



BEACONHOUSE NATIONAL UNIVERSITY

Wasail

PRJ-F23/333

REQUIREMENT ANALYSIS DOCUMENT

EXTERNAL SUPERVISOR

Hamza Zafar

INTERNAL SUPERVISOR

Huda Sarfraz

GROUP MEMBERS

Fatima Ali Tirmizi	F2020-718
Fizza Adeel	F2020-336
Irtaza Ahmed Khan	F2020-153
Malaika Sultan	F2020-661

SCHOOL OF COMPUTER & IT

Table of Contents

Introduction.....	6
Existing Systems.....	6
Literature Survey.....	12
Requirement Gathering and Fact Finding.....	14
Software Requirement Specification (SRS).....	17
System Environment.....	17
Machine Learning Requirements.....	17
Objective.....	17
Use Case.....	17
Data Collection.....	18
Local Grocery Stores.....	18
Local Vendor Dataset.....	18
Local Pharmacy Dataset.....	19
Kaggle.....	20
Corporación Favorita Grocery Sales Forecasting.....	20
Instacart Market Basket Analysis.....	25
Feature Engineering.....	26
Date.....	26
Product Name.....	26
Sales.....	26
Holiday.....	26
Location.....	27
Weather.....	27
ML Model Selection.....	27
Approach Considerations.....	27
Ensemble Approach.....	27
Non-Ensemble Approach.....	28
Type of ML Models.....	28
Autoregressive Models.....	28
Exponential Smoothing Models.....	28
Machine Learning Regression Models.....	28
Ensemble Models.....	28
Deep Learning Models.....	29
Competitive Analysis Models.....	29
Model Training and Testing.....	29
Integration with User Interface.....	30
Recommendation Updates.....	31
Feedback and Learning.....	31
Performance and Scalability.....	31
Testing and Validation.....	31

User Roles.....	32
User Stories.....	32
User Story: Placing the Order.....	32
User Story: Searching for a Product.....	33
Functional Requirements.....	34
Grocery Store and Vendor (FR1).....	34
FR1.1: Language Selection.....	34
FR1.2: Phone Registration.....	35
FR1.3: Phone Number Confirmation.....	36
FR1.4: Phone Number Exists.....	37
FR1.5: OTP Code Generation and Delivery.....	37
FR1.6: Login.....	39
FR1.7: Logout.....	40
FR1.8: Reset Password.....	40
FR1.9: View Profile.....	40
Grocery Store (FR2).....	41
FR2.9: Account Details.....	41
FR2.10: Search Product.....	42
FR2.11: Search Vendor.....	43
FR2.12: Browse Category.....	44
FR2.13: View Vendor Profile.....	45
FR2.14: Add Vendor to Vendor List.....	46
FR2.15: Contact Vendor.....	47
FR2.16: View Products on the Vendor's Profile.....	48
FR2.17: View Searched Product.....	49
FR2.18: Select Products.....	50
FR2.19: Order Recommendation.....	51
FR2.20: Quantity Selection.....	52
FR2.21: Remove Product.....	53
FR2.22: Order Placement.....	54
FR2.23: View Order.....	55
FR2.24: Order Tracking.....	56
FR2.25: View Vendor List.....	57
FR2.26: View Order History.....	58
FR2.27: Edit Profile.....	58
FR2.28: Notification.....	58
Vendor (FR3).....	59
FR3.9: Vendor Registration.....	59
FR3.10: Valid Password.....	59
FR3.11: Username Exists.....	59
FR3.12: Search Product in Inventory.....	60

FR3.13: Add Product to Inventory.....	61
FR3.14: All Products Search.....	62
FR3.15: Remove Product.....	63
FR3.16: Edit Details of Product.....	64
FR3.17: View Inventory.....	65
FR3.18: View Current Orders.....	66
FR3.19: View Grocery Stores List.....	67
FR3.20: View Grocery Store Profile.....	68
FR3.21: View Grocery Store Current Order.....	69
FR3.22: View Orders History.....	70
FR3.23: Order Dispatch Tracking.....	71
FR3.24: Edit Profile.....	72
FR3.25: Restrict Product Duplication.....	72
Admin Portal (FR4).....	73
FR4.1: Add New User.....	73
FR4.2: Add New User Details.....	74
FR4.3: User Login.....	75
FR4.4: Edit User Profile.....	76
FR4.5: Delete User Profile.....	77
FR4.6: View Grocery Store's Profile.....	78
FR4.7: Disable Grocery Store's Profile.....	79
FR4.8: View Vendor's Profile.....	80
FR4.9: Disable Vendor's Profile.....	81
FR4.10: Select Grocery Store's ML Model.....	82
FR4.11: Display RMSLE value.....	83
FR4.12: Display Analytics.....	84
FR4.13: Add Category.....	85
FR4.14: Update Category.....	86
FR4.15: Delete Category.....	87
FR4.16: Add Product.....	88
FR4.17: Update Product.....	89
FR4.18: Delete Product.....	90
Non-Functional Requirements.....	91
• Performance.....	91
• Security.....	91
• Usability.....	91
• Data Storage.....	91
• Notification System.....	91
• Error Handling.....	91
• Reporting and Logging.....	91
• Integration.....	91
• Mobile Responsiveness.....	91

Test Cases.....	92
Future Improvements.....	111
References.....	112
Appendix.....	114
Interview Transcripts.....	114
Z Mart.....	114
Express Store.....	115

Revision Chart

Version	Description	Date
1	Adding functional requirements	10-01-2024
2	Adding test cases	15-01-2024

Introduction

Inefficiencies, wastage and imprecise demand forecasting due to traditional inventory management processes have been an emerging challenge in the landscape of the food industry, specifically the supply chain system. There is a dire need for digitisation and automation within the industry, as evident by businesses' inability to streamline processes, predict demand accurately, and optimise resource usage. This deficiency hinders the possibility of an agile and responsive supply chain imperative to today's data-driven decision-making.

To tackle the aforementioned issue, our proposed solution employs advanced data analytics, specifically Machine Learning (ML), to enhance the supply chain in grocery and vendor retail and address current supply chain vulnerabilities. This will be done through an accurately predicted customer demand, owing to a demand forecasting system, having utilised sales, holiday, location, exchange rate (or fuel prices), and weather data.

The system will be integrated into a mobile-based platform connecting grocery stores with vendors to enable an efficient and profit-maximising operability. An intuitive platform design of a platform such as ours would assist sales and purchase of goods by reducing physical visits made by vendors and enhancing operational efficiency through a tracking feature for grocery stores. Novice grocery stores would be assisted in searching for specific products from reliable registered vendors.

This platform stands out from *Tajir*, *Dastgyr*, *Jugnu*, *Bazaar*, and *Retailo* with the aid of a ML based demand forecasting system. This move toward modernisation and automation sets it apart in pursuing the same goal. As efficiency and accuracy arise in the inventory management, the precision by ML will allow stores to order optimal quantities, mitigating risks of overstocking and understocking.

This approach, inspired by the insights presented in 'Advances in Supply Chain Management' [\[1\]](#), aims to redefine forecasting, curb waste, and optimise operational efficiency for grocery stores and vendors. Ultimately, the solution would strive to augment overall profitability in the food industry and create a smarter supply chain system.

Existing Systems

In the current supply chain scenario in Pakistan, the movement of products relies heavily on outdated and paperwork-intensive processes. This traditional approach poses significant challenges, including vulnerability to disruptions, limited visibility due to infrastructure constraints, and communication hurdles. Retailers face stockouts, unpredictable supply arrivals, and a slow replenishment process, leading to operational inefficiencies and working capital restrictions. The conventional procurement structure demands substantial time investment, with retailers spending an average of 25 hours per week navigating wholesale markets [\[2\]](#).

A notable player addressing these challenges is **Tajir**, shown in Figure 1.1, it is a platform revolutionising inventory management for mom-and-pop stores in Pakistan. Tajir acts as a vendor, offering a seamless solution for purchasing inventory. It enables retailers to order at their convenience, receive on-demand deliveries, access transparent pricing, and choose from an extensive product selection. Inspired by successes like Tajir, our proposed software solution aims to complement and enhance the existing landscape by providing unique features tailored to the specific needs of small and medium retailers in Pakistan.

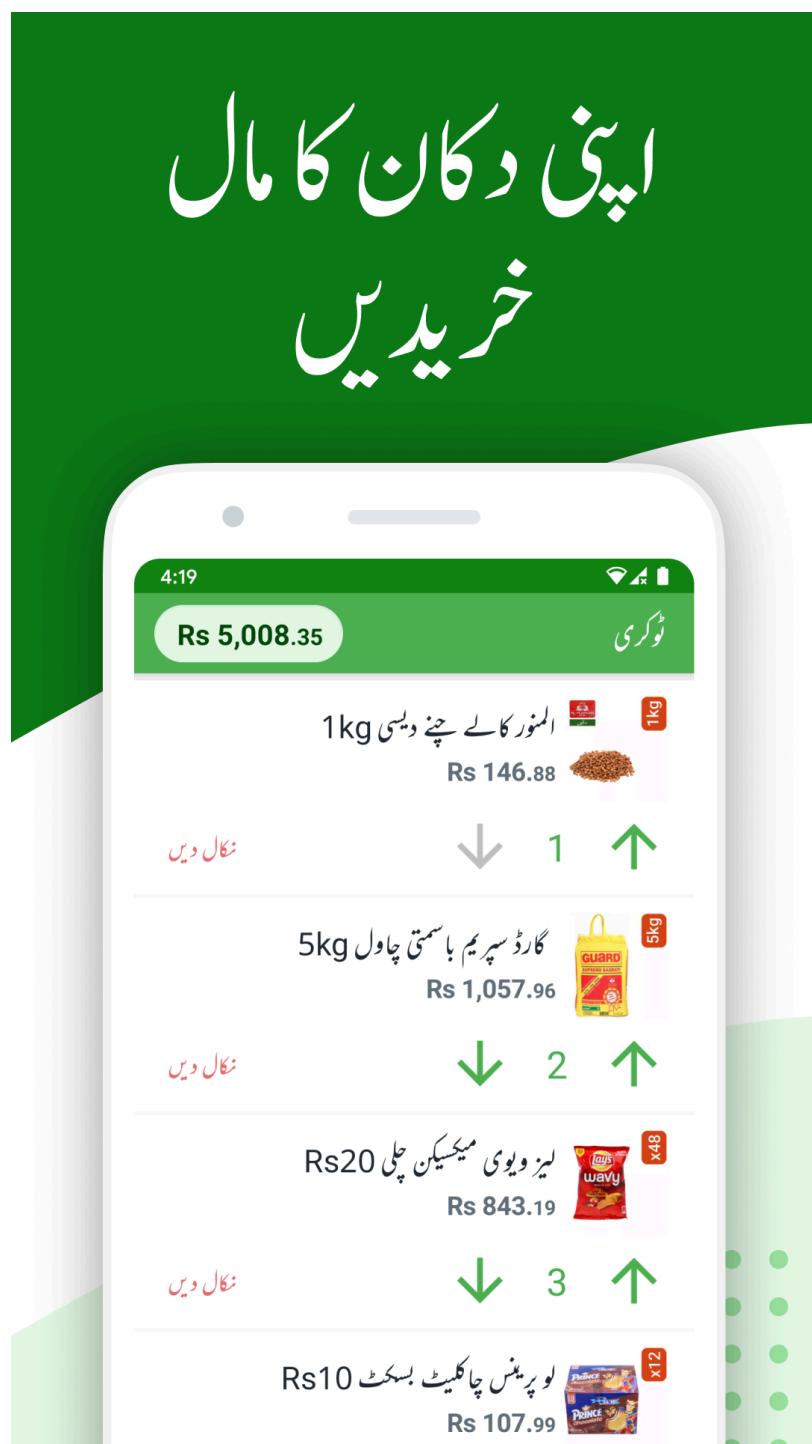


Figure 1.1 Tajir

Adding to this is **Jugnu**, a Business to Business (B2B) e-commerce platform founded in 2020. Jugnu, seen in Figure 1.2 strives for social and economic empowerment by connecting large suppliers to small and medium enterprises, driving growth in local economies. The platform operates across major cities in Pakistan, offering a user-friendly platform for ordering and receiving deliveries within a 24-hour window.

