ]
  }
]

```

```

    }
  ]
},
{
  "featureType": "administrative.province",
  "elementType": "geometry.stroke",
  "stylers": [
    {
      "color": "#62838e"
    }
  ]
},
{
  "featureType": "landscape",
  "elementType": "geometry.fill",
  "stylers": [
    {
      "color": "#dde3e3"
    }
  ]
},
{
  "featureType": "landscape.man_made",
  "elementType": "geometry.stroke",
  "stylers": [
    {
      "color": "#3f4a51"
    },
    {
      "weight": "0.30"
    }
  ]
},
{
  "featureType": "poi",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "simplified"
    }
  ]
},

```

```

{
  "featureType": "poi.attraction",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "on"
    }
  ]
},
{
  "featureType": "poi.business",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "off"
    }
  ]
},
{
  "featureType": "poi.government",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "off"
    }
  ]
},
{
  "featureType": "poi.park",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "on"
    }
  ]
},
{
  "featureType": "poi.place_of_worship",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "off"
    }
  ]
},

```

```

{
  "featureType": "poi.school",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "off"
    }
  ]
},
{
  "featureType": "poi.sports_complex",
  "elementType": "all",
  "stylers": [
    {
      "visibility": "off"
    }
  ]
},
{
  "featureType": "road",
  "elementType": "all",
  "stylers": [
    {
      "saturation": "-100"
    },
    {
      "visibility": "on"
    }
  ]
},
{
  "featureType": "road",
  "elementType": "geometry.stroke",
  "stylers": [
    {
      "visibility": "on"
    }
  ]
},
{
  "featureType": "road.highway",
  "elementType": "geometry.fill",
  "stylers": [
    {
      "color": "#bbcacf"
    }
  ]
}

```

```

    }
  ]
},
{
  "featureType": "road.highway",
  "elementType": "geometry.stroke",
  "stylers": [
    {
      "lightness": "0"
    },
    {
      "color": "#bbcacf"
    },
    {
      "weight": "0.50"
    }
  ]
},
{
  "featureType": "road.highway",
  "elementType": "labels",
  "stylers": [
    {
      "visibility": "on"
    }
  ]
},
{
  "featureType": "road.highway",
  "elementType": "labels.text",
  "stylers": [
    {
      "visibility": "on"
    }
  ]
},
{
  "featureType": "road.highway.controlled_access",
  "elementType": "geometry.fill",
  "stylers": [
    {
      "color": "#ffffff"
    }
  ]
}

```

```

    ]
  },
  {
    "featureType": "road.highway.controlled_access",
    "elementType": "geometry.stroke",
    "stylers": [
      {
        "color": "#a9b4b8"
      }
    ]
  },
  {
    "featureType": "road.arterial",
    "elementType": "labels.icon",
    "stylers": [
      {
        "invert_lightness": true
      },
      {
        "saturation": "-7"
      },
      {
        "lightness": "3"
      },
      {
        "gamma": "1.80"
      },
      {
        "weight": "0.01"
      }
    ]
  },
  {
    "featureType": "transit",
    "elementType": "all",
    "stylers": [
      {
        "visibility": "off"
      }
    ]
  },
  {
    "featureType": "water",

```

```

        "elementType": "geometry.fill",
        "stylers": [
            {
                "color": "#a3c7df"
            }
        ]
    }
}
]

```

## 4. VALUE MODULE

### 4.1. Array\_in.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <array name="initial_slider_values">
        <item>0</item>
        <item>2000</item>
    </array>
</resources>

```

### 4.2. Colors\_

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#f06917</color>
    <color name="colorPrimaryDark">#DD5300</color>
    <color name="colorAccent">#03DAC5</color>
    <color name="darkBlue">#1A237E</color>
    <color name="lightBlue">#3F51B5</color>
    <color name="blue400">#42A5F5</color>
    <color name="colorToolbar">#FFFFFF</color>
    <color name="colorText">#000000</color>
    <color name="colorButtonText">#FFFFFF</color>
    <color name="colorIcon">#000000</color>
    <color name="colorDialog">#FFFFFF</color>
    <color name="colorBG">#FFFFFF</color>
    <color name="semiTransparent">#D9FFFFFF</color>
    <color name="light">#EFEFEF</color>
    <color name="white">#FFFFFF</color>
    <color name="black">#000000</color>

```

```

<color name="grey">#808080</color>
<color name="lightGrey">#ADADAD</color>
<color name="light_Grey">#D8D8D8</color>
<color name="extraLightGrey">#EDED</color>
<color name="green">#09AF00</color>
<color name="greenDark">#0B8C04</color>
<color name="red">#FF0000 </color>
<color name="yellow">#FFC107</color>
<color name="blue">#2196F3</color>
<color name="pink">#FF367A</color>
<color name="extraLightBlue">#D8EBFF</color>
<color name="lightColor">#FFC4A1</color>
<color name="darkBlueColor">#273173</color>
<color name="orangeColor">#f17224</color>
<color name="greenColor">#4eb549</color>
<color name="pickDropColorDark">#830036</color>
<color name="pickDropColorLight">#ff4343</color>
<color name="firstColor">#F4D67E</color>
<color name="first1Color">#fceab7</color>
<color name="secondColor">#dd9be1</color>
<color name="second1Color">#fac6fd</color>
<color name="thirdColor">#d3e794</color>
<color name="third1Color">#e0f4a2</color>
<color name="forthColor">#a4a9f8</color>
<color name="forth1Color">#c9ccfe</color>
<color name="fifthColor">#f0ea46</color>
<color name="fifth1Color">#fcf9a1</color>
<color name="sixthColor">#92ca8d</color>
<color name="sixth1Color">#bbefb6</color>
<color name="sixth1Coor">#80BBEFB6</color>
</resources>

```

### 4.3. Dimens\_

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <dimen name="zero_dp">0dp</dimen>
  <dimen name="eight_dp">8dp</dimen>
  <dimen name="sixteen_dp">16dp</dimen>
  <dimen name="twenty_four_dp">24dp</dimen>
  <dimen name="thirty_two_dp">32dp</dimen>
  <dimen name="one_twenty_dp">120dp</dimen>

```



```

<dimen name="twenty_four_sp">24sp</dimen>
<dimen name="sixteen_sp">16sp</dimen>
<dimen name="twenty_five_dp">25dp</dimen>
<dimen name="fifty_dp">50dp</dimen>
<dimen name="four_dp">4dp</dimen>
<dimen name="fourteen_sp">14sp</dimen>
<dimen name="ten_dp">10dp</dimen>
<dimen name="twenty_sp">20sp</dimen>
<dimen name="forty_eight_dp">48dp</dimen>
<dimen name="fifty_six_dp">56dp</dimen>
<dimen name="eighty_dp">80dp</dimen>
<dimen name="forty_dp">40dp</dimen>
<dimen name="sixty_dp">60dp</dimen>
<dimen name="twelve_dp">12dp</dimen>
<dimen name="twelve_sp">12sp</dimen>
<dimen name="hundred_dp">100dp</dimen>
<dimen name="twenty_dp">20dp</dimen>
<dimen name="ten_sp">10sp</dimen>
<dimen name="fourteen_dp">14dp</dimen>
<dimen name="two_dp">2dp</dimen>
<dimen name="eight_sp">8sp</dimen>
<dimen name="six_dp">6dp</dimen>
<dimen name="eighteen_sp">18sp</dimen>
<dimen name="seventy_dp">70dp</dimen>
<dimen name="eighteen_dp">18dp</dimen>
<dimen name="thirty_dp">30dp</dimen>
<dimen name="ninety_dp">90dp</dimen>
<dimen name="viewpager_next_item_visible">12dp</dimen>
<dimen name="viewpager_current_item_horizontal_margin">12dp</dimen>
<dimen name="thirty_five_dp">35dp</dimen>
<dimen name="fifteen_dp">15dp</dimen>
<dimen name="twenty_eight_dp">28dp</dimen>
<dimen name="one_forty_dp">140dp</dimen>
</resources>

```

#### 4.4. String\_in.xml

```

<resources>
  <string name="app_name">Pepens</string>
  <string name="facebook_app_id">264059008531313</string>
  <string name="fb_login_protocol_scheme">fb264059008531313</string>
  <string name="google_map_api_key" translatable="false">AlzaSyBFncgd--
oUYT4qxcKJLHhQLlp4RMly1lk</string>

```

```

<string name="maps_key">"AlzaSyC1VA3PLc12xYqtbgFUOUjkMgAL6_FP_Jc"</string>
<string name="map_id">"11b3ca3269b8a2dd"</string>
<string name="direction_key">"AlzaSyBCgM9tjvcQGAYZCLY0H451bjS5lopNlc"</string>

<string name="phone_number">Mobile Number</string>
<string name="resend">"Didn't get the code? Resend Code"</string>
<string name="started">"Let's get started"</string>
<string name="login_success">You logged in successfully.</string>
<string name="no_internet">Something went wrong, please check your internet
connection.</string>
<string name="try_again">Try Again</string>
<string name="exit">Exit!</string>
<string name="yes">Yes</string>
<string name="no">No</string>
<string name="exit_message">Do you want to exit from Pepens?</string>
<string name="first_name">First Name</string>
<string name="last_name">Last Name</string>
<string name="live_in">Live In</string>
<string name="dob">I was born on</string>
<string name="camera">Camera</string>
<string name="gallery">Gallery</string>
<string name="feed">Feed</string>
<string name="home">Home</string>
<string name="account">Account</string>
<string name="cart">Cart</string>
<string name="offers">Offers</string>
<string name="email">Email</string>
<string name="address">Address</string>
<string name="city">City</string>
<string name="state">State</string>
<string name="country">Country</string>
<string name="zip">Zip Code</string>
<string name="update_profile">Update Profile</string>
<string name="record_not_found">Record not found.</string>
<string name="term"><![CDATA[* Terms & Condition]]></string>
<string name="term_condition"><![CDATA[Terms & Condition]]></string>
<string name="refer_code">Refer Code(Optional)</string>
<string name="profile">Profile</string>
<string name="delivery_charges">Delivery Charges</string>
<string name="taxes">Taxes and Charges</string>
<!-- TODO: Remove or change this placeholder text -->
<string name="hello_blank_fragment">Hello blank fragment</string>
</resources>

```

## 4.5. Styles\_in.xml

```
<resources>
  <!-- Base application theme. -->
  <style name="AppTheme" parent="Theme.MaterialComponents.DayNight.NoActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="android:textColorSecondary">@color/black</item>
    <item name="colorOnSecondary">@color/white</item>
  </style>
  <style name="RestaurantTheme"
parent="Theme.MaterialComponents.DayNight.NoActionBar">
    <item name="colorPrimary">@color/lightBlue</item>
    <item name="colorPrimaryDark">@color/darkBlue</item>
    <item name="colorAccent">@color/colorAccent</item>
    <item name="android:textColorSecondary">@color/black</item>
  </style>
  <style name="MySpinnerDatePickerStyle" parent="android:Theme.Material.Dialog">
    <item name="android:datePickerStyle">@style/MySpinnerDatePicker</item>
  </style>
  <style name="MySpinnerDatePicker" parent="android:Widget.Material.Light.DatePicker">
    <item name="android:datePickerMode">spinner</item>
  </style>
  <style name="BottomNavigationView" parent="@style/TextAppearance.AppCompat.Caption">
    <item name="android:textSize">@dimen/twelve_sp</item>
  </style>

  <style name="BottomNavigationView.Active"
parent="@style/TextAppearance.AppCompat.Caption">
    <item name="android:textSize">@dimen/twelve_sp</item>
  </style>
  <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar"
/>
  <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />
  <style name="ExpandedAppBar" parent="@android:style/TextAppearance.Medium">
    <item name="android:textSize">@dimen/eighteen_sp</item>
    <item name="android:textColor">#00000000</item>
    <item name="android:textStyle">bold</item>
  </style>
  <style name="CollapsedAppBar" parent="@android:style/TextAppearance.Medium">
    <item name="android:textSize">@dimen/eighteen_sp</item>
    <item name="android:textColor">@color/white</item>
    <item name="android:textStyle">bold</item>
  </style>
</resources>
```

```

</style>
<style name="Widget.MaterialComponents.TextInputLayout.OutlinedBox"
parent="Base.Widget.MaterialComponents.TextInputLayout">
    <item name="materialThemeOverlay">
        @style/ThemeOverlay.MaterialComponents.TextInputEditText.OutlinedBox
    </item>
    <item name="boxCollapsedPaddingTop">0dp</item>
</style>
<style name="ThemeOverlay.MaterialComponents.TextInputEditText.OutlinedBox">
    <item
name="editTextStyle">@style/Widget.MaterialComponents.TextInputEditText.OutlinedBox
    </item>
</style>
<style name="Widget.MaterialComponents.TextInputEditText.OutlinedBox"
parent="Base.Widget.MaterialComponents.TextInputEditText" />
<style name="Base.Widget.MaterialComponents.TextInputEditText"
parent="Widget.AppCompat.EditText">
    <item name="android:background">@null</item>
    <item name="android:paddingStart">@dimen/eight_dp</item>
    <item name="android:paddingEnd">@dimen/eight_dp</item>
    <item name="android:textSize">@dimen/fourteen_sp</item>
    <item name="android:paddingLeft">@dimen/eight_dp</item>
    <item name="android:paddingRight">@dimen/eight_dp</item>
    <item name="android:paddingTop">@dimen/eight_dp</item>
    <item name="android:paddingBottom">@dimen/eight_dp</item>
    <item name="android:textAppearance">?attr/textAppearanceSubtitle1</item>
</style>
<!-- Dialog Animation Theme-->
<style name="CustomDialogTheme" parent="@android:style/Theme.Dialog">
    <item name="android:windowAnimationStyle">@style/DialogAnimation</item>
</style>
<style name="DialogAnimation">
    <item name="android:windowEnterAnimation">@anim/scale_up</item>
    <item name="android:windowExitAnimation">@anim/scale_down</item>
</style>
<style name="Widget.App.PickDropButtonStyle"
parent="Widget.MaterialComponents.Button">
    <item name="materialThemeOverlay">@style/ButtonThemeOverlay</item>
</style>
<style name="ButtonThemeOverlay">
    <item name="colorPrimary">@color/pickDropColorDark</item>
</style>
<style name="VegSwitch" parent="Theme.AppCompat.Light">
    <item name="colorControlActivated">@color/green</item>
    <item name="colorSwitchThumbNormal">#E6E6E6</item>