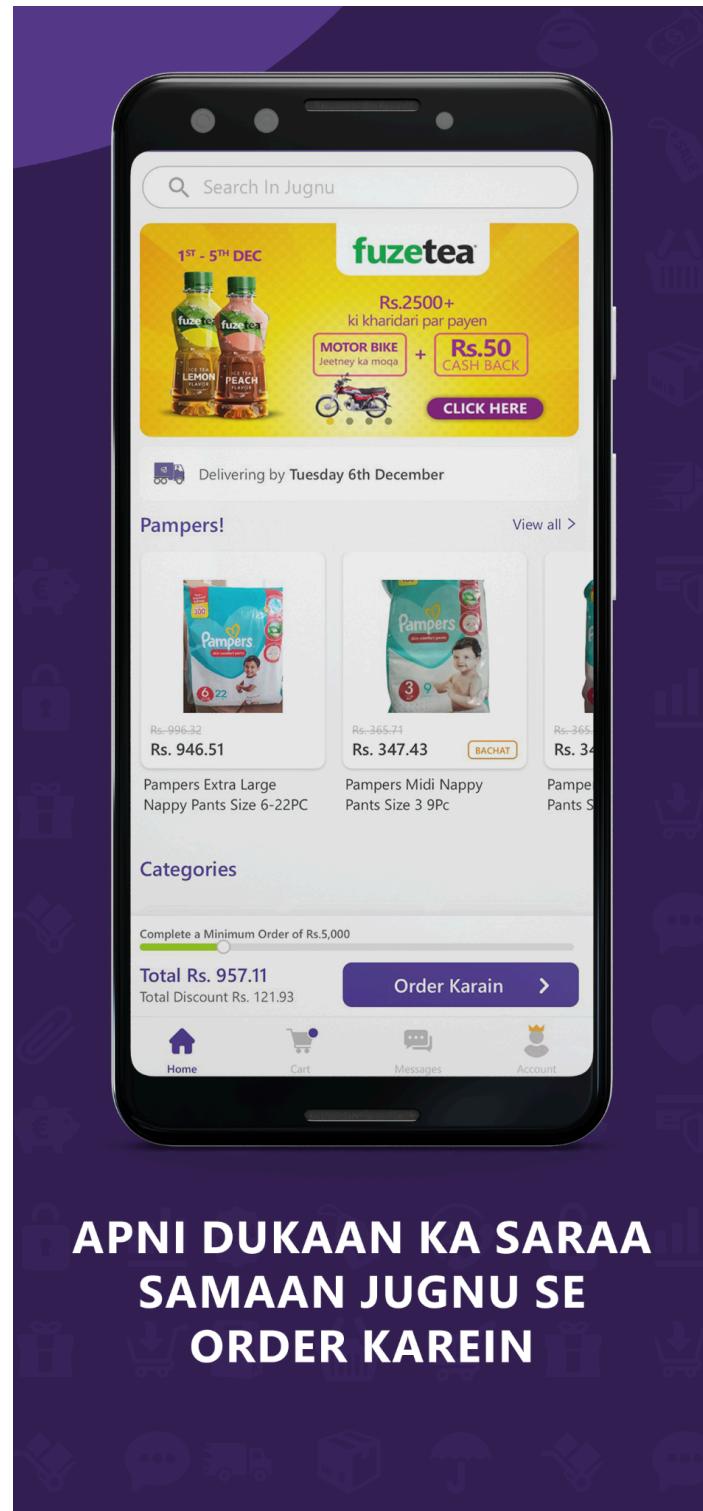


Figure 1.2 Jugnu

Below in Figure 1.3 is **Retailo**, another B2B marketplace, allows retailers to restock their shops conveniently with features like instant price comparisons and next-day delivery, eliminating the need for multiple distributors and weekly restocking hassles.

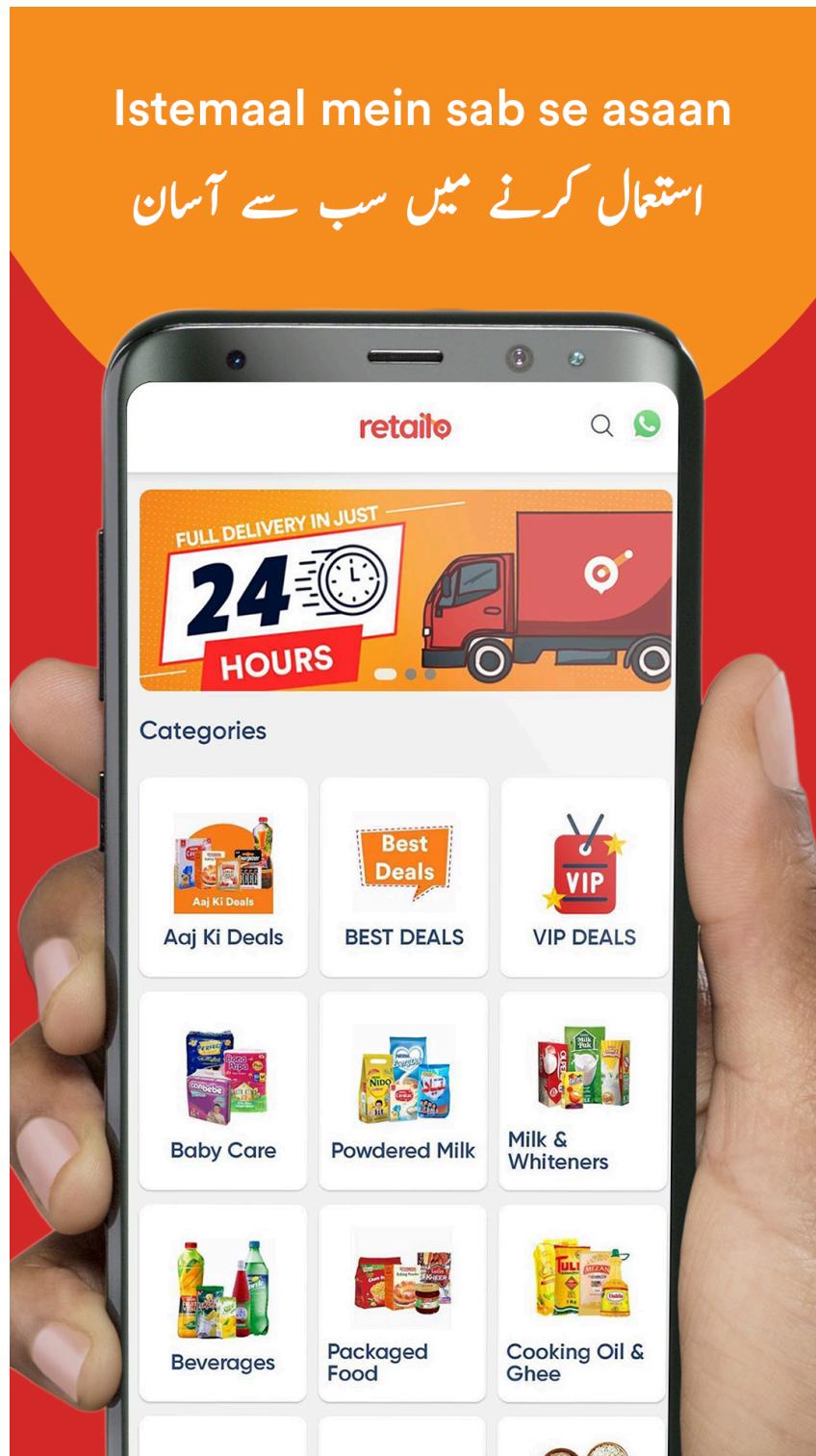


Figure 1.3 Retailo

Dastgyr, another B2B marketplace, addresses the pain points of small retailers by connecting them with manufacturers and suppliers. It offers a one-stop solution for inventory needs, allowing retailers to place orders in seconds, see Figure 1.4, and receive on-demand delivery. Additionally, retailers working with Dastgyr for three months become eligible for credit purchases, addressing the challenge of limited access to external capital for growth.