```

```

        <item name="android:colorForeground">#42221f1f</item>
    </style>
    <style name="NonVegSwitch" parent="Theme.AppCompat.Light">
        <item name="colorControlActivated">@color/red</item>
        <item name="colorSwitchThumbNormal">#E6E6E6</item>
        <item name="android:colorForeground">#42221f1f</item>
    </style>
</resources>

```

## 5. WORKING (BACK-END)

### 5.1. Location\_Activity

```

package com.app.pepens.activity
import android.Manifest
import android.app.Activity
import android.app.Dialog
import android.content.Context
import android.content.Intent
import android.content.IntentSender
import android.content.SharedPreferences
import android.content.pm.PackageManager
import android.graphics.Color
import android.graphics.drawable.ColorDrawable
import android.location.Address
import android.location.Geocoder
import android.location.Location
import android.location.LocationManager
import android.net.Uri
import android.os.Build
import android.os.Bundle
import android.os.Handler
import android.provider.Settings
import android.util.Log
import android.view.View
import android.view.Window
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast

```

```

import androidx.appcompat.app.AppCompatActivity
import androidx.constraintlayout.widget.ConstraintLayout
import androidx.core.app.ActivityCompat
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.app.pemens.R
import com.app.pemens.adapter.ItemsAdapter
import com.app.pemens.fragment.HomeFragment
import com.app.pemens.fragment.PickAndDrop.PickAndDropFragment
import com.app.pemens.fragment.PlaceOrderFragment
import com.app.pemens.fragment.Restaurants.PlaceRestaurantOrderFragment
import com.app.pemens.fragment.Restaurants.RestaurantsFragment
import com.app.pemens.fragment.ShoppingFragment
import com.app.pemens.model.AvailableLocations.AvailableLocation
import com.app.pemens.model.CartItemsList
import com.app.pemens.model.Coordinates.GeoCoordinates
import com.app.pemens.model.ItemsList
import com.app.pemens.model.LogData
import com.app.pemens.model.Login.Login
import com.app.pemens.model.RestaurantCartList
import com.app.pemens.utils.Url
import com.app.pemens.utils.Utils
import com.app.expensemanager.util.VolleySingleton
import com.android.volley.DefaultRetryPolicy
import com.android.volley.Request
import com.android.volley.toolbox.JsonObjectRequest
import com.android.volley.toolbox.StringRequest
import com.google.android.gms.common.ConnectionResult
import com.google.android.gms.common.api.GoogleApiClient
import com.google.android.gms.common.api.ResolvableApiException
import com.google.android.gms.location.*
import com.google.android.libraries.maps.CameraUpdateFactory
import com.google.android.libraries.maps.GoogleMap
import com.google.android.libraries.maps.OnMapReadyCallback
import com.google.android.libraries.maps.SupportMapFragment
import com.google.android.libraries.maps.model.CircleOptions
import com.google.android.libraries.maps.model.LatLng
import com.google.android.libraries.places.api.Places
import com.google.android.libraries.places.api.model.Place
import com.google.android.libraries.places.widget.AutoComplete

```

```

import com.google.android.libraries.places.widget.AutoCompleteActivity
import com.google.android.libraries.places.widget.model.AutoCompleteActivityMode
import com.google.android.material.dialog.MaterialAlertDialogBuilder
import com.google.gson.Gson
import kotlinx.android.synthetic.main.activity_location.*
import org.json.JSONObject
import kotlin.math.roundToInt

class LocationActivity : AppCompatActivity(), View.OnClickListener,
OnMapReadyCallback,
    GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener,
    LocationListener {
    private val TAG = LocationActivity::class.java.simpleName
    private lateinit var sharedPreferences: SharedPreferences
    private lateinit var progressDialog: Dialog
    private var login: Login? = null
    private var latitude: Double = 28.6139
    private var longitude: Double = 77.2090
    private var mMap: GoogleMap? = null
    private var mapFragment: SupportMapFragment? = null

    private var geoCoder: Geocoder? = null
    private var addressList: List<Address>? = null
    private var address: String? = null
    private var partnerId: Int = 0
    private var areaid: Int = 0
    private var knownName: String? = null
    private var locality: String? = null
    private var zipcode: String? = null
    private var city: String? = null
    private var state: String? = null
    private var country: String? = null
    private val AUTOCOMPLETE_REQUEST_CODE = 1
    private var zoom = 12F
    private val REQUEST_CODE = 101
    private val REQUEST_LOCATION = 102
    private var mGoogleApiClient: GoogleApiClient? = null
    private var location: Location? = null
    private var locationRequest: LocationRequest? = null
    private var locationManager: LocationManager? = null

```

```

private var oldAreaId = 0
private var accuracy = 0f
private var usingCurrentLocation = false
private var updateLocationPermission = false
private var availableLocation: AvailableLocation? = null
private var availableDialog: Dialog? = null
private var ivHeaderBack: ImageView? = null
private var tvHeader: TextView? = null
private var recyclerViewAvailable: RecyclerView? = null
private var availableArrayList: ArrayList<ItemsList> = ArrayList()
private var availableAdapter: ItemsAdapter? = null
companion object {
    var tvConnection: TextView? = null
}
fun dialog(value: Boolean) {
    tvConnection!!.visibility = View.VISIBLE
    if (value) {
        tvConnection!!.setBackgroundColor(Color.parseColor("#09AF00"))
        tvConnection!!.text = "Back Online."
        Handler().postDelayed({
            tvConnection!!.visibility = View.GONE
        }, 2000)
    } else {
        tvConnection!!.setBackgroundColor(Color.RED)
        tvConnection!!.text = "No connection."
    }
}
}
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_location)
//    Utils().restartAppAtCrash(this, LocationActivity::class.java.simpleName)
    tvConnection = findViewById(R.id.tv_connection)
    sharedPreferences = getSharedPreferences("USER_DATA",
Context.MODE_PRIVATE)
    progressDialog = Utils().progressDialog(this)
    mGoogleApiClient = GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)

```



```

        .build()
        Places.initialize(applicationContext, resources.getString(R.string.maps_key))
        geoCoder = Geocoder(this@LocationActivity)
        mapFragment = supportFragmentManager.findFragmentById(R.id.map) as
SupportMapFragment
        if (intent != null) {
            login = intent.getParcelableExtra("loginUser")
            Log.e(
                TAG,
                "onCreate: ${login!!.loginData!!.latitude}    ${login!!.loginData!!.longitude}"
            )
            if (login != null) {
                if (login!!.loginData!!.latitude != null && login!!.loginData!!.longitude != null) {
                    if (!login!!.loginData!!.latitude.isNullOrEmpty() &&
!login!!.loginData!!.longitude.isNullOrEmpty() && login!!.loginData!!.latitude != "0" &&
login!!.loginData!!.longitude != "0" && login!!.loginData!!.latitude != "0.0" &&
login!!.loginData!!.longitude != "0.0") {
                        zoom = 12F
                        latitude = login!!.loginData!!.latitude!!.toDouble()
                        longitude =
login!!.loginData!!.longitude!!.toDouble()
                        arealId = login!!.loginData!!.arealId.toInt()
                        partnerId = login!!.loginData!!.partnerId
                        if (tvAccuracy != null) {
                            mapFragment!!.getMapAsync(this)
                        }
                    } else {
                        checkLocation()
                    }
                } else {
                    checkLocation()
                }
            } else if (MainActivity.login != null) {
                zoom = 16F
                latitude = MainActivity.login!!.loginData!!.latitude!!.toDouble()
                longitude = MainActivity.login!!.loginData!!.longitude!!.toDouble()
                arealId = MainActivity.login!!.loginData!!.arealId.toInt()
                partnerId = MainActivity.login!!.loginData!!.partnerId
                if (tvAccuracy != null) {
                    mapFragment!!.getMapAsync(this)
                }
            }
        }
    }
}

```

```

        }
    } else {
        checkLocation()
    }
} else if (MainActivity.login != null) {
    zoom = 16F
    latitude = MainActivity.login!!.loginData!!.latitude!!.toDouble()
    longitude = MainActivity.login!!.loginData!!.longitude!!.toDouble()
    areald = MainActivity.login!!.loginData!!.areald.toInt()
    partnerId = MainActivity.login!!.loginData!!.partnerId
    if (tvAccuracy != null) {
        mapFragment!!.getMapAsync(this)
    }
} else {
    checkLocation()
}
checkAvailableLocations()
ivBack.setOnClickListener(this)
ivAutoDetect.setOnClickListener(this)
ivAutoDetectMSG.setOnClickListener(this)
btnChange.setOnClickListener(this)
btnAuto.setOnClickListener(this)
tvAvailable.setOnClickListener(this)
tvSave.setOnClickListener(this)
}

override fun onResume() {
    super.onResume()
    Utils().updatePagesLog(
        this,
        LogData(
            pageId = Urls.STARTUP_LOCATION_PAGE,
            title = "Update location",
            eventName = "Page Open"
        )
    )

    if (updateLocationPermission) {
        checkLocation()
    }
}

```

```

}
override fun onClick(v: View?) {
    when (v!!.id) {
        R.id.ivBack -> {
            finish()
        }
        R.id.ivAutoDetect -> {
            checkLocation()
        }
        R.id.ivAutoDetectMSG -> {
            checkLocation()
        }
        R.id.btnAuto -> {
            checkLocation()
        }
        R.id.btnChange -> {
            tvAccuracy.visibility = View.GONE
            progressBar.visibility = View.GONE
            usingCurrentLocation = false
            if (mGoogleApiClient!!.isConnected) {
                mGoogleApiClient!!.disconnect()
            }
            val fields = listOf(
                Place.Field.ID,
                Place.Field.NAME,
                Place.Field.LAT_LNG,
                Place.Field.ADDRESS,
                Place.Field.TYPES
            )
            val intent =
Autocomplete.IntentBuilder(AutocompleteActivityMode.FULLSCREEN, fields)
                .setCountry("IN").build(this)
            startActivityForResult(intent, AUTOCOMPLETE_REQUEST_CODE)
        }
        R.id.tvAvailable -> {
            tvAccuracy.visibility = View.GONE
            progressBar.visibility = View.GONE
            usingCurrentLocation = false
            if (mGoogleApiClient!!.isConnected) {
                mGoogleApiClient!!.disconnect()
            }
        }
    }
}

```

```

    }
    if (availableLocation != null) {
        dialogAvailable()
    } else {
        Toast.makeText(
            this,
            "Currently we are not available in this location",
            Toast.LENGTH_SHORT
        ).show()
    }
}
R.id.tvSave -> {
    if (areaId > 0 && partnerId > 0) {
        progressDialog.show()
        updatePinCode()
    } else {
        Toast.makeText(this, "Unable to save, please change or move location",
Toast.LENGTH_SHORT).show()
    }
}
}
}
private fun checkLocation() {
    tvErrorMsg.visibility = View.GONE
//    btnSave.isEnabled = false
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (checkSelfPermission(Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(
                this,
                arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
                REQUEST_CODE
            )
        } else {
            permission()
        }
    } else {
        permission()
    }
}
}

```

```

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    if (requestCode == AUTOCOMPLETE_REQUEST_CODE) {
        when (resultCode) {
            Activity.RESULT_OK -> {
                data?.let {
                    val place = Autocomplete.getPlaceFromIntent(data)
                    address = place.address
                    latitude = place.latLng!!.latitude
                    longitude = place.latLng!!.longitude
                    progressBar.visibility = View.GONE
                    tvAccuracy.visibility = View.GONE
                    usingCurrentLocation = false
                    progressDialog.show()
                    if (tvAccuracy != null) {
                        mapFragment!!.getMapAsync(this)
                    }
                }
            }
            AutocompleteActivity.RESULT_ERROR -> {
                data?.let {
                    val status = Autocomplete.getStatusFromIntent(data)
                    Log.e(TAG, status.statusMessage!!)
                }
            }
            Activity.RESULT_CANCELED -> {
                Log.e(TAG, "onActivityResult: Cancel")
            }
        }
        return
    } else if (requestCode == REQUEST_LOCATION) {
        when (resultCode) {
            Activity.RESULT_OK -> {
                usingCurrentLocation = true
                progressBar.visibility = View.VISIBLE
                if (mGoogleApiClient!!.isConnected) {
                    mGoogleApiClient!!.disconnect()
                }
                mGoogleApiClient!!.connect()
            }
        }
    }
}

```

```

        Activity.RESULT_CANCELED -> {
            AlertDialog.Builder(this)
                .setTitle("Location Permission")
                .setMessage("The app needs location permissions. Please grant location
permission to continue using the features of the app. Tap on 'YES' to grant permission
from setting.")
                .setCancelable(false)
                .setPositiveButton(R.string.yes) { dialog, id ->
                    checkLocation()
                }.show()
        }
    }
}
super.onActivityResult(requestCode, resultCode, data)
}

```

```

override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<out String>,
    grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    when (requestCode) {
        REQUEST_CODE -> {
            if (grantResults.isNotEmpty() && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                Log.e(TAG, "onRequestPermissionsResult: 1")
                permission()
            } else {
                Log.e(TAG, "onRequestPermissionsResult: 2")
                updateLocationPermission = false
                AlertDialog.Builder(this)
                    .setTitle("Location Permission")
                    .setMessage("The app needs location permissions. Please grant location
permission to continue using the features of the app. Tap on 'YES' to grant permission
from setting.")
                    .setCancelable(false)
                    .setPositiveButton(R.string.yes) { dialog, id ->
                        updateLocationPermission = true
                    }
            }
        }
    }
}

```

```

startActivity(Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS).apply {
    data = Uri.fromParts("package", "com.app.pepens", null)
})
}.show()
}
}
}
}
private fun permission() {
    locationManager = getSystemService(Context.LOCATION_SERVICE) as
LocationManager
    var hasGps =
locationManager!!.isProviderEnabled(LocationManager.GPS_PROVIDER)
    if (!hasGps) {
        locationRequest = LocationRequest.create()
            .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
            .setInterval(6000)
        var builder =
LocationSettingsRequest.Builder().addLocationRequest(locationRequest!!)
        var result =
LocationServices.getSettingsClient(this).checkLocationSettings(builder.build())
        result.addOnSuccessListener { locationSettingsResponse ->

    }
        result.addOnFailureListener { exception ->
            if (exception is ResolvableApiException) {
                try {
                    exception.startResolutionForResult(this, REQUEST_LOCATION)
                } catch (sendEx: IntentSender.SendIntentException) {
                    // Ignore the error.
                }
            }
        }
    } else {
        usingCurrentLocation = true
        progressBar.visibility = View.VISIBLE
        if (mGoogleApiClient!!.isConnected) {
            mGoogleApiClient!!.disconnect()
        }
    }
}

```

```

        progressDialog.show()
        mGoogleApiClient!!.connect()
    }
}

override fun onConnected(p0: Bundle?) {
    if (ActivityCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_FINE_LOCATION
    ) != PackageManager.PERMISSION_GRANTED
    ) {
        // TODO: Consider calling
        //   ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //   public void onRequestPermissionsResult(int requestCode, String[]
permissions,
        //                                   int[] grantResults)
        // to handle the case where the user grants the permission. See the
documentation
        // for ActivityCompat#requestPermissions for more details.
        return
    }
    if (mGoogleApiClient != null) {
        location = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient)
    }
    startLocationUpdates()
}

private fun startLocationUpdates() {
    locationRequest =
        LocationRequest.create().setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
            .setInterval(6000)
    if (ActivityCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_FINE_LOCATION
    ) != PackageManager.PERMISSION_GRANTED
    ) {
        // TODO: Consider calling
        //   ActivityCompat#requestPermissions
        // here to request the missing permissions, and then overriding
        //   public void onRequestPermissionsResult(int requestCode, String[] permissions,

```



```

int[] grantResults)
    // to handle the case where the user grants the permission. See the
documentation
    // for ActivityCompat#requestPermissions for more details.
    return
}
LocationServices.FusedLocationApi.requestLocationUpdates(
    mGoogleApiClient,
    locationRequest,
    this
)
}
override fun onConnectionSuspended(p0: Int) {
    Log.d("TAG", "Connection Suspended $p0")
    progressDialog.dismiss()
    progressBar.visibility = View.GONE
    if (mGoogleApiClient != null) {
        mGoogleApiClient!!.connect()
    }
}
override fun onConnectionFailed(p0: ConnectionResult) {
    Log.d("TAG", "Connection failed. Error: " + p0.errorCode)
    progressDialog.dismiss()
    progressBar.visibility = View.GONE
    if (mGoogleApiClient != null) {
        mGoogleApiClient!!.connect()
    }
}
override fun onLocationChanged(p0: Location?) {
    Log.d("requestMain1", "--->>>>  ${p0!!.latitude}, ${p0.longitude} ${p0.accuracy}")
    latitude = p0.latitude
    longitude = p0.longitude
    accuracy = p0.accuracy

    if (tvAccuracy != null) {
        progressDialog.dismiss()
        mapFragment!!.getMapAsync(this)
    }
}
}

```

```

        override fun onMapReady(googleMap: GoogleMap?) {
//      btnSave.isEnabled = false
        mMap = googleMap
        var radius = accuracy.toDouble()
        zoom = when {
            accuracy < 100.0 -> {
                18F
            }
            accuracy < 300.0 -> {
                16F
            }
            accuracy < 600.0 -> {
                14.5F
            }
            accuracy < 1500.0 -> {
                14F
            }
            else -> {
                13F
            }
        }
        Log.e(TAG, "onMapReady: $accuracy $radius")
        mMap!!.clear()
        if (usingCurrentLocation) {
            if (accuracy > 0 && accuracy <= Urls.MINIMUM_ACCURACY) {
                usingCurrentLocation = false
                progressBar.visibility = View.VISIBLE
                tvAccuracy.visibility = View.GONE
//      btnSave.isEnabled = true
                if (mGoogleApiClient != null) {
                    if (mGoogleApiClient!!.isConnected) {
                        progressBar.visibility = View.GONE
                        mGoogleApiClient!!.disconnect()
                    }
                }
            } else {
                tvAccuracy.visibility = View.VISIBLE
                tvAccuracy.text =
                    "Please wait finding accurate location. Current accuracy is
                    ${accuracy.roundToInt()}m"
            }
        }
    }
}

```

```

        tvErrorMsg.visibility = View.GONE
//        btnSave.isEnabled = false
    }
} else {
    progressBar.visibility = View.GONE
    tvAccuracy.visibility = View.GONE
    tvErrorMsg.visibility = View.GONE
//    btnSave.isEnabled = true
}
var add = LatLng(latitude, longitude)
mMap!!.addCircle(
    CircleOptions()
        .center(LatLng(latitude, longitude))
        .radius(radius)
        .strokeWidth(1.5f)
        .strokeColor(0x663F51B5)
        .fillColor(0x333F51B5)
)
mMap!!.moveCamera(CameraUpdateFactory.newLatLngZoom(add, zoom))
mMap!!.setOnCameraMoveStartedListener { it ->
    when (it) {
        GoogleMap.OnCameraMoveStartedListener.REASON_GESTURE -> {
            Log.e(TAG, "onMapReady: The user gestured on the map.")
            usingCurrentLocation = false
            progressBar.visibility = View.GONE
            tvAccuracy.visibility = View.GONE
//            btnSave.isEnabled = false
            if (mGoogleApiClient != null) {
                if (mGoogleApiClient!!.isConnected) {
                    mGoogleApiClient!!.disconnect()
                }
            }
        }
        GoogleMap.OnCameraMoveStartedListener.REASON_API_ANIMATION -> {
            Log.e(TAG, "onMapReady: The user tapped something on the map.")
            usingCurrentLocation = false
            progressBar.visibility = View.GONE
            tvAccuracy.visibility = View.GONE

```

```

//      btnSave.isEnabled = false
      if (mGoogleApiClient != null) {
        if (mGoogleApiClient!!.isConnected) {
          mGoogleApiClient!!.disconnect()
        }
      }
    }
    GoogleMap.OnCameraMoveStartedListener.REASON_DEVELOPER_ANIMATION
-> {
      Log.e(TAG, "onMapReady: The app moved the camera.")
    }
  }
}
mMap!!.setOnCameraIdleListener {
  mMap!!.clear()
  val midLatLng: LatLng = mMap!!.cameraPosition.target
  try {
    try {
      latitude = midLatLng.latitude
      longitude = midLatLng.longitude
      addressList = geoCoder!!.getFromLocation(latitude, longitude, 3)
    } catch (e: Exception) {
      latitude = midLatLng.latitude
      longitude = midLatLng.longitude
      addressList = null
    }
    Log.e(TAG, "onMapReady:- $accuracy $radius")
    mMap!!.addCircle(
      CircleOptions()
        .center(LatLng(latitude, longitude))
        .radius(radius)
        .strokeWidth(1.5f)
        .strokeColor(0x663F51B5)
        .fillColor(0x333F51B5)
    )
    if (addressList != null && addressList!!.isNotEmpty()) {
      var addressPosition = 0
      var addressType = 0
      var p =
        addressList!!.indexOfFirst { item -> !item.thoroughfare.isNullOrEmpty() }
    }
  }
}

```

```

    if (p > -1) {
        addressPosition = p
        addressType = 1
    } else {
        var p =
            addressList!!.indexOfFirst { item -> !item.featureName.isNullOrEmpty() }
        if (p > -1) {
            addressPosition = p
            addressType = 2
        } else {
            addressPosition = 0
            addressType = 0
        }
    }
    when (addressType) {
        0 -> {
            address =
                "${addressList!![addressPosition].subAdminArea},
                ${addressList!![addressPosition].adminArea},
                ${addressList!![addressPosition].countryName},
                ${addressList!![addressPosition].postalCode}"
        }
        1 -> {
            address =
                "${addressList!![addressPosition].thoroughfare},
                ${addressList!![addressPosition].subAdminArea},
                ${addressList!![addressPosition].adminArea},
                ${addressList!![addressPosition].countryName},
                ${addressList!![addressPosition].postalCode}"
        }
        2 -> {
            address =
                "${addressList!![addressPosition].featureName},
                ${addressList!![addressPosition].subAdminArea},
                ${addressList!![addressPosition].adminArea},
                ${addressList!![addressPosition].countryName},
                ${addressList!![addressPosition].postalCode}"
        }
    }
}

```

```

address = address!!.replace(
    "Unnamed Road",
    "${addressList!![addressPosition].locality}"
)
address = address!!.replace("null,", "")
locality = if
(!addressList!![addressPosition].locality.isNullOrEmpty()) {
    addressList!![addressPosition].locality
} else {
    null
}
city = if (!addressList!![addressPosition].subAdminArea.isNullOrEmpty()) {
    addressList!![addressPosition].subAdminArea
} else {
    null
}
if (city.isNullOrEmpty()) {
    city = if (!addressList!![addressPosition].locality.isNullOrEmpty()) {
        addressList!![addressPosition].locality
    } else {
        null
    }
}
state = if (!addressList!![addressPosition].adminArea.isNullOrEmpty()) {
    addressList!![addressPosition].adminArea
} else {
    null
}
country = if (!addressList!![addressPosition].countryName.isNullOrEmpty()) {
    addressList!![addressPosition].countryName
} else {
    null
}
zipcode = if (!addressList!![addressPosition].postalCode.isNullOrEmpty()) {
    addressList!![addressPosition].postalCode
} else {
    null
}

```

```

        knownName = if
(!addressList!![addressPosition].thoroughfare.isNullOrEmpty()) {
            addressList!![addressPosition].thoroughfare
        } else {
            null
        }
        tvAddress.text = "$address"
        checkGeoCoordinates()
    } else {
        locality = null
        progressDialog.dismiss()
        Toast.makeText(
            this,
            "Unable to fetch location please select location again.",
            Toast.LENGTH_SHORT
        ).show()
        tvSave.isEnabled = false
        tvSave.setBackgroundColor(resources.getColor(R.color.grey))
    }
} catch (e: Exception) {
    Log.e(TAG, "onMapReady: $e")
}
}
}

override fun onPause() {
    super.onPause()

    if (mGoogleApiClient != null) {
        if (mGoogleApiClient!!.isConnected) {
            progressDialog.dismiss()
            mGoogleApiClient!!.disconnect()
        }
    }
}

private fun checkGeoCoordinates() {
    val request = StringRequest(Request.Method.GET,
        "${Urls.GET_AREA_BY_GEOCORDINATES}$latitude&long=$longitude",

```

```

{ response ->
    Log.e(TAG, "checkGeoCoordinates: $response")
    if (tvAddress != null) {
        Utils().updatePagesLog(
            this,
            LogData(
                pageId = Urls.STARTUP_LOCATION_PAGE,
                title = "Check Location",
                eventName = "Location Change",
                note = "$response"
            )
        )
        progressDialog.dismiss()
        val jsonObject =
            JSONObject(response.toString())
            if (jsonObject.getBoolean("IsSuccess")) {
                var geoCoordinates =
                    Gson().fromJson(response.toString(), GeoCoordinates::class.java)
                partnerId = geoCoordinates!!.geoCoordinatesData!!.partnerId
                arealId = geoCoordinates!!.geoCoordinatesData!!.arealId
                tvErrorMsg.visibility = View.GONE
                tvAvailable.visibility = View.GONE

                if (accuracy > Urls.MINIMUM_ACCURACY && usingCurrentLocation) {
                    tvSave.isEnabled = false
                    tvSave.setBackgroundColor(resources.getColor(R.color.grey))
                    tvAccuracy.visibility = View.VISIBLE
                    tvAccuracy.text =
                        "Please wait finding accurate location. Current accuracy is
                        ${accuracy.roundToInt()}m"
                } else {
                    progressBar.visibility = View.GONE
                    tvAccuracy.visibility = View.GONE
                    tvSave.isEnabled = true
                    tvSave.setBackgroundColor(resources.getColor(R.color.colorPrimary))
                    if (mGoogleApiClient != null) {
                        if (mGoogleApiClient!!.isConnected) {
                            progressBar.visibility = View.GONE
                            mGoogleApiClient!!.disconnect()
                        }
                    }
                }
            }
    }
}

```



```

        }
    }
} else {
    partnerId = 0
    areaId = 0
    tvErrorMsg.visibility = View.VISIBLE
    tvAvailable.visibility = View.VISIBLE
    tvSave.isEnabled = false
    tvSave.setBackgroundColor(resources.getColor(R.color.grey))
}
}
},
{
    Log.e(TAG, "checkGeoCoordinates: $it")
    if (tvAccuracy != null) {
        progressDialog.dismiss()
        partnerId = 0
        areaId = 0
        tvErrorMsg.visibility = View.VISIBLE
        tvAvailable.visibility = View.VISIBLE
        tvSave.isEnabled = false
        tvSave.setBackgroundColor(resources.getColor(R.color.grey))
    }
})
request.retryPolicy = DefaultRetryPolicy(10000, 0, 1f)
VolleySingleton.getInstance(this).addToRequestQueue(request)
}

private fun checkAvailableLocations() {
    val request = StringRequest(Request.Method.GET,
        "${Urls.GET_AVAILABLE_LOCATIONS}", { response ->
            Log.e(TAG, "checkAvailableLocations: $response")
            if (tvAddress != null) {
                val jsonObject = JSONObject(response.toString())
                availableLocation = if (jsonObject.getBoolean("IsSuccess")) {
                    Gson().fromJson(response.toString(), AvailableLocation::class.java)
                } else {
                    null
                }
            }
        })
}