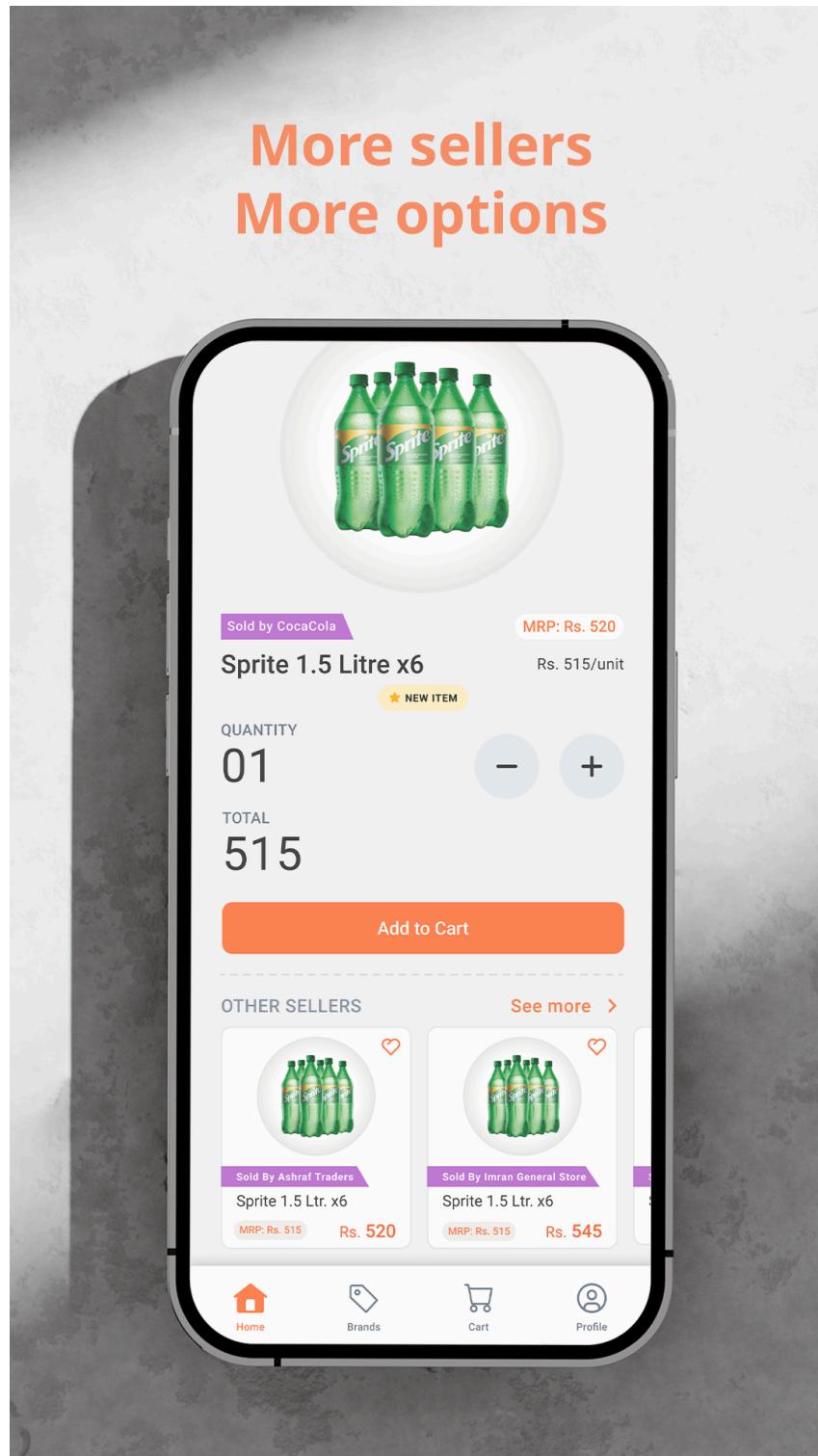


Figure 1.4 Dastgyr

Next is **Bazaar** in Figure 1.5 which contributes to digitising and growing businesses in Pakistan through its mobile app, providing small business owners access to a wide assortment of goods with free next-day delivery.



Figure 1.5 Bazaar

Additionally, we also explored **Candela RMS**, an enterprise retail software solution focusing on inventory management and POS for all kinds of retail. Candela RMS offers features like '*Manage Inventory Shrinkage*' which is essentially a reorder level feature that sends an alert when an item in the inventory reaches a lower limit and '*Edibles Expiry Management*' another feature that allows the entry of expiration dates during stock entry and subsequently printing the expiry date along with the barcode. While this feature aids in managing the expiration of products, it doesn't directly address the nuanced challenge of demand forecasting.

As we undertake this software solution project, we draw insights from the successes of these platforms, aiming to contribute to the modernisation and resilience of the supply chain ecosystem in Pakistan. Our market survey and research findings indicate a gap in the current solutions, with a need for a system that not only helps manage inventory but also accurately predicts demand and facilitates smooth communication between grocery stores and vendors. By incorporating elements from these innovative solutions, we seek to offer a comprehensive and effective software solution tailored to the specific challenges faced by small and medium grocery store retailers in the country.

Literature Survey

In addressing the multifaceted challenges of the food industry, recent studies have illuminated innovative solutions, offering insights that span crucial domains. One study explored the effectiveness of high-tech inventory management within supermarkets, attributing a significant 56.7% of their performance to automation [3]. Another research framework proposed the strategic implementation of AI and robotics to combat food loss during the pandemic, emphasising sensory enhancement and collaborative automation [4].

Recent advancements in deep learning, as highlighted in various studies, signal a transformative shift in the landscape of demand forecasting. Techniques such as multi-modal sales forecasting networks and the application of LSTM demonstrate superior accuracy and effectiveness [5, 6]. The debate between traditional and machine learning forecasting emerges, with a study showcasing the promise of a support vector machine in handling multiple demand series [7].

A groundbreaking study utilised low-cost sensors and machine learning to achieve a remarkable 92.65% accuracy in predicting for preventing food wastage, addressing a critical concern in the industry [8]. Global food supply chains are explored in another study, emphasising the importance of efficiency and behaviour change, particularly in affluent economies, to combat waste [9]. Strategies for proactive food waste reduction in the grocery sector are elucidated in a study that carefully balances customer satisfaction and inventory management [10].

The evolving role of e-grocery as an alternative to traditional retailing is highlighted, emphasising in-stock availability in customer decisions [11]. Generalised Additive Models for Location, Scale, and Shape (GAMLSS) are recommended in another exploration, specifically focusing on-demand distribution tails in e-grocery [12]. A simulation model dissects the benefits and drawbacks of Vendor-Managed Inventory (VMI) in the grocery supply chain, revealing that manufacturers reap more significant benefits from VMI adoption [13].

The chocolate industry is subject to a study employing machine learning for refined predictions based on regular and promotional sales data [14]. Retail firms, including Walmart, Costco, and Kroger, are analysed as economic indicators through statistical regression and machine learning, exposing operational inefficiencies [15]. The study on 'Corporacion Favorita,' a major grocery chain in Ecuador, offers insights into optimising predictions and mitigating stock-out and over-stocking issues [16].

In addition, abroad there are platforms like **Shelf Engine** shown in Figure 2.1 and **Guac** shown in Figure 2.2, both leverage machine learning for demand forecasting and order optimisation in the grocery retail sector. The major difference lies in their approach; Guac provides recommendations, leaving the final order decision to stores, while Shelf Engine actively decides orders and assists in placing them with

vendors. Drawing inspiration from global industry trends, our proposed system aligns with the approach of Guac, aiming to leverage technology, embrace automation, and address current vulnerabilities in our supply chain. Despite notable progress in these studies and platforms, challenges such as training set size, overfitting, and model complexity persist, necessitating further exploration for the development of a responsive and sustainable food system.

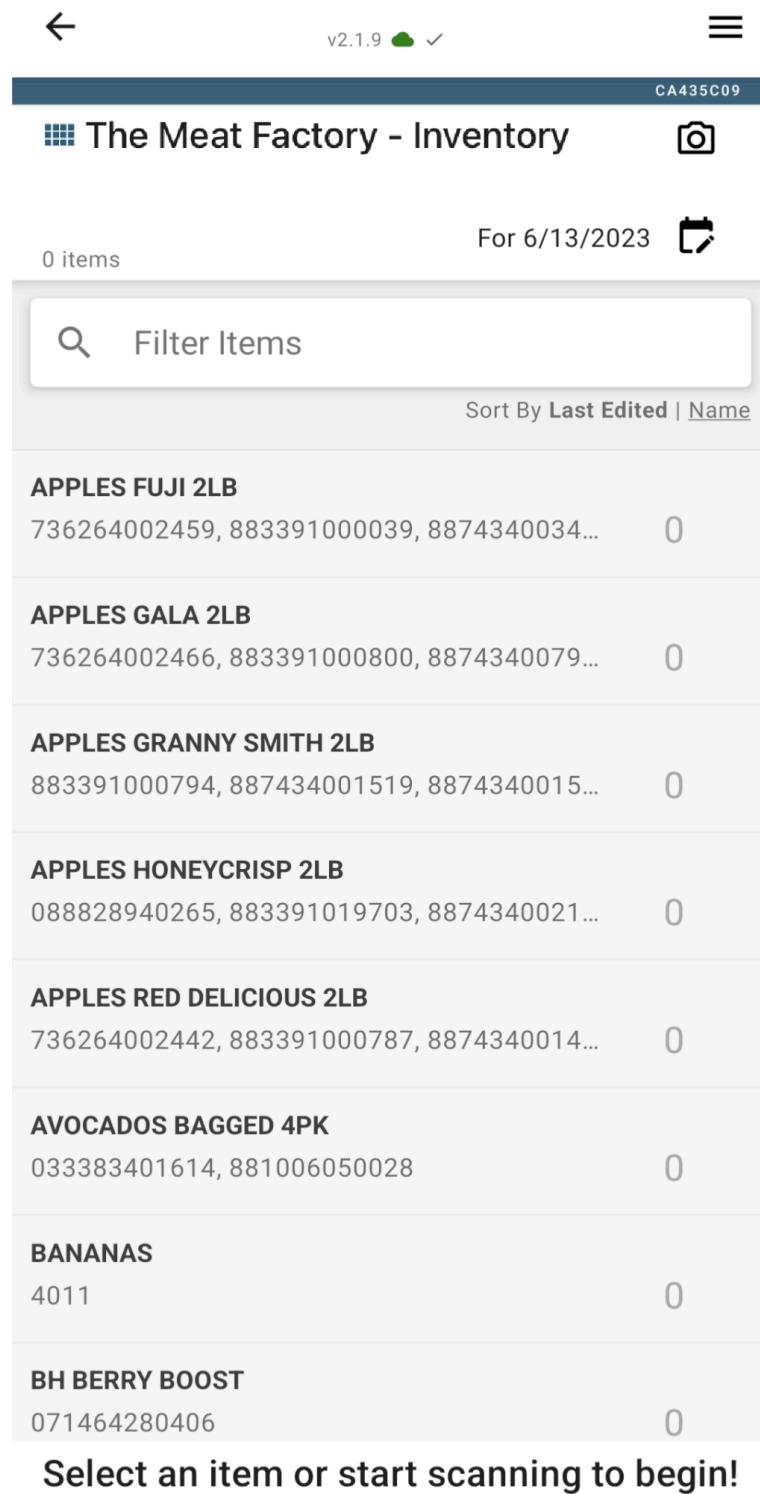


Figure 2.1 Shelf Engine

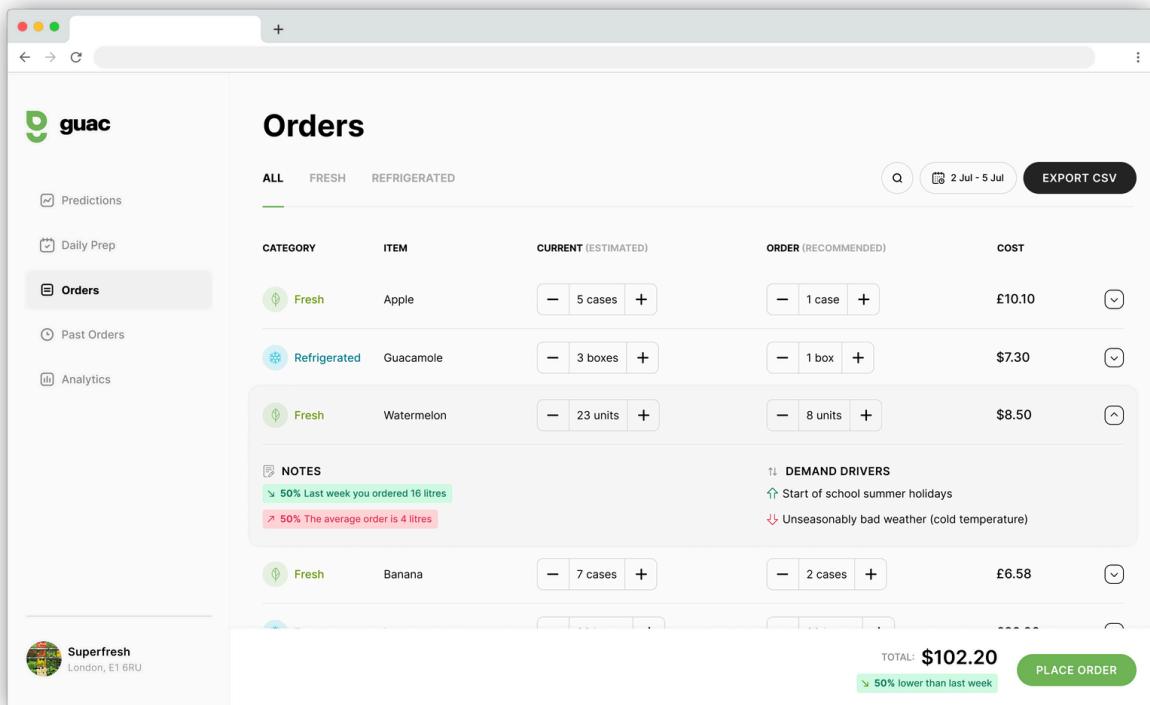


Figure 2.2 Guac

Requirement Gathering and Fact Finding

We engaged grocery stores through unstructured interviews (transcripts in Appendix) and navigating the stores to discuss concepts like strategic shelf placement for marketing and promotion.