```

```

    }
}, {
    Log.e(TAG, "checkAvailableLocations: $it")
    if (tvAccuracy != null) {
        availableLocation = null
    }
})
request.retryPolicy = DefaultRetryPolicy(10000, 0, 1f)
VolleySingleton.getInstance(this).addToRequestQueue(request)
}
private fun dialogAvailable() {
    availableDialog = Dialog(this)

availableDialog!!.requestWindowFeature(Window.FEATURE_NO_TITLE)
    availableDialog!!.setContentView(R.layout.dialog_layout)
availableDialog!!.window!!.setBackgroundDrawable(ColorDrawable(Color.TRANSPAREN
T))
    availableDialog!!.window!!.setLayout(
        ConstraintLayout.LayoutParams.MATCH_PARENT,
        ConstraintLayout.LayoutParams.MATCH_PARENT
    )
    ivHeaderBack = availableDialog!!.findViewById(R.id.ivBack) as ImageView
    tvHeader = availableDialog!!.findViewById(R.id.tvHeader) as TextView
    var ivSearch = availableDialog!!.findViewById(R.id.ivSearch) as ImageView
    ivSearch.visibility = View.GONE
    tvHeader!!.text = "Available Locations"
    recyclerViewAvailable = availableDialog!!.findViewById(R.id.recyclerView) as
RecyclerView
    ivHeaderBack!!.setOnClickListener {
        availableDialog!!.dismiss()
    }
    availableArrayList.clear()
    for (i in availableLocation!!.availableLocationData!!.indices) {
        availableArrayList.add(
            ItemsList(
                availableLocationData = availableLocation!!.availableLocationData!![i],
                name = "${availableLocation!!.availableLocationData!![i].areaName},
${availableLocation!!.availableLocationData!![i].state}",
                isSelected = false
            )
        )
    }
}

```

```

        )
    )
}
recyclerViewAvailable!!.layoutManager =
    LinearLayoutManager(this, RecyclerView.VERTICAL, false)
availableAdapter = ItemsAdapter(this, availableArrayList)
recyclerViewAvailable!!.adapter = availableAdapter
availableAdapter!!.onItemClickListener = { it ->
    availableDialog!!.dismiss()
    tvErrorMsg.visibility = View.GONE
    tvAvailable.visibility = View.GONE
    tvAccuracy.visibility = View.GONE
    progressBar.visibility = View.GONE
    usingCurrentLocation = false
    if (mGoogleApiClient!!.isConnected) {
        mGoogleApiClient!!.disconnect()
    }
    areald = it.availableLocationData!!.areald
    tvAddress.text =
        "${it.availableLocationData!!.areaName}, ${it.availableLocationData!!.state},
${it.availableLocationData!!.zipcode}"
    tvErrorMsg.visibility = View.GONE
    latitude = it.availableLocationData!!.latitude!!.toDouble()
    longitude = it.availableLocationData!!.longitude!!.toDouble()
    progressDialog.show()
    mapFragment!!.getMapAsync(this)
}
availableDialog!!.show()
}
private fun updatePinCode() {
    val jsonObject = JSONObject()
    jsonObject.put("UserId", sharedPreferences.getString("USER_ID", null))
    jsonObject.put("PartnerId", partnerId)
    jsonObject.put("Areald", areald)
    if (locality.isNullOrEmpty() || locality == "null") {
        jsonObject.put("Locality", city)
    } else {
        jsonObject.put("Locality", locality)
    }
    jsonObject.put("Zipcode", zipcode)
}

```

```

jsonObject.put("City", city)
jsonObject.put("State", state)
jsonObject.put("Country", country)
jsonObject.put("Latitude", latitude)
jsonObject.put("Longitude", longitude)
jsonObject.put("CurrentLocation", address)
jsonObject.put("IsUseCurrentLocation", usingCurrentLocation)
Log.e(TAG, "updateLocation: $jsonObject")
Utils().updatePagesLog(
    this,
    LogData(
        arealId = "$arealId",
        pageId = Urls.STARTUP_LOCATION_PAGE,
        title = "Update Location",
        eventName = "Save & Proceed",
        note = "$jsonObject"
    )
)
val request = JsonObjectRequest(
    Request.Method.POST,
    rls.UPDATE_SHOPPING,
    jsonObject,
    { response ->
        Log.e(TAG, "updateLocation: $response")
        progressDialog.dismiss()
        val jsonObject = JSONObject(response.toString())
        if (jsonObject.getBoolean("IsSuccess")) {
            PlaceRestaurantOrderFragment.addressData = null
            PlaceOrderFragment.addressData = null
            HomeFragment.apiDataReceived = false
            RestaurantsFragment.apiDataReceived = false
            ShoppingFragment.apiDataReceived = false
            PickAndDropFragment.apiDataReceived = false
            PickAndDropFragment.addressType = null
            if (login != null) {
                try {
                    oldAreaId = login!!.loginData!!.partnerId
                    login!!.loginData!!.zipcode = zipcode
                    login!!.loginData!!.latitude = latitude.toString()
                    login!!.loginData!!.longitude = longitude.toString()
                }
            }
        }
    }
)

```

```

        login!!.loginData!!.currentLocation = address
        login!!.loginData!!.isUseCurrentLocation = usingCurrentLocation
        login!!.loginData!!.areald = areald.toString()
        login!!.loginData!!.partnerId = partnerId
        login!!.loginData!!.locality = locality
        login!!.loginData!!.areald = areald.toString()
        var intent = Intent(applicationContext, MainActivity::class.java)
        intent.flags =
            Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_CLEAR_TOP
        intent.putExtra("loginUser", login)
        startActivity(intent)
        finish()
    } catch (e: Exception) {
        var intent = Intent(applicationContext, MainActivity::class.java)
        intent.flags =
            Intent.FLAG_ACTIVITY_NEW_TASK or
Intent.FLAG_ACTIVITY_CLEAR_TASK or Intent.FLAG_ACTIVITY_CLEAR_TOP
        intent.putExtra("loginUser", login)
        startActivity(intent)
        finish()
    }
} else {
    try {
        oldAreaId = MainActivity.login!!.loginData!!.partnerId
        MainActivity.login!!.loginData!!.zipcode = zipcode
        MainActivity.login!!.loginData!!.latitude = latitude.toString()
        MainActivity.login!!.loginData!!.longitude = longitude.toString()
        MainActivity.login!!.loginData!!.currentLocation = address
        MainActivity.login!!.loginData!!.isUseCurrentLocation =
            usingCurrentLocation
        MainActivity.login!!.loginData!!.areald = areald.toString()
        MainActivity.login!!.loginData!!.partnerId = partnerId
        MainActivity.login!!.loginData!!.locality = locality
        MainActivity.login!!.loginData!!.areald = areald.toString()
        if (oldAreaId != partnerId!!.toInt()) {
            progressDialog.show()
            clearCart()
        } else {

```

```

        finish()
    }
    } catch (e: Exception) {
        if (oldAreaId != partnerId!!.toInt()) {
            progressDialog.show()
            clearCart()
        } else {
            finish()
        }
    }
}
} else{
    tvErrorMsg.visibility = View.VISIBLE
}
}, {
    Log.e(TAG, "updateLocation: $it")
    progressDialog.dismiss()
    Utils().volleyError(it, this)
})
request.retryPolicy = DefaultRetryPolicy(10000, 0, 1f)
VolleySingleton.getInstance(this).addToRequestQueue(request)
}

```

```

private fun clearCart() {
    progressDialog.show()
    val request = StringRequest(
        Request.Method.GET,
        "{$
            Urls.CLEAR_CART + sharedPreferences.getString(
                "USER_ID",
                "0"
            )
        }${Urls.BUSINESS_CATEGORY_ID}0${Urls.BUSINESS_ID}0",
        { response ->
            Log.e(TAG, "getCartItems: $response")
            progressDialog.dismiss()
            var cartArrayList: ArrayList<CartItem> = ArrayList()
            cartArrayList.clear()
            Utils().updateItemInCart(this, cartArrayList)
        }
    )
}

```

```

        var restaurantCartList: ArrayList<RestaurantCartList> = ArrayList()
        restaurantCartList.clear()
        Utils().updateRestaurantInCart(this, restaurantCartList)
        finish()
    },
    {
        Log.e(TAG, "getCartItems: $it")
        progressDialog.dismiss()
    })
    request.retryPolicy = DefaultRetryPolicy(10000, 2, 1f)
    VolleySingleton.getInstance(this).addToRequestQueue(request)
}
}

```

## 5.2. Login\_Activity

```

package com.app.pepens.activity
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.app.pepens.R
import com.app.pepens.fragment.NumberFragment
class LoginActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)
        val fragment = NumberFragment()
        val fragmentManager = supportFragmentManager
        val fragmentTransaction = fragmentManager!!.beginTransaction()
        fragmentTransaction.replace(R.id.frameLayout, fragment)
        fragmentTransaction.addToBackStack(null)
        fragmentTransaction.commit()
    }
    override fun onBackPressed() {
        val fragmentManager = supportFragmentManager
        var fragment = supportFragmentManager.findFragmentById(R.id.frameLayout)
        when (fragment) {
            is NumberFragment -> {
                finish()
            }
            else -> {
                if (fragmentManager.backStackEntryCount > 0) {

```

```

        fragmentManager.popBackStack()
    } else {
        super.onBackPressed()
    }
}
}
}
}
}

```

### 5.3. Main\_Activity

```

package com.app.pemens.activity
import android.app.Activity
import android.content.*
import android.graphics.Color
import android.net.ConnectivityManager
import android.os.Build
import android.os.Bundle
import android.os.Handler
import android.provider.Settings
import android.util.Log
import android.view.View
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.fragment.app.Fragment
import androidx.navigation.fragment.NavHostFragment
import androidx.navigation.ui.NavigationUI
import androidx.navigation.ui.setupWithNavController
import com.app.pemens.BuildConfig
import com.app.pemens.R
import com.app.pemens.fragment.AddUpdateAddressFragment
import com.app.pemens.fragment.HomeFragment
import com.app.pemens.fragment.LocationFragment
import com.app.pemens.fragment.OfferFragment
import com.app.pemens.interfaces.DialogItemClick
import com.app.pemens.interfaces.NotificationUpdate
import com.app.pemens.model.CartItemsList
import com.app.pemens.model.Categories.Categories
import com.app.pemens.model.Login.Login
import com.app.pemens.model.RestaurantCartList

```



```

import com.app.pemens.utils.AnalyticsApplication
import com.app.pemens.utils.CheckConnectivity
import com.app.pemens.utils.Urls
import com.app.pemens.utils.Utils
import com.app.expensemanager.util.VolleySingleton
import com.android.volley.DefaultRetryPolicy
import com.android.volley.Request
import com.android.volley.toolbox.JsonObjectRequest
import com.android.volley.toolbox.StringRequest
import com.google.android.gms.analytics.Tracker
import com.google.android.material.badge.BadgeDrawable
import com.google.android.material.bottomnavigation.BottomNavigationView
import com.google.firebase.iid.FirebaseInstanceId
import com.google.gson.Gson
import org.json.JSONObject
import java.util.*
import kotlin.collections.HashMap

open class MainActivity : AppCompatActivity(), NotificationUpdate, DialogItemClick {
    private var TAG = MainActivity::class.java.simpleName
    private lateinit var sharedPreferences: SharedPreferences
    private var mNetworkReceiver: BroadcastReceiver? = null
    private var badgeDrawable: BadgeDrawable? = null
    private var navHostFragment: NavHostFragment? = null
    companion object {
        var mTracker: Tracker? = null
        var id = 0
        var showDialog = false
        var login: Login? = null
        var bottomNavigationView: BottomNavigationView? = null
        var tvConnection: TextView? = null
        var cartArrayList: ArrayList<CartItem> = ArrayList()
        var restaurantCartList: ArrayList<RestaurantCart> = ArrayList()
        var adFromR = false
        var categories: Categories? = null
    }
    fun dialog(value: Boolean) {
        if (showDialog) {
            tvConnection!!.visibility = View.VISIBLE
            if (value) {
                tvConnection!!.setBackgroundColor(Color.parseColor("#09AF00"))
                tvConnection!!.text = "Back Online."
                Handler().postDelayed({
                    tvConnection!!.visibility = View.GONE

```

```

        }, 2000)
    } else {
        tvConnection!!.setBackgroundColor(Color.RED)
        tvConnection!!.text = "No connection."
    }
}
}
}
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
    Utils().restartAppAtCrash(this, MainActivity::class.java.simpleName)
    mNetworkReceiver = CheckConnectivity()
    registerReceiver(mNetworkReceiver,
IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION))
    sharedPreferences = getSharedPreferences("USER_DATA", Context.MODE_PRIVATE)
    if (intent != null) {
        login = intent.getParcelableExtra("loginUser")
    }
    val application: AnalyticsApplication = application as AnalyticsApplication
    mTracker = application.defaultTracker
    if (SplashActivity.checkLoginDetails) {
        getLoginUser()
    }
    bottomNavigationView = findViewById(R.id.bottomNavigationView)
    navHostFragment =
        supportFragmentManager.findFragmentById(R.id.fragmentContainerView) as
NavHostFragment
    bottomNavigationView!!.setupWithNavController(navHostFragment!!.navController)
    tvConnection = findViewById(R.id.tv_connection)
    bottomNavigationView!!.setOnNavigationItemSelectedListener {
        when (it.itemId) {
            R.id.homeFragment -> {
                HomeFragment.apiDataReceived = false
                navHostFragment!!.navController.navigate(R.id.homeFragment)
            }
            R.id.offerFragment -> {
                OfferFragment.apiDataReceived = false
                navHostFragment!!.navController.navigate(R.id.offerFragment)
            }
            R.id.mainDialogFragment -> {
                navHostFragment!!.navController.navigate(R.id.mainDialogFragment)
            }
            R.id.feedFragment -> {