Data Collection Technique: Interviews and Observation

Reasoning: Interviews provide a strategic and adaptable approach, capturing detailed, context-specific information essential for understanding the challenges faced by grocery store owners in the supply chain.

Issues Identified

Amongst the challenges found, an outdated ordering system is the biggest one. This manifests in discounted products nearing expiration made available at stores as seen in Figure 3.1, which raises concerns over the quality and safe use, and monopolisation of certain products where grocery stores do not allow specific products to be sold by vendors to any other grocery store than them e.g. a grocery store in a housing society tells the vendor that they will only purchase a specific product if the vendor does not sell it to any other store in that housing society. In-person interviews highlighted the impact of these issues, emphasising that existing tools (inventory management systems and point of sale systems) are

rendered ineffective due to their complexity. Even grocery store workers who are tech-savvy try to avoid inventory-related tasks, such as adding received inventory. They only use it for selling as it makes the process of selling faster and more convenient for them. Notably, the grocery store workforce is not technologically inept; rather, the current inventory systems are user-unfriendly. There is also an issue of '*mobilers*', a community term for individuals that are fake/pretend vendors selling either fake or expired products.



Figure 3.1 Jalal Son's

Summarised

An outdated ordering system that has consequences such as:

- Existing tools are too difficult to use for grocery stores
- 'Mobilers' selling fake or expired products
- Limited vendor options for new products discoverability
- Product Monopolisation

Interview Insights

Insights into the ordering process included manual observation of daily sales i.e. qualitative analysis by the grocery store owner to decide the quantity for the next order, with budget constraints such as inflation leading to reduced stock purchases. Some grocery stores do not have access to specific products by a shared vendor on demand of another grocery store.

On the digital frontier, software challenges include a dependency on electricity, a lack of backup during power outages, and manual data entry during disruptions. The workforce's adeptness with technology clashes with the current software's complexity, leading to underutilization.

Summarised

1. Ordering Process:

- Qualitative analysis to determine order quantities
- Budget constraints impact stock purchasing decisions
- Limited access to specific products due to shared vendor restrictions (grocery stores restricting vendors to only sell a specific product to them)

2. Software Issues:

- Dependency on electricity for software usage
- Lack of backup in case of power outages
- Manual data entry during power disruptions

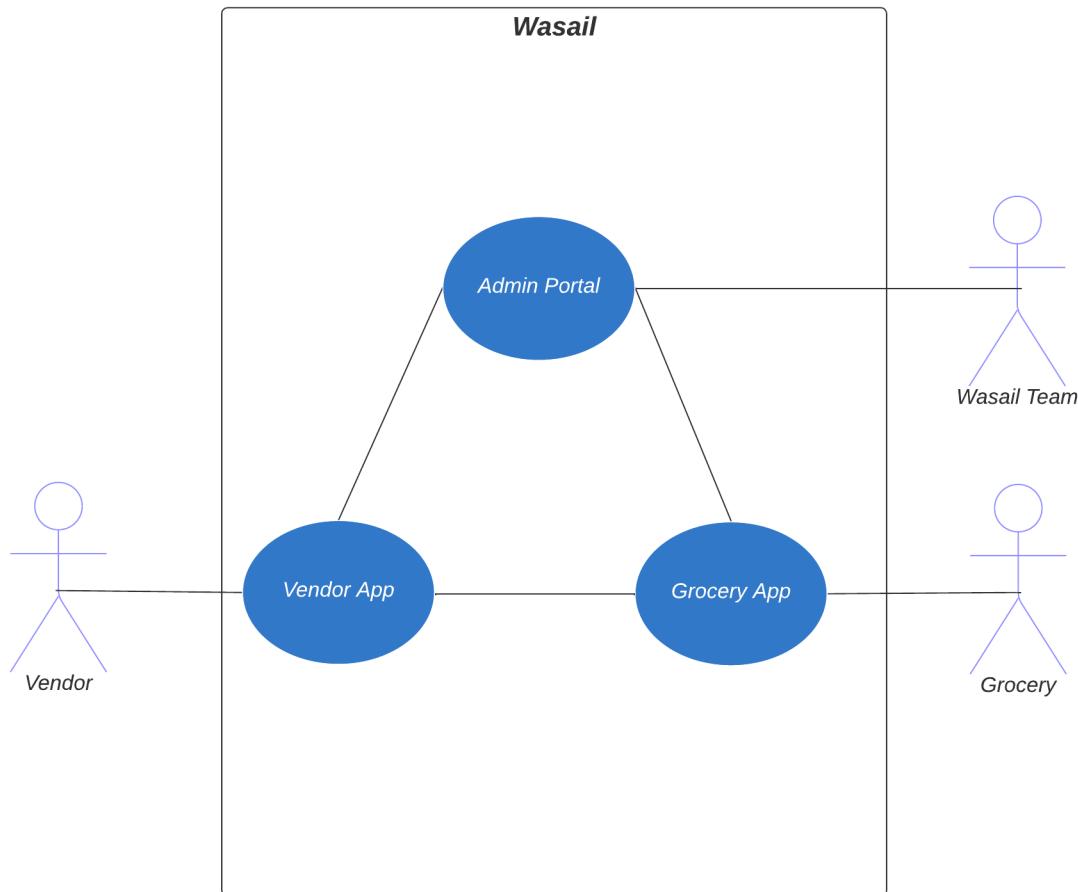
Conclusion: The insights gathered through interviews with grocery stores highlight critical issues in the current supply chain, forming a crucial foundation for the development of our software solution. The experiences shared by Z Mart, Express Store, and My Store underscore the necessity for a digital solution to tackle challenges related to order management, inventory control, and system reliability. Furthermore, the additional research reveals significant gaps in the existing mechanisms, including the absence of a formal communication channel with vendors, authentication issues, and a reliance on your vendors as the sole source for new products.

Potential Improvements & Proposed Solution:

- **Demand Forecasting and Automation:** Develop a sophisticated system for demand-forecasting, integrated in a platform managing orders and distribution, supported by machine learning and use variables like sales trends, weather, location, and holidays, optimising the inventory management.
- **Waste Reduction and Profitability Augmentation:** Redefine forecasting to curb food wastage whilst optimising operational efficiency for both grocery stores and vendors. As sales increase so does profitability.
- **Streamlining Processes:** Introduce an easier-to-use system for a seamless user experience.
- **Increased Visibility:** Enhance vendor lookup and search functionalities.
- **Collaborative Platform:** Verify the legitimacy of vendors for a secure collaborative environment and reduce physical visits to grocery stores by vendors through the platform.

Software Requirement Specification (SRS)

System Environment



Machine Learning Requirements

Objective

Provide a recommended amount of product to order for the grocery stores.

Use Case

"As a grocery store, I want to receive a recommended amount of product to order, based on my previous sales, that would ensure maximum profit."

Data Collection

Local Grocery Stores

The following list of grocery stores have verbally agreed to provide us with data:

- [Rubaika Cash & Carry](#)
- [My Super Store](#)
- Z Mart (Falcon Complex)
- Express Store (Askari 5)

Local Vendor Dataset

Attributes:

1. Item ID
2. Item Description
3. Date
4. Qty Received
5. Item Cost
6. Actual Cost
7. Assembly Qty
8. Assembly (\$)
9. Adjust Qty
10. Adjust (\$)
11. Quantity Sold
12. Cost of Sales
13. Remaining Qty
14. Remain Value

Instances: 16,000

Format: CSV

Time Frame: October 2019 to December 2021

Description: The data, seen in Figure 4.1, needs to be cleaned. Assembly Qty, Assembly (\$), Adjust Qty, and Adjust (\$) need to be removed as they are mostly empty. Instances where a product has been purchased (received), the attributes related to sales are empty and vice versa. Hence, the data needs to be separated into two different datasets, one for purchases and one for sales, to get rid of all the empty cells. Analysis of this dataset aids to better understanding of how vendors sell and grocery store purchase products. It can also be used in the future to offer demand forecasting for vendors as well. This dataset is uploaded on our GitHub repository.

Item ID	Item Description	Date	Qty Received	Item Cost	Actual Cost	Assembly Qty	Assembly (\$)	Adjust Qty	Adjust (\$)	Quantity Sold	Cost of Sales	Remaining Qty	Remain Value
AAP001	AGAR AGAR POWDER 25G											100.00	29,744.00
AAP001	AGAR AGAR POWDER 25G	1/27/20	100.00	297.44	29,744.00							100.00	29,744.00
AAP001	AGAR AGAR POWDER 25G	1/29/20										100.00	29,744.00
ABC01	ABC SWEET SOY SAUCE 500ML											96.00	69,233.28
ABC01	ABC SWEET SOY SAUCE 500ML	7/24/20	96.00	721.18	69,233.28							96.00	69,233.28
ABC01	ABC SWEET SOY SAUCE 500ML	7/25/20										72.00	51,924.96
ABC01	ABC SWEET SOY SAUCE 500ML	7/25/20										72.00	51,924.96
ABC01	ABC SWEET SOY SAUCE 500ML	8/7/20	120.00	710.25	85,230.00							120.00	85,230.00
ABC01	ABC SWEET SOY SAUCE 500ML	8/8/20										36.00	25,569.00
ABC01	ABC SWEET SOY SAUCE 500ML	8/11/20										84.00	59,661.00
ABC01	ABC SWEET SOY SAUCE 500ML	8/23/20	72.00	672.60	48,427.20							72.00	48,427.20
ABC01	ABC SWEET SOY SAUCE 500ML	8/25/20										72.00	48,427.20
ABC01	ABC SWEET SOY SAUCE 500ML	9/7/20	72.00	672.60	48,427.20							72.00	48,427.20
ABC01	ABC SWEET SOY SAUCE 500ML	9/12/20										48.00	32,284.80
ABC01	ABC SWEET SOY SAUCE 500ML	9/27/20	288.00	714.30	205,718.40							312.00	221,860.80
ABC01	ABC SWEET SOY SAUCE 500ML	9/30/20										60.00	41,857.20
ABC01	ABC SWEET SOY SAUCE 500ML	11/3/20										60.00	42,858.00
ABC01	ABC SWEET SOY SAUCE 500ML	11/21/20										120.00	85,716.00
ABC01	ABC SWEET SOY SAUCE 500ML	11/24/20										68.00	48,572.40
ABC01	ABC SWEET SOY SAUCE 500ML	1/31/21										4.00	2,857.20
AC01	AMERI COLOR												
ACC01	ANCHOR CHEDDAR KG												
ACTC01	ANTICA CANTINA SALTED 200G												
ACTC01	ANTICA CANTINA SALTED 200G	8/29/20										29.00	18,880.74
ACTC01	ANTICA CANTINA SALTED 200G	8/29/20	30.00	651.06	19,531.80							1.00	651.06
ACTC02	ANTICA CANTINA CHILLI 200G												
ACTC02	ANTICA CANTINA CHILLI 200G	8/29/20										30.00	19,531.80
ACTC02	ANTICA CANTINA CHILLI 200G	8/29/20	30.00	651.06	19,531.80							-30.00	-19,531.80
ACTC03	ANTICA CANTINA N.CHEESE 200G											30.00	19,531.80
ACTC03	ANTICA CANTINA N.CHEESE 200G	8/29/20										30.00	19,531.80
ACTC03	ANTICA CANTINA N.CHEESE 200G	8/29/20	30.00	651.06	19,531.80							-30.00	-19,531.80
ACTC04	ANTICA CANTINA BBQ 200G											30.00	19,531.80
ACTC04	ANTICA CANTINA BBQ 200G	8/29/20										30.00	19,531.80
ACTC04	ANTICA CANTINA BBQ 200G	8/29/20	30.00	651.06	19,531.80							-30.00	-19,531.80
ACTD01	ANTICA CANTINA MILD SALSA 300G											12.00	10,982.28
ACTD01	ANTICA CANTINA MILD SALSA 300G	8/29/20										12.00	10,982.28
ACTD01	ANTICA CANTINA MILD SALSA 300G	8/29/20	12.00	915.19	10,982.28								
ACTD02	ANTICA CANTINA MED SALSA 300G												
ACTD02	ANTICA CANTINA MED SALSA 300G	8/29/20										12.00	10,982.28
ACTD02	ANTICA CANTINA MED SALSA 300G	8/29/20	12.00	915.19	10,982.28							-12.00	-10,982.28
ACTD02	ANTICA CANTINA MED SALSA 300G	8/29/20	12.00	915.19	10,982.28								