```

```

        navHostFragment!!.navController.navigate(R.id.feedFragment)
    }
    R.id.notificationFragment -> {
        navHostFragment!!.navController.navigate(R.id.notificationFragment)
    }
}
true
}
badgeDrawable = bottomNavigationView!!.getOrCreateBadge(R.id.notificationFragment)
badgeDrawable!!.backgroundColor = Color.RED
badgeDrawable!!.badgeTextColor = Color.WHITE
badgeDrawable!!.maxCharacterCount = 2
getMainCategories()
Handler().postDelayed({
    showDialog = true
    if (Utils().verifyAvailableNetwork(this)) {
        updateDevice()
    }
}, 3000)
}
override fun updateNotification(visible: Boolean, count: Int) {
    if (visible) {
        badgeDrawable!!.number = count
        badgeDrawable!!.isVisible = true
    } else {
        badgeDrawable!!.number = count
        badgeDrawable!!.isVisible = false
    }
}
}
override fun itemDialogClick(id: Int) {
    when (id) {
        Urls.ID_RESTAURANT_CATEGORY -> {
            navHostFragment!!.navController.navigate(R.id.restaurantsFragment)
        }
        Urls.ID_GROCERY_CATEGORY -> {
            var bundle = Bundle()
            bundle.putInt("BusinessCategoryId", id)
            navHostFragment!!.navController.navigate(R.id.shoppingFragment, bundle)
        }
        Urls.ID_PICK_DROP_CATEGORY -> {
            navHostFragment!!.navController.navigate(R.id.pickAndDropFragment)
        }
    }
}

```

```

    }
    else -> {
        var bundle = Bundle()
        bundle.putInt("BusinessCategoryId", id)
        navHostFragment!!.navController.navigate(R.id.shoppingFragment, bundle)
    }
}
}
private fun getLoginUser() {
    val request = StringRequest(Request.Method.GET,
        Urls.GET_USER_ON_STARTUP + sharedPreferences.getString("USER_ID", "0") +
        Urls.TOKEN,
        { response ->
            Log.e(TAG, "getLoginUser: $response")
            val jsonObject = JSONObject(response.toString())
            if (jsonObject.getBoolean("IsSuccess")) {
                login = Gson().fromJson(response.toString(), Login::class.java)
            }
        }, {
            Log.e(TAG, "getLoginUser: $it")
        })
    request.retryPolicy = DefaultRetryPolicy(10000, 0, 1f)
    VolleySingleton.getInstance(this).addToRequestQueue(request)
}
private fun updateDevice() {
    val params = HashMap<String, String>()
    params["AppDeviceId"] = "0"
    params["Told"] = sharedPreferences.getString("USER_ID", "0").toString()
    params["ToType"] = "4"
    params["DeviceId"] = Settings.Secure.getString(this.contentResolver,
        Settings.Secure.ANDROID_ID)
    params["AppVersionNumber"] = BuildConfig.VERSION_CODE.toString()
    params["AppVersionName"] = BuildConfig.VERSION_NAME
    params["DeviceOS"] = "Android"
    params["DeviceOSVersion"] = Build.VERSION.SDK
    params["DeviceModel"] = Build.MODEL
    params["DeviceManufacturer"] = Build.MANUFACTURER
    params["DeviceName"] = "${Build.MANUFACTURER} - ${Build.MODEL}"
    params["IPAddress"] = Utils().getMobileIPAddress().toString()
    params["FirebaseToken"] = FirebaseInstanceId.getInstance().token.toString()
    val jsonObject = JSONObject(params as Map<*, *>)
    Log.e(TAG, "updateDevice: $jsonObject")
    val request = JsonObjectRequest(

```

```

        Request.Method.POST,
        Urls.AddUpdateDevice,
        jsonObject,
        { response ->
            Log.e("TAG", "updateDevice: \$response")
        },
        {
            Log.e("TAG", "updateDevice: \$it")
        })
        request.retryPolicy = DefaultRetryPolicy(10000, 0, 1f)
        VolleySingleton.getInstance(this).addToRequestQueue(request)
    }

    override fun onSupportNavigateUp(): Boolean {
        return NavigationUI.navigateUp(navHostFragment!!.navController, null)
    }

    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
        super.onActivityResult(requestCode, resultCode, data)
        when (requestCode) {
            102 -> {
                when (resultCode) {
                    Activity.RESULT_OK -> {
                        val navHostFragment: Fragment? =
                            supportFragmentManager.primaryNavigationFragment
                        val fragment: Fragment = navHostFragment!!.childFragmentManager.fragments[0]
                        if (fragment is AddUpdateAddressFragment) {
                            (fragment as AddUpdateAddressFragment).gpsEvent(true)
                        } else if (fragment is LocationFragment) {
                            (fragment as LocationFragment).gpsEvent(true)
                        }
                    }
                }
                // val fragment: AddUpdateAddressFragment =
                supportFragmentManager.findFragmentById(R.id.frameLayout) as AddUpdateAddressFragment
                // fragment.gpsEvent(true)
            }
            else -> {
                val navHostFragment: Fragment? =
                    supportFragmentManager.primaryNavigationFragment
                val fragment: Fragment = navHostFragment!!.childFragmentManager.fragments[0]
                if (fragment is AddUpdateAddressFragment) {
                    (fragment as AddUpdateAddressFragment).gpsEvent(false)
                } else if (fragment is LocationFragment) {
                    (fragment as LocationFragment).gpsEvent(false)
                }
            }
        }
    }

```

```

//          val fragment: AddUpdateAddressFragment =
supportFragmentManager.findFragmentById(R.id.frameLayout) as AddUpdateAddressFragment
//          fragment.gpsEvent(false)
        }
    }
}
}
override fun onDestroy() {
    super.onDestroy()
    unregisterNetworkChanges()
}
private fun unregisterNetworkChanges() {
    try {
        unregisterReceiver(mNetworkReceiver)
    } catch (e: IllegalArgumentException) {
        e.printStackTrace()
    }
}
fun getMainCategories() {
    val request = StringRequest(
        Request.Method.GET,
        Urls.GET_HOME_CATEGORIES + login!!.loginData!!.areaId + Urls.USER_ID +
sharedPreferences.getString(
    "USER_ID",
    "0"
),
    { response ->
        Log.e(TAG, "getMainCategories: $response")
        if (bottomNavigationView != null) {
            val jsonObject = JSONObject(response.toString())
            categories = if (jsonObject.has("IsSuccess") && jsonObject.getBoolean("IsSuccess")) {
                Gson().fromJson(response.toString(), Categories::class.java)
            } else {
                null
            }
            categories!!.categoriesData!!.removeAll { item -> item.businessCategoryId !=
Urls.ID_GROCERY_CATEGORY }
        }
    },
    {
        Log.e(TAG, "getMainCategories: $it")
    })
}

```

```
        request.retryPolicy = DefaultRetryPolicy(10000, 2, 1f)
        VolleySingleton.getInstance(this).addToRequestQueue(request)
    }
}
```

## **6. TEST MODULE CLASS**

### **6.1. Addresses\_Adaptor**

```
package com.app.pemens.adapter
import android.content.Context
import android.graphics.Color
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
```

```

import androidx.recyclerview.widget.RecyclerView
import com.app.pemens.R
import com.app.pemens.activity.MainActivity
import com.app.pemens.fragment.AddressListFragment
import com.app.pemens.model.Address.Addresses
import com.app.pemens.model.Address.AddressData
import com.google.android.material.button.MaterialButton
import kotlinx.android.synthetic.main.custom_addresses_layout.view.*
class AddressesAdapter : RecyclerView.Adapter<AddressesAdapter.ViewHolder> {
    private val context: Context
    private val addresses: Addresses
    private val clickEvent: ClickEvent
    constructor(
        context: Context,
        addresses: Addresses,
        addressListFragment: AddressListFragment
    ) : super() {
        this.context = context
        this.addresses = addresses
        this.clickEvent = addressListFragment
    }
    interface ClickEvent {
        fun editAddress(position: Int)
        fun deleteAddress(position: Int)
        fun deliverToAddress(position: Int)
    }
    override fun onCreateViewHolder(viewGroup: ViewGroup, position: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context).inflate(
                R.layout.custom_addresses_layout,
                viewGroup,
                false
            )
        )
    }
    override fun getItemCount(): Int {
        return if (addresses.addressData == null) {
            0
        } else {
            addresses.addressData.size
        }
    }
    override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {

```



```

viewHolder.tvType.text = addresses.addressData[position].addressType
var add = ""
if (!addresses.addressData[position].firstName.isNullOrEmpty())
    add += "${addresses.addressData[position].firstName}"
if (!addresses.addressData[position].lastName.isNullOrEmpty())
    add += " ${addresses.addressData[position].lastName}"
if (!addresses.addressData[position].addressLine1.isNullOrEmpty())
    add += "\n${addresses.addressData[position].addressLine1}, "
if (!addresses.addressData[position].addressLine2.isNullOrEmpty())
    add += "${addresses.addressData[position].addressLine2}"
// if (!addresses.addressData[position].landmark.isNullOrEmpty())
//     add += "\nLandmark: ${addresses.addressData[position].landmark}"
if (!addresses.addressData[position].city.isNullOrEmpty())
    add += "\n${addresses.addressData[position].city}"
if (!addresses.addressData[position].zipcode.isNullOrEmpty())
    add += " (${addresses.addressData[position].zipcode})"
if (!addresses.addressData[position].state.isNullOrEmpty())
    add += ", ${addresses.addressData[position].state}"
if (!addresses.addressData[position].country.isNullOrEmpty())
    add += ", ${addresses.addressData[position].country}"
viewHolder.tvAddress.text = add
if (addresses.addressData[position].isDefault) {
    viewHolder.tvDefault.visibility = View.VISIBLE
} else {
    viewHolder.tvDefault.visibility = View.GONE
}
if (AddressListFragment.goBackForAddress) {
    if (addresses.addressData[position].areald !=
MainActivity.login!!.loginData!!.areald.toInt()) {
        viewHolder.tvType.setTextColor(Color.parseColor("#888888"))
        viewHolder.tvAddress.setTextColor(Color.parseColor("#888888"))
        viewHolder.tvDefault.setTextColor(Color.parseColor("#888888"))
    }
}
viewHolder.btnEdit.setOnClickListener {
    clickEvent.editAddress(position)
}
viewHolder.btnDelete.setOnClickListener {
    clickEvent.deleteAddress(position)
}
viewHolder.itemView.setOnClickListener {
    clickEvent.deliverToAddress(position)
}

```

```

    }
}
fun removeItem(position: Int) {
    addresses.addressData.removeAt(position)
    notifyItemRemoved(position)
}
fun restoreItem(item: AddressData, position: Int) {
    addresses.addressData.add(position, item)
    notifyItemInserted(position)
}
fun getData(): ArrayList<AddressData> {
    return addresses.addressData
}
class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val tvDefault: TextView = view.tvDefault
    val tvType: TextView = view.tvType
    val tvAddress: TextView = view.tvAddress
    val btnEdit: MaterialButton = view.btnEdit
    val btnDelete: MaterialButton = view.btnDelete
}
}

```

## 6.2. Ads\_Pager\_Adapter

```

package com.app.pepens.adapter
import android.content.Context
import android.content.Intent
import android.net.Uri
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import androidx.navigation.Navigation
import androidx.navigation.findNavController
import androidx.recyclerview.widget.RecyclerView
import com.app.pepens.R
import com.app.pepens.activity.MainActivity
import com.app.pepens.model.Ads.Ads
import com.app.pepens.model.LogData
import com.app.pepens.utils.Url
import com.app.pepens.utils.Utils

```

```

import com.bumptech.glide.Glide
import kotlinx.android.synthetic.main.custom_ads_layout.view.*
/**
 * Created by Vikas Kumar Singh on 10/07/20.
 */
class AdsPagerAdapter : RecyclerView.Adapter<AdsPagerAdapter.ViewHolder> {
    private var context: Context
    private var ads: Ads
    constructor(context: Context, ads: Ads) : super() {
        this.context = context
        this.ads = ads
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context).inflate(R.layout.custom_ads_layout, parent, false)
        )
    }
    override fun getItemCount(): Int {
        return if (ads!!.adsData == null) {
            0
        } else {
            ads!!.adsData!!.count()
        }
    }
    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        var sharedPreferences = context!!.getSharedPreferences("USER_DATA",
Context.MODE_PRIVATE)
        Glide.with(context!!)
            .load(ads!!.adsData!![position].coverPictureUrl)
            .into(holder.ivImage)
        holder.itemView.setOnClickListener {
            if (ads!!.adsData!![position].linkTypeId == 1) {

                when (Navigation.findNavController(it).currentDestination?.id) {
                    R.id.homeFragment -> {
                        Utils().updatePagesLog(
                            context,
                            LogData(
                                arealId = MainActivity.login!!.loginData!!.arealId,
                                pageId = Urls.HOME_PAGE,
                                title = "${ads!!.adsData!![position].title}",
                                eventName = "Ad Click",
                                pageEntityId = "${ads!!.adsData!![position].entityId}"