Figure 4.1 Local Vendor Dataset

Local Pharmacy Dataset

Attributes:

1. saleinvcode
2. customerref
3. date
4. invdiscperc
5. flatdisc
6. misccharges
7. invsalestax
8. remarks
9. looseqty
10. packqty
11. price
12. itemdiscperc
13. packunits
14. batch
15. expiry
16. salestax
17. itemname
18. datestring
19. customer
20. rowid

Instances: 300,000

Format: XLS

Time Frame: July 2022 to June 2023

Description: The data, seen in Figure 4.2, needs to be cleaned. saleinvcode, customerref, invdiscperc, flatdisc, misccharges, invsalestax, packqty, itemdiscperc, batch, salestax, datestring customer, and rowid need to be removed as they are either empty, zero, same for every entry, or irrelevant to demand forecasting. The remarks attribute is used to enter the names of customers, therefore, it should be removed to ensure privacy. The date attribute contains the date and the time, it should be split into two parts, date and time. The sales are only recorded in terms of looseqty. The dataset also contains expiry dates which will enable us to analyse the perishable aspect of products. This dataset is uploaded on our GitHub repository.

saleinvcode	CustomerRef	date	invdiscperc	flatdisc	mischarges	invsalestax	remarks	looseqty	packqty	price	itemdiscperc	packunits	batch	expiry	salestax	itemname	datestring	customer	rowid	
268401		4/1/23 9:07	0	0	0	0		1	0	175	0	1		12/12/24 0:00	0	CARE BABY WIPES(80)	14/2023	***CASH SALES CUSTOMER	524334	
268402		4/1/23 9:28	0	0	0	0		1	0	105.2	3	10		10/12/24 0:00	0	ATEM NEBULIZER 10S	14/2023	***CASH SALES CUSTOMER	524335	
268402		4/1/23 9:28	0	0	0	0		1	0	97.6	3	10		12/12/24 0:00	0	CLENILA INJ	14/2023	***CASH SALES CUSTOMER	524336	
268403		4/1/23 9:30	0	0	0	0		1	0	22.06	0	1		12/12/24 0:00	0	OTICOM DYSRHYGINIUM(10'S)	14/2023	***CASH SALES CUSTOMER	524337	
268403		4/1/23 9:30	0	0	0	0		1	0	30	0	100		10/2/24 0:00	0	10CC SHIFA DISYRINGE(UNITS/100)	14/2023	***CASH SALES CUSTOMER	524338	
268404		4/1/23 9:32	0	0	0	0		23	0	10	0	1		2/20/24 0:00	0	FACE MASK 3PLY GREEN RS(5)	14/2023	***CASH SALES CUSTOMER	524339	
268405		4/1/23 9:42	1	0	0	0		4	0	25.33	0	30		12/12/24 0:00	0	DEXXO 30MG CAP 30'S	14/2023	***CASH SALES CUSTOMER	524340	
268406		4/1/23 9:58	0	3	0	0		1	0	17.9	0	10		11/12/24 0:00	0	MARX 40MG CAP 10'S (AXIS PHARMA)	14/2023	***CASH SALES CUSTOMER	524341	
268407		4/1/23 10:54	5	0	0	0		1	0	138.61	0	1		11/20/24 0:00	0	LAXOBENON 120ML LIQ	14/2023	***CASH SALES CUSTOMER	524342	
268407		4/1/23 10:54	5	0	0	0		10	0	4.45	0	100		10/4/24 0:00	0	ULTRAC 100MG TAB	14/2023	***CASH SALES CUSTOMER	524343	
268407		4/1/23 10:54	5	0	0	0		10	0	36	0	10		7/19/24 0:00	0	ULPTOR 10MG TAB(PARKE DAVIS)	14/2023	***CASH SALES CUSTOMER	524344	
268408		4/1/23 10:57	0	0	0	0		1	0	64.52	0	20		9/12/24 0:00	0	ENTERO GERMINA INJ	14/2023	***CASH SALES CUSTOMER	524345	
268409		4/1/23 11:04	0	0	0	0		4	0	22	0	10		4/1/24 0:00	0	PIZOER G 15MG TAB	14/2023	***CASH SALES CUSTOMER	524346	
268409		4/1/23 11:04	0	0	0	0		1	0	22	0	10		12/12/24 0:00	0	PIZOER G 15/20MG TAB	14/2023	***CASH SALES CUSTOMER	524347	
268410		4/1/23 11:20	10	0	0	0		60	0	25.06	0	20		11/12/24 0:00	0	DIAMICRON MR 60MG TAB	14/2023	***CASH SALES CUSTOMER	524348	
268411		4/1/23 11:28	2	0	0	0		1	0	75.43	0	1		9/12/24 0:00	0	DRUG 100MG CREAM	14/2023	***CASH SALES CUSTOMER	524349	
268411		4/1/23 11:28	2	0	0	0		1	0	35	0	1		11/2/24 0:00	0	KYNOSINE 100MG CAPS(15ML ADULT)	14/2023	***CASH SALES CUSTOMER	524350	
268412		4/1/23 11:29	7	0	0	0	0	YOUASF KAMAL	14	0	27.93	0	14		10/1/24 0:00	0	SANTE 40MG CAP	14/2023	***CASH SALES CUSTOMER	524351
268412		4/1/23 11:29	7	0	0	0	0	YOUASF KAMAL	1	0	27.93	0	14		11/12/24 0:00	0	SANTE 40MG CAP	14/2023	***CASH SALES CUSTOMER	524352
268412		4/1/23 11:29	7	0	0	0	0	YOUASF KAMAL	15	0	35	0	10		10/1/24 0:00	0	E TO-OD 150MG TAB	14/2023	***CASH SALES CUSTOMER	524353
268412		4/1/23 11:29	7	0	0	0	0	YOUASF KAMAL	15	0	12.5	0	30		10/12/24 0:00	0	NERVON TAB	14/2023	***CASH SALES CUSTOMER	524354
268412		4/1/23 11:29	7	0	0	0	0	YOUASF KAMAL	1	0	41	0	14		12/1/24 0:00	0	VONOZAN 20MG TAB	14/2023	***CASH SALES CUSTOMER	524355
268412		4/1/23 11:29	7	0	0	0	0	YOUASF KAMAL	2	0	24.05	0	14		10/12/24 0:00	0	VONOZAN 20/40MG TAB	14/2023	***CASH SALES CUSTOMER	524356
268412		4/1/23 11:29	7	0	0	0	0	YOUASF KAMAL	30	0	5.13	0	30	143413	4/1/24 0:00	0	ALP 0.25MG TAB	14/2023	***CASH SALES CUSTOMER	524357
268413		4/1/23 11:34	10	0	0	0	0	YOUASF KAMAL	3	0	74.75	0	4		11/12/24 0:00	0	ICON 100MG CAP(FERZONS)	14/2023	***CASH SALES CUSTOMER	524358
268413		4/1/23 11:34	10	0	0	0	0	YOUASF KAMAL	3	0	49.75	0	20		11/12/24 0:00	0	CUTIS 250MG TAB	14/2023	***CASH SALES CUSTOMER	524359
268414		4/1/23 11:37	0	0	0	0	0	YOUASF KAMAL	3	0	41.07	0	14		1/1/24 0:00	0	GABICA 150MG CAP	14/2023	***CASH SALES CUSTOMER	524360
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	21	0	39.9	0	21		12/1/24 0:00	0	RONICOL 2MG TAB	14/2023	***CASH SALES CUSTOMER	524361
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	26	0	39.93	0	31		12/1/24 0:00	0	DRUG 100MG TAB	14/2023	***CASH SALES CUSTOMER	524362
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	30	0	40.02	0	30		9/1/24 0:00	0	O CO 50MG CAP	14/2023	***CASH SALES CUSTOMER	524363
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	9	0	52	0	20		11/12/24 0:00	0	MAXFLOW 0.4MG CAP	14/2023	***CASH SALES CUSTOMER	524364
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	6	0	52	0	20		12/1/24 0:00	0	MAXFLOW 0.4MG CAP	14/2023	***CASH SALES CUSTOMER	524365
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	30	0	9.75	0	30		5/22/24 0:00	0	VITRUM TAB(SEARLE)	14/2023	***CASH SALES CUSTOMER	524366
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	15	0	19.42	0	20		12/1/24 0:00	0	E LAXINE 15MG TAB (MIRTAZAPINE)	14/2023	***CASH SALES CUSTOMER	524367
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	14	0	2.25	0	14		11/1/24 0:00	0	TENEX 20MG TAB	14/2023	***CASH SALES CUSTOMER	524368
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	10	0	13.37	0	30		12/1/24 0:00	0	SINEMET 100MG TAB	14/2023	***CASH SALES CUSTOMER	524369
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	40	0	13.37	0	30		1/18/25 0:00	0	SINEMET EXTRA 25+100MG TAB	14/2023	***CASH SALES CUSTOMER	524370
268415		4/1/23 11:41	7	0	0	0	0	YOUASF KAMAL	15	0	51.56	0	30		12/12/24 0:00	0	LAMICTAL 50MG TAB	14/2023	***CASH SALES CUSTOMER	524371
268416		4/1/23 11:48	0	0	0	0	0	YOUASF KAMAL	1	0	17.15	10	20		11/1/24 0:00	0	TONOFLEX P TAB (NEW)	14/2023	***CASH SALES CUSTOMER	524372
268416		4/1/23 11:48	0	0	0	0	0	YOUASF KAMAL	10	0	8.12	0	30	CTH441	8/1/25 0:00	0	NEUXAM 0.5 TAB	14/2023	***CASH SALES CUSTOMER	524373
268416		4/1/23 11:48	0	0	0	0	0	YOUASF KAMAL	4	0	16.06	0	30	ctr918	12/1/25 0:00	0	NEUXAM 1MG TAB	14/2023	***CASH SALES CUSTOMER	524374
268416		4/1/23 11:48	0	0	0	0	0	YOUASF KAMAL	2	0	9.35	0	30		10/1/25 0:00	0	NEUXAM 1/2MG TAB NEW	14/2023	***CASH SALES CUSTOMER	524375
268416		4/1/23 11:48	0	0	0	0	0	YOUASF KAMAL	1	0	20	10	20		12/1/25 0:00	0	MOVAX 2MG TAB NEW	14/2023	***CASH SALES CUSTOMER	524376
268416		4/1/23 11:48	0	0	0	0	0	YOUASF KAMAL	1	0	6.96	10	30		9/19/24 0:00	0	ZYRETEC 10MG TAB(30'S)	14/2023	***CASH SALES CUSTOMER	524377
268416		4/1/23 11:48	0	0	0	0	0	YOUASF KAMAL	1	0	32.14	10	21		4/12/25 0:00	0	RISEK 40MG CAP NEW (21'S)	14/2023	***CASH SALES CUSTOMER	524378
268417		4/1/23 11:49	5.5	2	0	0	0	YOUASF KAMAL	14	0	27.5	0	14		12/1/24 0:00	0	CITANEW 10MG TAB	14/2023	***CASH SALES CUSTOMER	524379
268417		4/1/23 11:49	5.5	2	0	0	0	YOUASF KAMAL	14	0	33.93	0	14		11/1/24 0:00	0	DRUG 100MG TAB	14/2023	***CASH SALES CUSTOMER	524380
268418		4/1/23 11:54	5.5	2	0	0	0	YOUASF KAMAL	14	0	8.7	0	20		11/1/24 0:00	0	NISTE TAB	14/2023	***CASH SALES CUSTOMER	524381
268418		4/1/23 11:54	10	0	0	0	0	YOUASF KAMAL	7	0	28.57	0	14		11/12/24 0:00	0	MONTIKA 10MG TAB(SAMI)	14/2023	***CASH SALES CUSTOMER	524382
268418		4/1/23 11:54	10	0	0	0	0	YOUASF KAMAL	7	0	13.75	0	30		12/1/25 0:00	0	LORIN NSA TAB(30'S)	14/2023	***CASH SALES CUSTOMER	524384
268418		4/1/23 11:54	10	0	0	0	0	YOUASF KAMAL	4	0	48.26	0	10		11/12/25 0:00	0	KLARICID 250MG TAB (ABBOTT)	14/2023	***CASH SALES CUSTOMER	524385
268418		4/1/23 11:54	10	0	0	0	0	YOUASF KAMAL	7	0	23.81	0	100		11/12/25 0:00	0	METHYCOBAL TAB	14/2023	***CASH SALES CUSTOMER	524386
268418		4/1/23 11:54	0	0	0	0	0	YOUASF KAMAL	1	0	50	0	12		12/12/24 0:00	0	KNIGHT RIDER CONDOM 3'S	14/2023	***CASH SALES CUSTOMER	524387