```

```

        )
    )
}
R.id.shoppingFragment -> {
    Utils().updatePagesLog(
        context,
        LogData(
            arealId = MainActivity.login!!.loginData!!.arealId,
            pageId = Urls.SHOPPING_PAGE,
            title = "${ads!!.adsData!![position].title}",
            eventName = "Ad Click",
            pageEntityId = "${ads!!.adsData!![position].entityId}"
        )
    )
}
R.id.restaurantsFragment -> {
    Utils().updatePagesLog(
        context,
        LogData(
            arealId = MainActivity.login!!.loginData!!.arealId,
            pageId = Urls.RESTAURANT_PAGE,
            title = "${ads!!.adsData!![position].title}",
            eventName = "Ad Click",
            pageEntityId = "${ads!!.adsData!![position].entityId}"
        )
    )
}
}
when (ads!!.adsData!![position].entityType) {
    1 -> {
        Utils().clickStream(
            context,
            sharedPreferences.getString("USER_ID", "0")!!,
            "16",
            "${ads!!.adsData!![position].adId}",
            "2",
            "${ads!!.adsData!![position].adId}",
            ""
        )
        var bundle = Bundle()
        bundle.putInt("businessId", ads!!.adsData!![position].entityId)
        bundle.putString("businessName", "")
        when (Navigation.findNavController(it).currentDestination?.id) {

```

```

R.id.homeFragment -> {
    it.findNavController().navigate(
        R.id.action_homeFragment_to_categoryBusinessDetailsFragment,
        bundle
    )
}
R.id.shoppingFragment -> {
    it.findNavController().navigate(
        R.id.action_shoppingFragment_to_categoryBusinessDetailsFragment,
        bundle
    )
}
R.id.restaurantsFragment -> {
    it.findNavController().navigate(
        R.id.action_restaurantsFragment_to_categoryBusinessDetailsFragment,
        bundle
    )
}
}
2 -> {
    Utils().clickStream(
        context,
        sharedPreferences.getString("USER_ID", "0")!!,
        "7",
        "${ads!!.adsData!![position].adId}",
        "2",
        "${ads!!.adsData!![position].adId}",
        ""
    )
    var bundle = Bundle()
    bundle.putInt("offerId", ads!!.adsData!![position].entityId)
    when (Navigation.findNavController(it).currentDestination?.id) {
        R.id.homeFragment -> {
            it.findNavController()
                .navigate(
                    R.id.action_homeFragment_to_offerDetailsFragment,
                    bundle
                )
        }
        R.id.shoppingFragment -> {
            it.findNavController()
                .navigate(

```

```

        R.id.action_shoppingFragment_to_offerDetailsFragment,
        bundle
    )
}
R.id.restaurantsFragment -> {
    it.findNavController()
        .navigate(
            R.id.action_restaurantsFragment_to_offerDetailsFragment,
            bundle
        )
}
}
}
3 -> {
    Utils().clickStream(
        context,
        sharedPreferences.getString("USER_ID", "0")!!,
        "6",
        "${ads!!.adsData!![position].adId}",
        "2",
        "${ads!!.adsData!![position].adId}",
        ""
    )
    var bundle = Bundle()
    bundle.putInt("id", ads!!.adsData!![position].entityId)
    when (Navigation.findNavController(it).currentDestination?.id) {
        R.id.homeFragment -> {
            it.findNavController()
                .navigate(
                    R.id.action_homeFragment_to_feedDetailsFragment,
                    bundle
                )
        }
        R.id.shoppingFragment -> {
            it.findNavController()
                .navigate(
                    R.id.action_shoppingFragment_to_feedDetailsFragment,
                    bundle
                )
        }
        R.id.restaurantsFragment -> {
            it.findNavController()
                .navigate(

```

```

        R.id.action_restaurantsFragment_to_feedDetailsFragment,
        bundle
    )
    }
}
}
4 -> {
    when (ads!!.adsData!![position].entityId) {
        1 -> {
        }
        2 -> {
            when (Navigation.findNavController(it).currentDestination?.id) {
                R.id.homeFragment -> {
                    it.findNavController()
                        .navigate(R.id.action_homeFragment_to_referralFragment)
                }
                R.id.shoppingFragment -> {
                    it.findNavController()
                        .navigate(R.id.action_shoppingFragment_to_referralFragment)
                }
                R.id.restaurantsFragment -> {
                    it.findNavController()
                        .navigate(R.id.action_restaurantsFragment_to_referralFragment)
                }
            }
        }
    }
    3 -> {
        when (Navigation.findNavController(it).currentDestination?.id) {
            R.id.homeFragment -> {
                it.findNavController()
                    .navigate(R.id.action_homeFragment_to_rewardsFragment)
            }
            R.id.shoppingFragment -> {
                it.findNavController()
                    .navigate(R.id.action_shoppingFragment_to_rewardsFragment)
            }
            R.id.restaurantsFragment -> {
                it.findNavController()
                    .navigate(R.id.action_restaurantsFragment_to_rewardsFragment)
            }
        }
    }
}
4 -> {

```

```

when (Navigation.findNavController(it).currentDestination?.id) {
    R.id.homeFragment -> {
        it.findNavController()
            .navigate(R.id.action_homeFragment_to_shoppingFragment)
    }
    R.id.shoppingFragment -> {

    }
    R.id.restaurantsFragment -> {
        it.findNavController()
            .navigate(R.id.action_restaurantsFragment_to_shoppingFragment)
    }
}
}
}
5 -> {
    if (ads!!.adsData!![position].entityId == 0) {
        if (!MainActivity.adFromR) {
            when (Navigation.findNavController(it).currentDestination?.id) {
                R.id.homeFragment -> {
                    it.findNavController()
                        .navigate(R.id.action_homeFragment_to_restaurantsFragment)
                }
                R.id.shoppingFragment -> {
                    it.findNavController()
                        .navigate(R.id.action_shoppingFragment_to_restaurantsFragment)
                }
                R.id.restaurantsFragment -> {

                }
            }
        }
    } else {
        var bundle = Bundle()
        bundle.putInt("businessId", ads!!.adsData!![position].entityId)
        bundle.putString("businessName", "")
        bundle.putInt("itemId", 0)
        when (Navigation.findNavController(it).currentDestination?.id) {
            R.id.homeFragment -> {
                it.findNavController().navigate(
                    R.id.action_homeFragment_to_restaurantDetailsFragment,
                    bundle
                )
            }
        }
    }
}

```



```

        )
    }
    R.id.shoppingFragment -> {
        it.findNavController().navigate(
            R.id.action_shoppingFragment_to_restaurantDetailsFragment,
            bundle
        )
    }
    R.id.restaurantsFragment -> {
        it.findNavController().navigate(
            R.id.action_restaurantsFragment_to_restaurantDetailsFragment,
            bundle
        )
    }
}
}
}
}
} else if (ads!!.adsData!![position].linkTypeId == 2) {
    val defaultBrowser = Intent.makeMainSelectorActivity(Intent.ACTION_MAIN,
Intent.CATEGORY_APP_BROWSER)
    defaultBrowser.data = Uri.parse(ads!!.adsData!![position].linkUrl)
    context!!.startActivity(defaultBrowser)
}
}
}

class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val ivImage: ImageView = view.ivImage
}
}

```

### 6.3. All\_Categories\_Adapter

```

package com.app.popens.adapter
import android.content.Context
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView

```

```

import androidx.navigation.Navigation
import androidx.navigation.findNavController
import androidx.recyclerview.widget.GridLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.app.pepens.R
import com.app.pepens.fragment.ShoppingFragment
import com.app.pepens.model.ItemCategories.ItemCategories
import com.app.pepens.utils.Utils
import kotlinx.android.synthetic.main.custom_all_categories_layout.view.*
class AllCategoriesAdapter : RecyclerView.Adapter<AllCategoriesAdapter.ViewHolder> {
    private var context: Context
    private var itemCategories: ItemCategories
    constructor(context: Context, itemCategories: ItemCategories) {
        this.context = context
        this.itemCategories = itemCategories
    }
    override fun onCreateViewHolder(viewGroup: ViewGroup, position: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context)
                .inflate(R.layout.custom_all_categories_layout, viewGroup, false)
        )
    }
    override fun getItemCount(): Int {
        return if (itemCategories!!.itemCategoriesData == null) {
            0
        } else {
            itemCategories!!.itemCategoriesData!!.size
        }
    }
    override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {
        var sharedPreferences = context.getSharedPreferences("USER_DATA",
Context.MODE_PRIVATE)
        viewHolder.tvTitle.text = "Shop by

${itemCategories!!.itemCategoriesData!![position].itemCategoryName}"

        if (itemCategories!!.itemCategoriesData!![position].maxDiscountPercentage <= 0) {
            viewHolder.tvDiscount.visibility = View.GONE
        } else {
            viewHolder.tvDiscount.visibility = View.VISIBLE
            viewHolder.tvDiscount.text = "Upto
${itemCategories!!.itemCategoriesData!![position].maxDiscountPercentage}%    OFF"
        }
    }
}

```

```

viewHolder.recyclerView.layoutManager = GridLayoutManager(context, 3)
var categoriesAdapter = AllSubCategoriesAdapter(
    context,
    itemCategories.itemCategoriesData!![position].itemCategories!!
)
viewHolder.recyclerView.adapter = categoriesAdapter

viewHolder.tvViewAll.setOnClickListener {
    Utils().clickStream(
        context,
        sharedPreferences.getString("USER_ID", "0")!!,
        "25",
        "${itemCategories!!.itemCategoriesData!![position].itemCategoryId}",
        "23",
        "${itemCategories!!.itemCategoriesData!![position].itemCategoryId}",
        ""
    )
    var bundle = Bundle()
    bundle.putInt("BusinessCategoryId", ShoppingFragment.shoppingBusinessCategoryId)
    bundle.putInt("CategoryId",
itemCategories!!.itemCategoriesData!![position].itemCategoryId)
    bundle.putInt("SubCategoryId", 0)
    if (Navigation.findNavController(it).currentDestination?.id == R.id.shoppingFragment)

it.findNavController().navigate(R.id.action_shoppingFragment_to_shoppingProductsFragment,
bundle)
    }
}
class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val tvTitle: TextView = view.tvTitle
    val tvViewAll: TextView = view.tvViewAll
    val tvDiscount: TextView = view.tvDiscount
    val recyclerView: RecyclerView = view.recyclerView
}
}

```

#### 6.4. All\_Product\_Adapter

```

package com.app.pepens.adapter
import android.content.Context
import android.view.LayoutInflater
import android.view.View

```

```

import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.RecyclerView
import com.app.pepens.R
import com.app.pepens.fragment.AllProductsFragment
import com.app.pepens.model.ProductsWithCategory.BusinessItem
import kotlinx.android.synthetic.main.custom_business_items_layout.view.*
class AllProductsAdapter : RecyclerView.Adapter<AllProductsAdapter.ViewHolder> {
    private var context: Context
    private var businessItems: ArrayList<BusinessItem>
    private var clickEvent: ClickEvent
    constructor(
        context: Context,
        businessItems: ArrayList<BusinessItem>,
        allProductsFragment: AllProductsFragment
    ) {
        this.context = context
        this.businessItems = businessItems
        this.clickEvent = allProductsFragment
    }
    interface ClickEvent {
        fun viewBrands(itemId: BusinessItem)
    }
    override fun onCreateViewHolder(viewGroup: ViewGroup,
position: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context)
                .inflate(R.layout.custom_business_items_layout, viewGroup, false)
        )
    }
    override fun getItemCount(): Int {
        return businessItems.count()
    }
    override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {
        viewHolder.tvTitle.text = businessItems!![position].item!!.itemName
        if (businessItems!![position].inStock) {
            viewHolder.ivCheck.setColorFilter(
                ContextCompat.getColor(context, R.color.green),
                android.graphics.PorterDuff.Mode.SRC_IN
            )
        }
    }
}

```

```

        viewHolder.tvTitle.setTextColor(context.resources.getColor(R.color.colorText))
        viewHolder.tvBrands.setTextColor(context.resources.getColor(R.color.green))
        viewHolder.tvBrands.setBackgroundResource(R.drawable.view_brand_active)
    } else {
        viewHolder.ivCheck.setColorFilter(
            ContextCompat.getColor(context, R.color.grey),
            android.graphics.PorterDuff.Mode.SRC_IN
        )
        viewHolder.tvTitle.setTextColor(context.resources.getColor(R.color.grey))
        viewHolder.tvBrands.setTextColor(context.resources.getColor(R.color.grey))
        viewHolder.tvBrands.setBackgroundResource(R.drawable.view_brand_inactive)
    }
    if (businessItems!![position].noOfBrands > 0) {
        viewHolder.tvBrands.visibility = View.VISIBLE
    } else {
        viewHolder.tvBrands.visibility = View.GONE
    }
    viewHolder.tvBrands.setOnClickListener {
//        if (businessItems[position].inStock){
            clickEvent.viewBrands(businessItems[position])
//        }
    }
}

class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val ivCheck: ImageView = view.ivCheck
    val tvTitle: TextView = view.tvTitle
    val tvBrands: TextView = view.tvBrands
}
}

```

## 6.5. Brand\_Adapter

```

package com.app.pepens.adapter
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.core.content.ContextCompat
import androidx.recyclerview.widget.RecyclerView

```

```

import com.app.pemens.R
import com.app.pemens.model.Brands.Brands
import kotlinx.android.synthetic.main.custom_business_items_layout.view.*
class BrandsAdapter : RecyclerView.Adapter<BrandsAdapter.ViewHolder> {
    private var context: Context
    private var brands: Brands
    constructor(context: Context, brands: Brands) {
        this.context = context
        this.brands = brands
    }
    override fun onCreateViewHolder(viewGroup: ViewGroup, position: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context)
                .inflate(R.layout.custom_business_items_layout, viewGroup, false)
        )
    }
    override fun getItemCount(): Int {
        return if (brands!!.brandsData == null) {
            0
        } else {
            brands!!.brandsData!!.count()
        }
    }
    override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {
        viewHolder.tvTitle.text = brands!!.brandsData!![position].brand!!.brandName
        viewHolder.ivCheck.visibility = View.GONE
        viewHolder.tvBrands.visibility = View.GONE
        viewHolder.ivCheck1.visibility = View.VISIBLE
        if (brands!!.brandsData!![position].inStock) {
            viewHolder.ivCheck1.setColorFilter(ContextCompat.getColor(context, R.color.green),
            android.graphics.PorterDuff.Mode.SRC_IN)
            viewHolder.tvTitle.setTextColor(context.resources.getColor(R.color.black))
        } else {
            viewHolder.ivCheck1.setColorFilter(ContextCompat.getColor(context, R.color.grey),
            android.graphics.PorterDuff.Mode.SRC_IN)
            viewHolder.tvTitle.setTextColor(context.resources.getColor(R.color.grey))
        }
    }
    class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val ivCheck: ImageView = view.ivCheck
        val ivCheck1: ImageView = view.ivCheck1
        val tvTitle: TextView = view.tvTitle
        val tvBrands: TextView = view.tvBrands
    }
}

```

## 6.6. Business\_Adapter

```
package com.app.pepens.adapter
import android.content.Context
import android.content.res.ColorStateList
import android.graphics.Color
import android.os.Build
import android.os.Bundle
import android.util.DisplayMetrics
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.annotation.RequiresApi
import androidx.constraintlayout.widget.ConstraintLayout
import androidx.core.content.ContextCompat
import androidx.fragment.app.FragmentActivity
import androidx.navigation.findNavController
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.app.pepens.R
import com.app.pepens.fragment.CategoryBusinessFragment
import com.app.pepens.model.Business.Business
import com.app.pepens.utils.Utils
import com.bumptech.glide.Glide
import com.google.android.material.card.MaterialCardView
import kotlinx.android.synthetic.main.custom_business_layout.view.*
import kotlinx.android.synthetic.main.fragment_category_business_details.*
import java.math.RoundingMode
/**
 * Created by Vikas Kumar Singh on 11/07/20.
 */
class BusinessAdapter : RecyclerView.Adapter<BusinessAdapter.ViewHolder> {
    private var context: Context
    private var business: Business
    private var eventListeners: EventListeners
    interface EventListeners {
        fun showHours(position: Int)
        fun makeCall(position: Int)
        fun openLocation(position: Int)
        fun openOffer(position: Int)
        fun makeFavorite(position: Int)
        fun openGallery(position: Int)
```

```

        fun viewImage(position: Int, i: Int)
        fun closeDialog()
    }
    constructor(
        context: Context,
        business: Business,
        categoryBusinessFragment: CategoryBusinessFragment
    ) : super() {
        this.context = context
        this.business = business
        this.eventsListeners = categoryBusinessFragment
    }
    override fun onCreateView(parent: ViewGroup, viewType: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context).inflate(R.layout.custom_business_layout, parent, false)
        )
    }
    override fun getItemCount(): Int {
        return if (business == null) {
            0
        } else {
            business.businessData!!.count()
        }
    }
}
@RequiresApi(Build.VERSION_CODES.LOLLIPOP)
override fun onBindViewHolder(holder: ViewHolder, position: Int) {
    var sharedPreferences = context!!.getSharedPreferences("USER_DATA",
Context.MODE_PRIVATE)
    var displayMetrics = DisplayMetrics()
    (context as FragmentActivity).windowManager.defaultDisplay.getMetrics(displayMetrics)
    var width = displayMetrics.widthPixels
    var height = (width / 2.33).toInt()
    holder.ivBanner.layoutParams.height = height
    Glide.with(context!!)
        .load(business!!.businessData!![position].coverPictureUri)
        .placeholder(R.drawable.business_banner)
        .into(holder.ivBanner)
    Glide.with(context!!)
        .load(business!!.businessData!![position].defaultPictureUri)
        .placeholder(R.mipmap.ic_launcher)
        .into(holder.ivProfilePic)
    holder.tvTitle.text =

```



```

business!!.businessData!![position].businessName
    if (business!!.businessData!![position].distance.isNullOrEmpty()) {
        holder.tvDistance.text = "0 KM"
    } else {
        holder.tvDistance.text = "${
            business!!.businessData!![position].distance!!.toBigDecimal()
                .setScale(2, RoundingMode.HALF_UP).toDouble()
        } KM"
    }
// holder.tvProducts.text = business!!.businessData!![position].noOfItems.toString()
holder.tvCategory.text = business!!.businessData!![position].businessCategoryName
if (business!!.businessData!![position].isFavorite) {
    holder.clFav.setBackgroundColor(context.resources.getColor(R.color.pink))
    holder.ivFavorite.setImageResource(R.drawable.ic_favorite)
} else {
    holder.clFav.setBackgroundColor(context.resources.getColor(R.color.black))
    holder.ivFavorite.setImageResource(R.drawable.ic_favorite_border)
}
if (business!!.businessData!![position].noOfActiveOffers > 0) {
    holder.tvOfferCount.visibility = View.VISIBLE
    holder.tvOfferCount.text =
        business!!.businessData!![position].noOfActiveOffers.toString()
} else {
    holder.tvOfferCount.visibility = View.GONE
}
if (business!!.businessData!![position].itemsAvailabilityPercentage > 0) {
    holder.tvProductCount.visibility = View.VISIBLE
    holder.tvProductCount.text =
        "${business!!.businessData!![position].itemsAvailabilityPercentage}%"
} else {
    holder.tvProductCount.visibility = View.GONE
}
if (business!!.businessData!![position].isAvailableHomeDelivery) {
    holder.tvDelivery.visibility = View.VISIBLE
} else {
    holder.tvDelivery.visibility = View.GONE
}
if (business!!.businessData!![position].noOfMatchedItems > 0) {
    if (business!!.businessData!![position].noOfMatchedItems == 1) {
        holder.tvMatch.text =
            "${business!!.businessData!![position].noOfMatchedItems} matched"
    } else {
        holder.tvMatch.text =
            "${business!!.businessData!![position].noOfMatchedItems} matched"
    }
}

```

```

    }
    holder.tvMatch.visibility = View.VISIBLE
} else {
    holder.tvMatch.visibility = View.GONE
}
if (business.businessData!![position].rating > 0) {
    holder.tvRatingValue.text =
"${business.businessData!![position].rating.toString().replace(".", "")} rating"
} else {
    holder.tvRatingValue.text = "No rating"
}
when {
    business.businessData!![position].rating <= 1 -> {
        holder.cv6.background.setTint(Color.parseColor("#FF0000"))
        holder.ivRatingStar.imageTintList =
            ColorStateList.valueOf(Color.parseColor("#FF0000"))
    }
    business.businessData!![position].rating <= 3 -> {
        holder.cv6.background.setTint(Color.parseColor("#FF8C00"))
        holder.ivRatingStar.imageTintList =
            ColorStateList.valueOf(Color.parseColor("#FF8C00"))
    }
    else -> {
        holder.cv6.background.setTint(Color.parseColor("#09AF00"))
        holder.ivRatingStar.imageTintList =
            ColorStateList.valueOf(Color.parseColor("#09AF00"))
    }
}

if (business!!.businessData!![position].items != null) {
    if (business!!.businessData!![position].items!!.size >
0) {
        holder.ivAutoDetectArrow.visibility = View.VISIBLE
        holder.recyclerViewProductsItems.visibility = View.VISIBLE
        val layoutManager = LinearLayoutManager(context, RecyclerView.HORIZONTAL, false)
        holder.recyclerViewProductsItems.layoutManager = layoutManager
        var businessAdapter =
            ProductItemsAdapter(context, business.businessData!![position].items!!)
        holder.recyclerViewProductsItems.adapter = businessAdapter
    } else {
        holder.recyclerViewProductsItems.visibility = View.GONE
        holder.ivAutoDetectArrow.visibility = View.GONE
    }
} else {

```

```

        holder.recyclerViewProductsItems.visibility = View.GONE
        holder.ivAutoDetectArrow.visibility = View.GONE
    }
    when (business.businessData!![position].onOffStatus) {
        0 -> {
            holder.tvStatus.text = "CLOSED"
            holder.tvStatus.setTextColor(context.resources.getColor(R.color.red))
            holder.tvStatus.setBackgroundResource(R.drawable.view_shop_close)
            holder.ivArrow.setColorFilter(
                ContextCompat.getColor(context, R.color.red),
                android.graphics.PorterDuff.Mode.SRC_IN
            )
        }
        1 -> {
            holder.tvStatus.text = "OPEN"
            holder.tvStatus.setTextColor(context.resources.getColor(R.color.green))
            holder.tvStatus.setBackgroundResource(R.drawable.view_shop_open)
            holder.ivArrow.setColorFilter(
                ContextCompat.getColor(context, R.color.green),
                android.graphics.PorterDuff.Mode.SRC_IN
            )
        }
        2 -> {
            holder.tvStatus.text = "BUSY"
            holder.tvStatus.setTextColor(context.resources.getColor(R.color.grey))
            holder.tvStatus.setBackgroundResource(R.drawable.view_shop_busy)
            holder.ivArrow.setColorFilter(
                ContextCompat.getColor(context, R.color.grey),
                android.graphics.PorterDuff.Mode.SRC_IN
            )
        }
    }
    holder.tvStatus.setOnClickListener {
        if (!business!!.businessData!![position].businessHours.isNullOrEmpty()) {
            eventsListeners.showHours(position)
        }
    }
    holder.cl1.setOnClickListener {
        eventsListeners.openLocation(position)
    }
    holder.cl2.setOnClickListener {
        eventsListeners.makeCall(position)
    }
}

```

```

holder.cl3.setOnClickListener {
    eventsListeners.openOffer(position)
}
holder.cl4.setOnClickListener {
    eventsListeners.makeFavorite(position)
}
holder.cl5.setOnClickListener {
    eventsListeners.openGallery(position)
}
holder.cl6.setOnClickListener {
    var bundle = Bundle()
    bundle.putInt("businessId", business.businessData!![position].businessId)
    bundle.putString("businessName", business.businessData!![position].businessName)
    bundle.putString("imageUrl", business.businessData!![position].defaultPictureUrl)
    bundle.putDouble("rate", business.businessData!![position].rating)
    bundle.putInt("count", business.businessData!![position].noOfRatings)
    if (it.findNavController().currentDestination?.id == R.id.categoryBusinessFragment)
        it.findNavController()
        .navigate(R.id.action_categoryBusinessFragment_to_ratingFragment, bundle)
}
// holder.ivProfilePic.setOnClickListener {
//     eventsListeners.viewImage(position, 0)
// }
//
// holder.ivBanner.setOnClickListener {
//     eventsListeners.viewImage(position, 1)
// }
holder.itemView.setOnClickListener {
    Utils().clickStream(
        context,
        sharedPreferences.getString("USER_ID", "0")!!,
        "16",
        "${business!!.businessData!![position].businessId}",
        "8",
        "${business!!.businessData!![position].businessId}",
        ""
    )
    eventsListeners.closeDialog()
    CategoryBusinessFragment.scrollPage = false
    var bundle = Bundle()
    bundle.putParcelable("data", business!!.businessData!![position])
    if (it.findNavController().currentDestination?.id == R.id.categoryBusinessFragment)
        it.findNavController().navigate(
            R.id.action_categoryBusinessFragment_to_categoryBusinessDetailsFragment,

```

```

        bundle
    )
}
}
class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val ivBanner: ImageView = view.ivBanner
    val ivProfilePic: ImageView = view.ivProfilePic
    val tvTitle: TextView = view.tvTitle
    val tvDistance: TextView = view.tvDistance
    val cl1: ConstraintLayout = view.cl1
    val cl2: ConstraintLayout = view.cl2
    val cl3: ConstraintLayout = view.cl3
    val cl4: ConstraintLayout = view.cl4
    val clFav: ConstraintLayout = view.clFav
    val cl5: ConstraintLayout = view.cl5
    val cl6: ConstraintLayout = view.cl6
    val tvRatingValue: TextView = view.tvRatingValue
    val cv6: MaterialCardView = view.cv6
    val ivRatingStar: ImageView = view.ivRatingStar
    val ivFavorite: ImageView = view.ivFavorite
    val recyclerViewProductsItems: RecyclerView = view.recyclerViewProductsItems
    val tvOfferCount: TextView = view.tvOfferCount
    val tvProductCount: TextView = view.tvProductCount
    val ivAutoDetectArrow: ImageView = view.ivAutoDetectArrow
    val tvCategory: TextView = view.tvCategory
    val tvMatch: TextView = view.tvMatch
    val tvStatus: TextView = view.tvStatus
    val ivArrow: ImageView = view.ivArrow
    val tvDelivery: ImageView = view.tvDelivery
}
}

```

## 6.7. Review\_Adapter

```

package com.app.pепens.adapter
import android.content.Context
import android.content.res.ColorStateList
import android.graphics.Color
import android.os.Build
import android.view.LayoutInflater
import android.view.View

```

```

import android.view.ViewGroup
import android.widget.ImageView
import android.widget.RatingBar
import android.widget.TextView
import android.widget.Toast
import androidx.annotation.RequiresApi
import androidx.constraintlayout.widget.ConstraintLayout
import androidx.recyclerview.widget.RecyclerView
import com.app.pepens.R
import com.app.pepens.fragment.RatingFragment
import com.app.pepens.model.Rating.Rating
import com.app.pepens.utils.Utils
import com.bumptech.glide.Glide
import com.google.android.material.button.MaterialButton
import kotlinx.android.synthetic.main.custom_review_layout.view.*
class ReviewAdapter :
    RecyclerView.Adapter<ReviewAdapter.ViewHolder> {
    private val context: Context
    private val rating: Rating
    private val clickEvent: ClickEvent
    constructor(
        context: Context,
        rating: Rating,
        ratingFragment: RatingFragment
    ) : super() {
        this.context = context
        this.rating = rating
        this.clickEvent = ratingFragment
    }
    interface ClickEvent {
        fun itemClick(position: Int)
    }
    override fun onCreateViewHolder(viewGroup: ViewGroup, position: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context).inflate(R.layout.custom_review_layout, viewGroup, false)
        )
    }
    override fun getItemCount(): Int {
        return if (rating.ratingData.reviews.isNullOrEmpty()) {
            0
        } else {
            rating.ratingData.reviews.size
        }
    }
}