Figure 4.2 Local Pharmacy Dataset

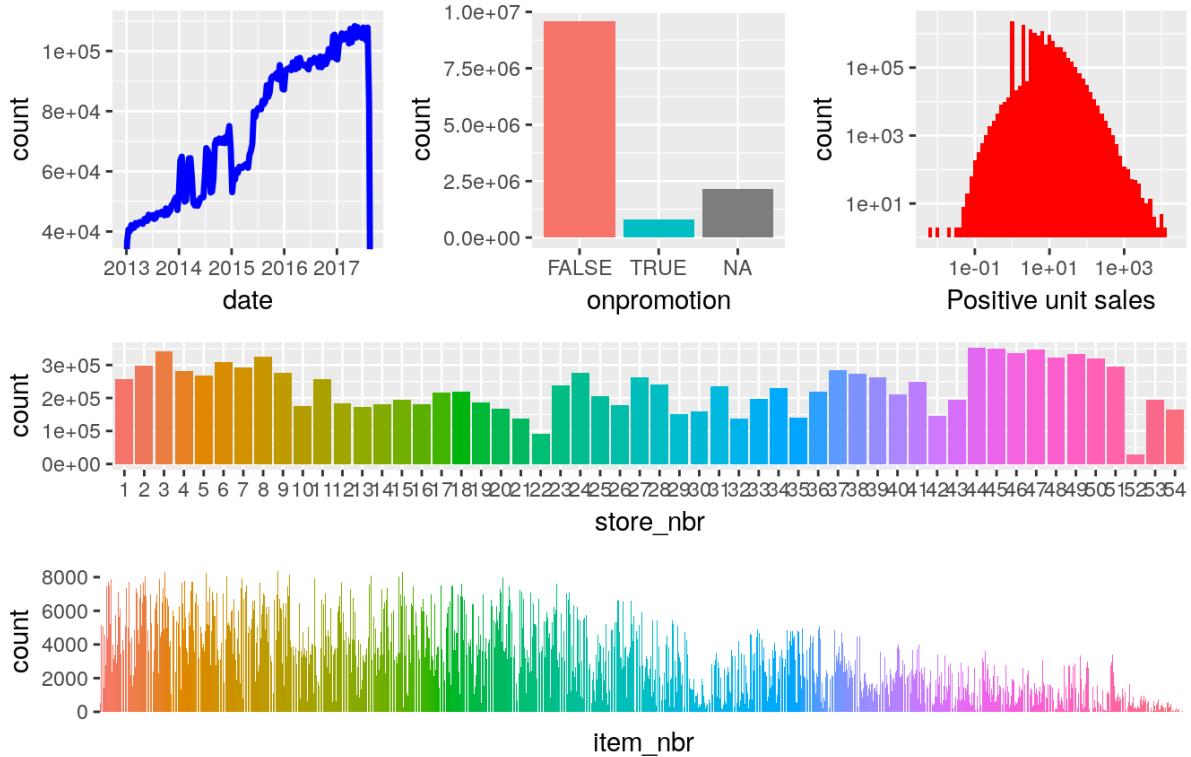
Kaggle

Corporación Favorita Grocery Sales Forecasting

Datasets:

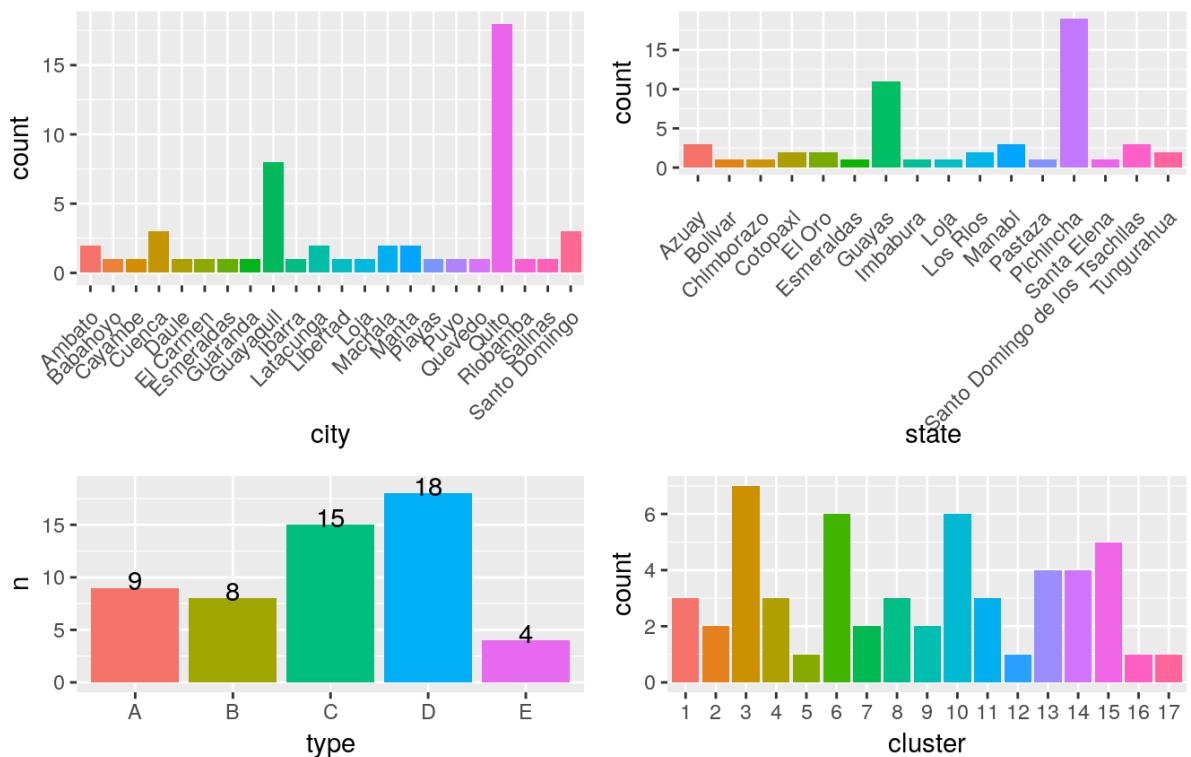
Training Data

1. Date
2. On Promotion
3. Unit Sales
4. Store Number
5. Item Number



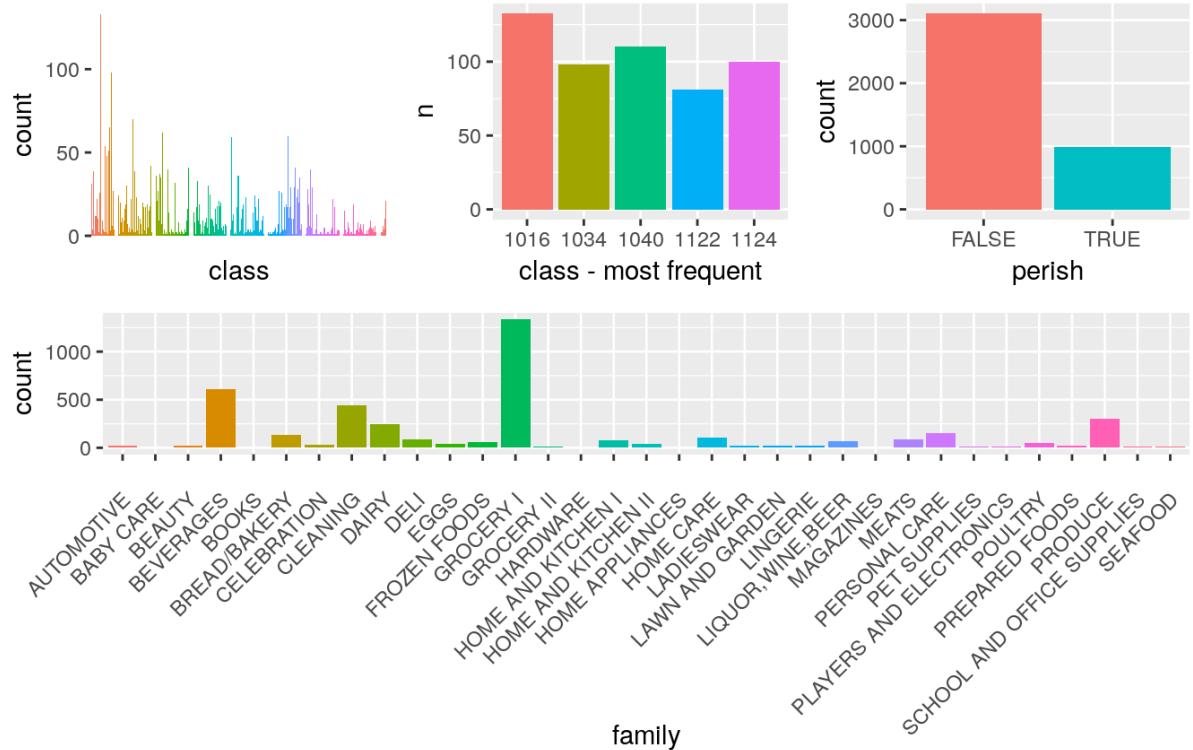
Stores

1. City
2. State
3. Type
4. Cluster



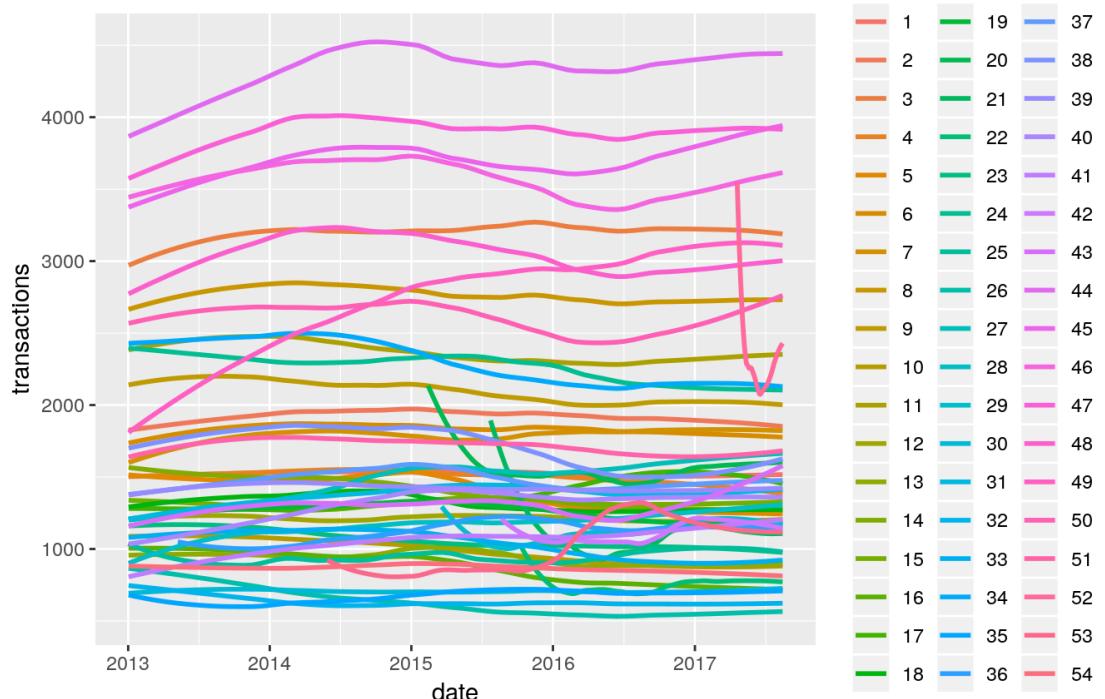
Items

1. Class
2. Perishable
3. Family



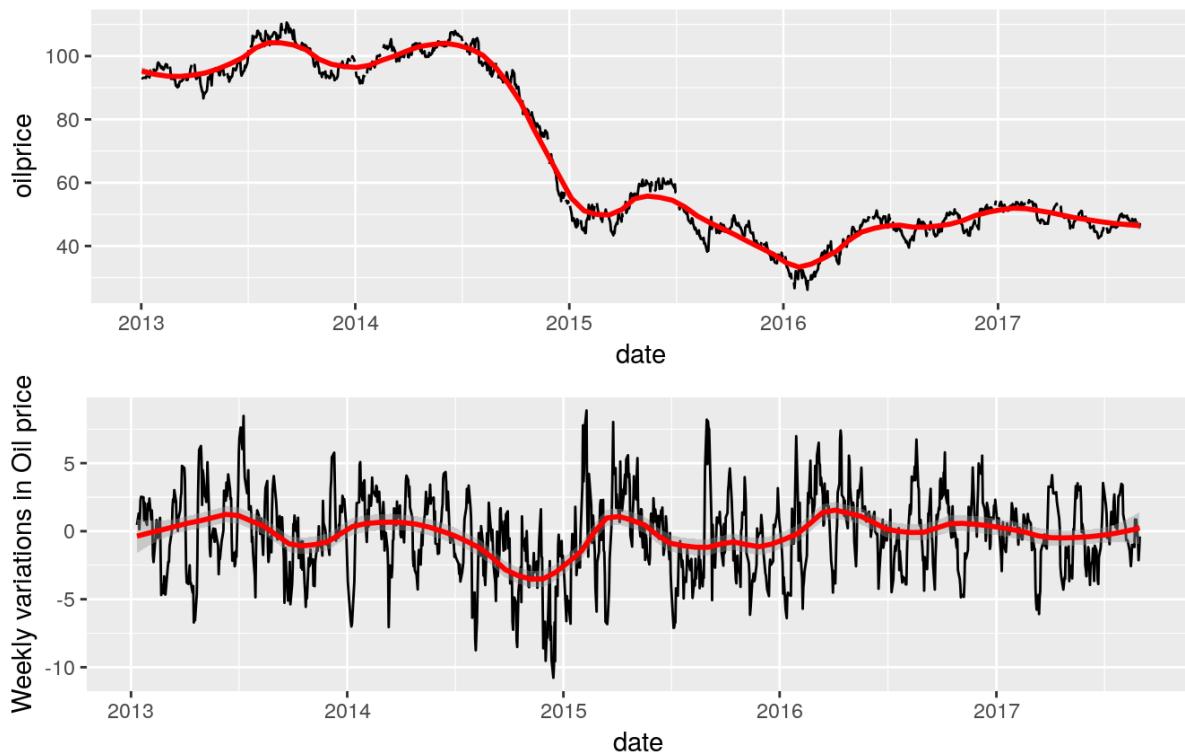
Transactions

1. Date
2. Transactions



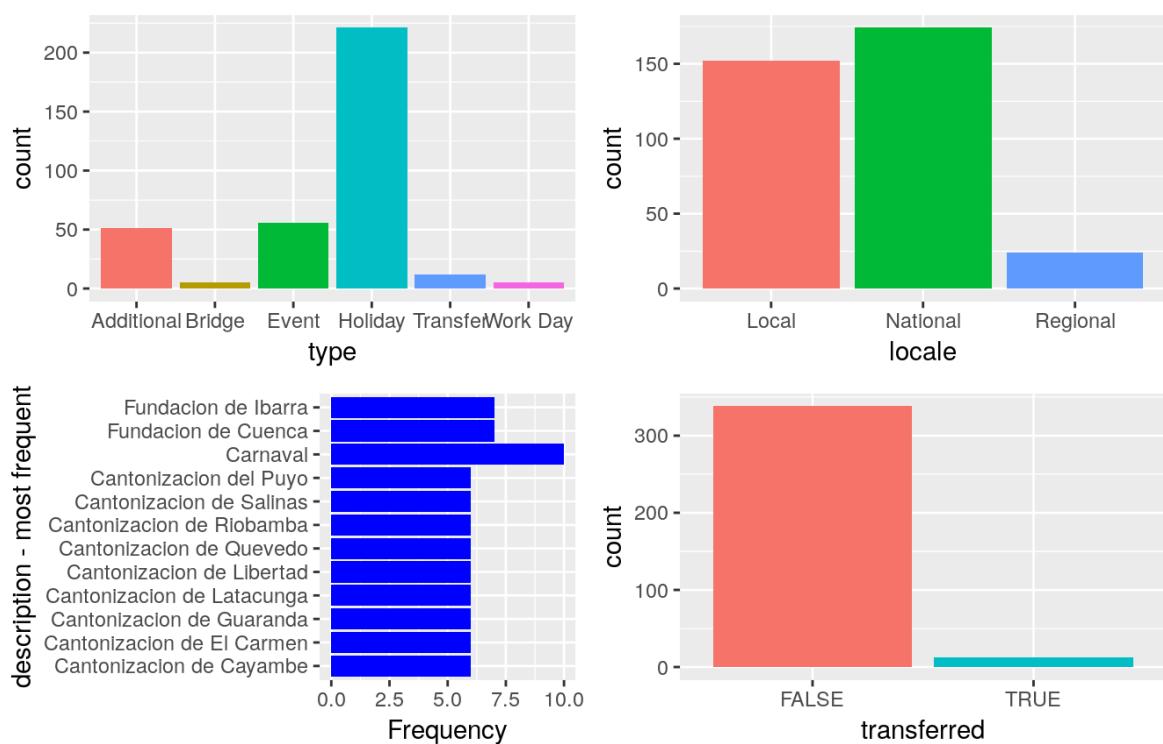
Oil

1. Date
2. Oil Price



Holidays

1. Type
2. Locale



Instances: 126 Million

Size: 4.7 GB (Training Data)

Format: CSV

Time Frame: January 2013 to August 2017

Description: This is a dataset from Corporación Favorita Grocery Sales Forecasting competition hosted on Kaggle 6 years ago. The dataset contains millions of instances, spanning over 5 years, including key variables such as holidays, oil prices, and location. The dataset is considered credible based on the fact it's shared by a large grocery store chain, hosted as a competition on Kaggle with prizes of \$30,000 and 1500+ participants, and most importantly is used for demand forecasting in several research papers including [\[16\]](#) and [\[25\]](#).

Current Work: Even though the Corporación Favorita Grocery Sales Forecasting competition ended on Jan 16, 2018, Kaggle created a new competition [Store Sales - Time Series Forecasting](#) with the same data which runs indefinitely with a rolling leaderboard (we intend to submit predictions on this data of our final ML model in the competition). This has proven to be a very helpful resource as we have been able to access recent works of people using the latest advancements in neural networks for time series analysis.

Source: All the graphs for this dataset are taken from [Shopping for Insights - Favorita EDA](#). (The purpose for including these graphs was that we intend to perform such data analysis on the dataset(s) we use as well.)