```

```

    }
}
@RequiresApi(Build.VERSION_CODES.LOLLIPOP)
override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {
    if (rating.ratingData.reviews[position].reviewId != 0) {
        var sharedPreferences = context.getSharedPreferences("USER_DATA",
Context.MODE_PRIVATE)
        viewHolder.viewReview.visibility = View.GONE
        Glide.with(context)
            .load(sharedPreferences.getString("DEFAULT_PICTURE_URL", null))
            .into(viewHolder.ivLogo)
        viewHolder.layoutReview.visibility = View.GONE
        viewHolder.layoutEditableReview.visibility = View.VISIBLE
//        viewHolder.tvDate.text =

Utils().getDate(rating.ratingData.reviews[position].reviewOn!!)
        viewHolder.tvReviewEditable.text =
rating.ratingData.reviews[position].reviewText//Editable.Factory.getInstance().newEditable(ratin
g.ratingData.reviews[position].reviewText)
        viewHolder.ratingBarEditable.rating =
            rating.ratingData.reviews[position].rating.toFloat()
        if (viewHolder.ratingBarEditable.rating == 1f) {
            viewHolder.ratingBarEditable.progressTintList =
ColorStateList.valueOf(Color.parseColor("#FF0000"))
            viewHolder.ratingBarEditable.secondaryProgressTintList =
ColorStateList.valueOf(Color.parseColor("#FF0000"))
        } else if (viewHolder.ratingBarEditable.rating == 2f || viewHolder.ratingBarEditable.rating
== 3f) {
            viewHolder.ratingBarEditable.progressTintList =
ColorStateList.valueOf(Color.parseColor("#FF8C00"))
            viewHolder.ratingBarEditable.secondaryProgressTintList =
ColorStateList.valueOf(Color.parseColor("#FF8C00"))
        } else if (viewHolder.ratingBarEditable.rating == 4f || viewHolder.ratingBarEditable.rating
== 5f) {
            viewHolder.ratingBarEditable.progressTintList =
ColorStateList.valueOf(Color.parseColor("#09AF00"))
            viewHolder.ratingBarEditable.secondaryProgressTintList =
ColorStateList.valueOf(Color.parseColor("#09AF00"))
        }
    } else {
        viewHolder.layoutEditableReview.visibility = View.GONE
        viewHolder.layoutReview.visibility = View.VISIBLE
        viewHolder.tvDate.text = Utils().getDate(rating.ratingData.reviews[position].reviewOn!!)
    }
}

```

```

viewHolder.tvReview.text = rating.ratingData.reviews[position].reviewText
viewHolder.ratingBar.rating = rating.ratingData.reviews[position].rating.toFloat()
viewHolder.tvName.text = rating.ratingData.reviews[position].userEntityName
Glide.with(context)
    .load(rating.ratingData.reviews[position].userEntityDefaultPictureUri)
    .into(viewHolder.ivLogoIcon)
viewHolder.viewReview.visibility = View.VISIBLE
if (viewHolder.ratingBar.rating == 1f) {
    viewHolder.ratingBar.progressTintList =
        ColorStateList.valueOf(Color.parseColor("#FF0000"))
} else if (viewHolder.ratingBar.rating == 2f || viewHolder.ratingBar.rating == 3f) {
    viewHolder.ratingBar.progressTintList =
        ColorStateList.valueOf(Color.parseColor("#FF8C00"))
} else if (viewHolder.ratingBar.rating == 4f || viewHolder.ratingBar.rating == 5f) {
    viewHolder.ratingBar.progressTintList =
        ColorStateList.valueOf(Color.parseColor("#09AF00"))
}
viewHolder.tvDate.setOnClickListener {
    Toast.makeText(
        context,
        "${Utils().getDateTime(rating.ratingData.reviews[position].reviewOn!!)}",
        Toast.LENGTH_SHORT
    ).show()
}
}
viewHolder.btnEdit.setOnClickListener {
    clickEvent.itemClick(position)
//    val popup = PopupMenu(context, viewHolder.ivMore)
//    popup.inflate(R.menu.review_menu)
//    popup.setOnMenuItemClickListener { item ->
//        when (item.itemId) {
//            R.id.action_edit -> {
//                popup.dismiss()
//                clickEvent.itemClick(position)
//                true
//            }
//            else -> false
//        }
//    }
//    popup.show()
}
}

```



```

class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val layoutEditableReview: ConstraintLayout = view.layoutEditableReview
    val ratingBarEditable: RatingBar = view.ratingBarEditable
    val ivLogo: ImageView = view.ivLogo
    val tvReviewEditable: TextView = view.tvReviewEditable
    val btnEdit: MaterialButton = view.btnEdit
    val layoutReview: ConstraintLayout = view.layoutReview
    var ivLogolcon: ImageView = view.ivLogolcon
    var tvName: TextView = view.tvName
    val ratingBar: RatingBar = view.ratingBar
    val tvDate: TextView = view.tvDate
    val tvReview: TextView = view.tvReview
    val viewReview: View = view.viewReview
}
}

```

## 6.8. Select\_Item\_Adapter

```

package com.app.pepens.adapter
import android.content.Context
import android.text.Editable
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.EditText
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import com.app.pepens.R
import com.app.pepens.model.SelectedItemsList
import kotlinx.android.synthetic.main.custom_item_discount_layout.view.*
class SelectedItemsAdapter : RecyclerView.Adapter<SelectedItemsAdapter.ViewHolder> {
    private var context: Context
    private var selectedItemList: ArrayList<SelectedItemsList>
    private var type: Int
    private var editable: Boolean
    constructor(
        context: Context,
        selectedItemList: ArrayList<SelectedItemsList>,
        type: Int,
        editable: Boolean) {
        this.context = context
        this.selectedItemList = selectedItemList
        this.type = type
    }
}

```

```

        this.editable = editable
    }
    override fun onCreateViewHolder(viewGroup: ViewGroup, position: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context)
                .inflate(R.layout.custom_item_discount_layout, viewGroup, false)
        )
    }
    override fun getItemCount(): Int {
        return selectedItemsList.size
    }
    override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {
        viewHolder.tvTitle.text = selectedItemsList[position].name
        viewHolder.etMrp.text =
            Editable.Factory.getInstance().newEditable(selectedItemsList[position].mrp)
        viewHolder.etSellingValue.text =
            Editable.Factory.getInstance().newEditable(selectedItemsList[position].sellingPrice)
        viewHolder.etMrp.background = null
        viewHolder.etMrp.isEnabled = false
        viewHolder.etSellingValue.background = null
        viewHolder.etSellingValue.isEnabled = false
    }
    class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
        val tvTitle: TextView = view.tvTitle
        val etMrp: EditText = view.etMrp
        val etSellingValue: EditText = view.etSellingValue
    }
}

```

## 6.9. Selected\_Product\_Item\_Adapter

```

package com.app.pemens.adapter
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView
import com.app.pemens.R
import com.app.pemens.fragment.CategoryBusinessFragment

```

```

import com.app.pepens.model.Business.Item
import kotlinx.android.synthetic.main.custom_selected_products_1_layout.view.*
class SelectedProductItemsAdapter :
RecyclerView.Adapter<SelectedProductItemsAdapter.ViewHolder> {
    private var context: Context
    private var businessItems: ArrayList<Item>
    private var updateEvent: UpdateEvent
    constructor(
        context: Context,
        businessItems: ArrayList<Item>,
        categoryBusinessFragment: CategoryBusinessFragment
    ){
        this.context = context
        this.businessItems = businessItems
        this.updateEvent = categoryBusinessFragment
    }
    interface UpdateEvent{
        fun removeItem(position: Int)
    }
    override fun onCreateViewHolder(viewGroup: ViewGroup, position: Int): ViewHolder {
        return ViewHolder(
            LayoutInflater.from(context)
                .inflate(R.layout.custom_selected_products_1_layout, viewGroup, false)
        )
    }
    override fun getItemCount(): Int {
        return if (businessItems == null) {
            0
        } else {
            businessItems!!.count()
        }
    }
    override fun onBindViewHolder(viewHolder: ViewHolder, position: Int) {
        viewHolder.tvTitle.text = businessItems!![position].itemName

        viewHolder.ivClose.setOnClickListener {
            updateEvent.removeItem(position)
        }
    }
}

```

```

class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val tvTitle: TextView = view.tvTitle
    val ivClose: ImageView = view.ivClose
}

}

```

## 6.10. Shopping\_Item\_Categories\_Adapter

```

package com.app.pemens.adapter
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.GridLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.app.pemens.R
import com.app.pemens.fragment.ShoppingCategoriesListFragment
import com.app.pemens.model.ItemCategories.ItemCategories
import com.bumptech.glide.Glide
import kotlinx.android.synthetic.main.custom_shopping_item_category_layout.view.*
import kotlinx.android.synthetic.main.custom_shopping_item_category_layout.view.tvDiscount
import kotlinx.android.synthetic.main.custom_shopping_item_category_layout.view.tvTitle
/**
 * Created by Vikas Kumar Singh on 15/08/20.
 */
class ShoppingItemCategoriesAdapter :
    RecyclerView.Adapter<ShoppingItemCategoriesAdapter.ViewHolder> {
    private var context: Context
    private var itemCategories: ItemCategories
    private var clickEvent: ClickEvent
    constructor(
        context: Context,
        itemCategories: ItemCategories,
        shoppingCategoriesListFragment: ShoppingCategoriesListFragment
    ) : super() {
        this.context = context
        this.itemCategories = itemCategories
        this.clickEvent = shoppingCategoriesListFragment
    }
}

```

```

interface ClickEvent {
    fun clickOnItem(position: Int)
}
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
    return ViewHolder(
        LayoutInflater.from(context)
            .inflate(R.layout.custom_shopping_item_category_layout, parent, false)
    )
}
override fun getItemCount(): Int {
    return if (itemCategories == null) {
        0
    } else {
        itemCategories!!.itemCategoriesData!!.count()
    }
}
override fun onBindViewHolder(holder: ViewHolder, position: Int) {
    Glide.with(context)
        .load(itemCategories!!.itemCategoriesData!![position].defaultPictureUrl)
        .placeholder(R.drawable.ic_category_icon)
        .into(holder.imageView)
    holder.tvTitle.text = itemCategories!!.itemCategoriesData!![position].itemCategoryName
    if (itemCategories!!.itemCategoriesData!![position].maxDiscountPercentage <= 0) {
        holder.tvDiscount.visibility = View.GONE
    } else {
        holder.tvDiscount.visibility = View.VISIBLE
        holder.tvDiscount.text = "Upto
    ${itemCategories!!.itemCategoriesData!![position].maxDiscountPercentage}% OFF"
    }
    holder.recyclerViewSubCategories!!.layoutManager = GridLayoutManager(context, 3)
    var itemsAdapter = ShoppingItemSubCategoriesAdapter(context,
itemCategories!!.itemCategoriesData!![position].itemCategories!!,
itemCategories!!.itemCategoriesData!![position].itemCategoryId)
    holder.recyclerViewSubCategories!!.adapter = itemsAdapter
    if (itemCategories!!.itemCategoriesData!![position].isOpen) {
        holder.ivArrow.setImageResource(R.drawable.ic_arrow_up)
        holder.recyclerViewSubCategories.visibility = View.VISIBLE
    } else {
        holder.ivArrow.setImageResource(R.drawable.ic_arrow_down)
        holder.recyclerViewSubCategories.visibility = View.GONE
    }

    holder.itemView.setOnClickListener {
        clickEvent.clickOnItem(position)
    }
}

```

```

    }
}

class ViewHolder(view: View) : RecyclerView.ViewHolder(view) {
    val imageView: ImageView = view.imageView
    val tvTitle: TextView = view.tvTitle
    val tvDiscount: TextView = view.tvDiscount
    val ivArrow: ImageView = view.ivArrow
    val recyclerViewSubCategories: RecyclerView = view.recyclerViewSubCategories
}
}

```

# CHAPTER 6

## BIBLIOGRAPHY

1. Pantanowitz L, Valenstein PN, Evans AJ, et al. Review of the current state of whole slide imaging in pathology. *J Pathol Inform.* 2011;2:36. doi:10.4103/2153-3539.83746
2. Gilbertson JR, Ho J, Anthony L, Jukic DM, Yagi Y, Parwani AV. Primary histologic diagnosis using automated whole slide imaging: a validation study. *BMC Clin Pathol.* 2006;6:4. doi:10.1186/1472-6890-6-4
3. Gilbertson JR, Patel AA, Yagi Y, et al. Digital imaging in pathology: the case for standardization. *J Telemed Telecare.* 2005;11(3):109-116. doi:10.1258/1357633054068983
4. Nielsen PS, Lindebjerg J, Rasmussen J, Starklint H, Waldstrøm M, Nielsen B. Virtual microscopy: an evaluation of its validity and diagnostic performance in routine histologic diagnosis of skin tumors. *Hum Pathol.* 2010;41(12):1770-1776. doi:10.1016/j.humpath.2010.07.001
5. Bauer TW, Schoenfield L, Slaw RJ, Yerian L, Sun Z, Henricks WH. Validation of whole slide imaging for primary diagnosis in surgical pathology. *Arch Pathol Lab Med.* 2013;137(4):518-524. doi:10.5858/arpa.2012-0114-OA
6. Kim GH, Kim HS, Oh YL, Choi YL, Kang GH. Automated slide scanning systems can increase efficiency in performing manual microdissection for molecular analysis. *J Pathol Inform.* 2016;7:49. doi:10.4103/2153-3539.194781
7. Vodovnik A. The automated slide analysis workflow: a combination of automated microscopy and image analysis for high-throughput analysis of histology slides. *J Pathol Inform.* 2019;10:27. doi:10.4103/jpi.jpi\_11\_19
8. Al-Janabi S, Huisman A, Vink A, Leguit RJ, Offerhaus GJ, Ten Kate FJ. Whole slide images for primary diagnostics of gastrointestinal tract pathology: a feasibility study. *Hum Pathol.* 2012;43(5):702-707. doi:10.1016/j.humpath.2011.07.011