Instacart Market Basket Analysis

Datasets:

Aisles

1. aisle_id (1, 2, 3)
2. aisle (prepared soups salads, specialty cheeses, energy granola bars)

Departments

1. department_id (1, 2, 3)
2. department (frozen, other, bakery)

Prior Product Orders

1. order_id (1, 1, 1)
2. product_id (49302, 11109, 10246)
3. add_to_cart_order (1, 2, 3)
4. reordered (1, 1, 0)

Orders

1. order_id (2539329, 2398795, 473747)
2. user_id (1, 1, 1)
3. eval_set (prior, prior, prior)
4. order_number (1, 2, 3)
5. order_dow (2, 3, 3)
6. order_hour_of_day (08, 07, 12)
7. days_since_prior_order (NA, 15.0, 21.0)

Products

1. product_id (1, 2, 3)
2. product_name (Chocolate Sandwich Cookies, All-Seasons Salt, Robust Golden Unsweetened Oolong Tea)
3. aisle_id (61, 104, 94)
4. department_id (19, 13, 7)

Instances: 3.4 Million

Format: CSV

Description: This is a dataset from the Instacart Market Basket Analysis competition hosted on Kaggle 6 years ago. The dataset contains millions of instances, including key variables such as aisle, department, day of week, and hour of day. The dataset is considered credible based on the fact it's shared by a large grocery delivery company, hosted as a competition on Kaggle with prizes of \$25,000 and 2500+ participants.

Other Datasets for Further Exploration

- [Grupo Bimbo Inventory Demand](#)
- [Store Item Demand Forecasting Challenge](#)

Feature Engineering

Date

The date provides a time series of sales data, enabling the model to identify and capture seasonal patterns, trends, and recurring events that influence demand. For instance, it can recognize increased sales during holidays, weekends, or specific times of the year. For perishable products, knowing the date of sales is crucial for managing inventory effectively, ensuring that products are sold before they reach their expiration dates. [11] explains how the variable date can be further divided into three numeric attributes which are the day of the month, the month, and the year to maintain the important weekly, monthly and yearly seasonal information.

Product Name

The product name allows for the categorization of items into specific product types or categories. This categorization is essential for understanding and forecasting demand patterns within different product groups. Different products may exhibit varying levels of demand volatility. By considering the product name, the forecasting model can account for the unique demand characteristics of each item, whether it's a fast-moving consumer good or a slow-moving, seasonal product. The product name is critical for inventory management. It enables the model to forecast demand for each product individually, allowing businesses to optimise stock levels, reduce overstocking or understocking issues, and minimise the risk of waste. Certain products may experience fluctuations in demand based on seasons or trends. The product name allows the model to identify and capture these variations, helping in accurate demand forecasting.

Sales

Sales revenue directly reflects the monetary value of products sold. By analysing historical sales revenue data, the forecasting model can gain insights into the overall financial performance of specific products, categories, or the entire business. Sales revenue data provides a comprehensive view of demand trends and patterns over time. Analysing revenue fluctuations allows the model to identify seasonal variations, product life cycles, and other factors influencing demand. Changes in sales revenue may be linked to pricing strategies. The model can analyse how alterations in product prices impact revenue and, consequently, adjust demand forecasts based on pricing dynamics. [16] has used the sales variable as a numeric value which represents the number of units sold.

Holiday

Holidays often lead to an increase in consumer spending, as people purchase more goods for celebrations, and gatherings. Incorporating holiday data into demand forecasting allows the model to anticipate and accommodate this surge in demand. Consumer preferences for certain products often change during holidays. For

example, there may be increased demand for specific food items like vermicelli, sugar or milk during the eid holidays. Holiday data helps the model identify and predict shifts in product preferences. Holidays can lead to variations in product demand, and businesses need to adjust their inventory levels accordingly. Knowing the timing and significance of holidays allows for better inventory planning to meet increased demand during these periods. [11] mentions that since holidays affect sales they used it as a binary attribute where '0' indicated that it was an ordinary day and '1' indicated that it was a holiday. However, in [16] they have divided the variable 'holiday' into multiple variables including holiday type, holiday locale, holiday locale name, holiday description, and holiday transferred.

Location

Different locations may exhibit variations in consumer preferences, purchasing power, and demand for specific products. Geographical data allows the model to differentiate between regions and tailor demand forecasts accordingly. Geographical data helps identify areas with higher population density, which may experience different demand patterns than sparsely populated regions. This information is valuable for understanding the potential customer base in each location. Economic conditions can vary by location, affecting consumer spending habits, enabling more accurate predictions of demand based on local economic factors.

Weather

Weather patterns are often closely tied to seasons. Understanding how weather changes with the seasons helps the model predict seasonal variations in demand. For example, demand for cold drinks and juices increases during the summers or demand for dry fruits increases during the winters. Rain, snow, or other forms of precipitation can impact consumer mobility and preferences. For example, heavy rainfall may reduce foot traffic at brick-and-mortar stores. Weather data helps the model account for these effects. Weather information including maximum and minimum temperature, and relative humidity were also used while preparing their data and selecting the variables for demand forecasting as explained in [11].

ML Model Selection

Approach Considerations

Ensemble Approach

1. **Diverse Patterns:** When demand patterns vary significantly, an ensemble approach is explored to combine models effectively capturing diverse patterns for robust predictions.
2. **Model Complementarity:** Combining models with complementary strengths enhances the accuracy of the forecasting system.

3. **Increased Robustness:** Ensemble models minimise overfitting risks, ensuring reliable predictions with new data.

Non-Ensemble Approach

1. **Interpretability is Crucial:** For critical interpretability and easy explanation, a non-ensemble approach, such as linear regression, may be considered.
2. **Computational Efficiency:** In cases of limited computational resources or a need for quicker predictions, non-ensemble models offer a more efficient solution.
3. **Homogeneous Patterns:** For relatively homogeneous demand patterns, a single, well-tailored model may suffice without the added complexity of an ensemble.

Type of ML Models

Autoregressive Models

1. **Description:** Capture relationships between observations and lagged data, suitable for time series forecasting.
2. **Example:** Predicting daily sales based on historical sales data.

Exponential Smoothing Models

1. **Models:** SES, Holt's method, Holt-Winters method.
2. **Description:** Address trends and seasonality in time series data through exponential decay of past observations.
3. **Example:** Forecasting monthly revenue, considering regular patterns and long-term trends.

Machine Learning Regression Models

1. **Models:** Linear regression, polynomial regression, decision trees.
2. **Description:** Utilise historical data and relevant features for predicting future demand.
3. **Example:** Predicting weekly sales based on factors like promotions, holidays, and location.

Ensemble Models

1. **Models:** Random Forest Regressor, Gradient Boosting Regressor, XGBoost.
2. **Description:** Combine predictions from multiple models for enhanced accuracy and robustness.

3. **Example:** Integrating Random Forest with Gradient Boosting for a diverse and accurate demand forecast.

Deep Learning Models

1. **Models:** Recurrent Neural Networks (RNNs), LSTM, GRU.
2. **Description:** Capture sequential dependencies for handling complex patterns in time series data.
3. **Example:** Using LSTM to predict daily sales, considering sequential dependencies and temporal nuances.

Competitive Analysis Models

1. **Highlighted Model:** Random Forest Regressor for flexibility, simplicity, and explainability.
2. **Other Models:** Gradient Boosting techniques and LSTM recognized for capturing nuanced demand patterns.

Model Training and Testing

Ensemble Approach: *Training Process*

- **Data Source:** Historical data from various stores.
- **Implementation:** Collaborative contribution of models (Random Forest, Gradient Boosting, LSTM) for collective sales prediction, leveraging unique strengths of each model.

Validation Process

- **Validation Data Set:** 20% of historical data allocated for comprehensive accuracy assessment.
- **Evaluation Metrics:** RMSLE, RMSE, MAPE, MAE provide insights into ensemble performance.

Iterative Improvement

- **Continuous Learning:** Designed for ongoing learning from new data, enhancing collective model knowledge.
- **Model Exploration:** Exploration of new models and techniques ensures a dynamic system adapting to changing demand patterns.

Non-Ensemble Approach: *Training Process*

- **Data Source:** Historical data from major stores.
- **Implementation:** Relies on a single, well-tailored model (e.g., Linear Regression) for sales forecasting, chosen based on suitability for specific demand patterns.

Validation Process

- **Validation Data Set:** 20% of historical data reserved for validation purposes.
- **Evaluation Metrics:** Same metrics (RMSLE, RMSE, MAPE, MAE) used to assess the performance of the selected non-ensemble model.

Iterative Improvement

- **Continuous Learning:** Similar to the ensemble approach, the non-ensemble system continuously learns from new data, refining the single model for improved accuracy.

Various models, including Linear Regression and Random Forest Regression, were traditionally used for short-term demand. However, boosting algorithms like Gradient Boosting Regressor [17], Light Gradient Boosting Machine Regressor [18], XGBoost [19], and Cat Boost Regressor [20] outperform traditional methods, especially when dealing with both numerical and categorical features. Additionally, Long-Short Term Memory (LSTM) models [21], including Bidirectional LSTMs, prove effective for sequential data like time series, making them suitable for long-term demand scenarios due to their memory retention capabilities.

Recently, Recurrent Neural Networks (RNNs), Long-Short Term Memory (LSTM), and Bi-directional Long-Short Term Memory (Bi-LSTM) gained prominence for their capacity to model nonlinear functions and capture long-term time-dependent patterns. Hewamalage et al. [22] conducted an empirical study on RNN forecasting models, revealing their ability to directly model seasonality with uniform patterns. However, deseasonalization is necessary for non-uniform patterns. LSTM, a specialised RNN, handles longer input-output connections, as demonstrated by Xu and Wang's [23] sales forecasting based on univariate time series. Bi-LSTM, examined for multivariate time series data, outperformed statistical methods due to the prevalent nonlinear trends in most time series data [24].

Integration with User Interface

The ML model will generate a recommended amount to order for each product as shown in Figure 5.1 while the grocery store is placing the order. The grocery store will display this recommended amount next to the input field for quantity. The grocery store can either order the recommended amount or enter the amount to order manually.

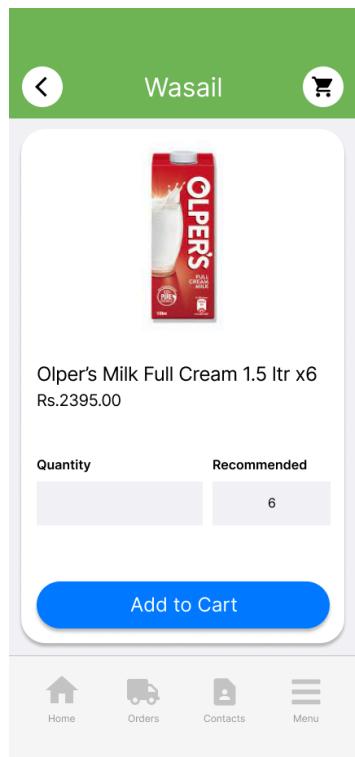


Figure 5.1

Recommendation Updates

To generate recommendations, the ML model will be provided with real-time data - product name, price, date, and store's location from the system and then holiday, exchange rate, and weather through APIs (such as [Open Weather API](#)).

Feedback and Learning

The data accumulated through the inventory management module will be used in the next iteration of the training. An increase in data overtime will increase the overall performance of the model as well. However, even though increasing the data is the most influential factor, it is not the only way to improve performance. The LSTM model (1995) has been around for some time, whereas TFT (2019) and N-HITS (2022) are relatively new models. We shall continue to explore new discoveries in deep learning models, they might be more apt to target our problem.

Performance and Scalability

To ensure the performance and scalability of the ML component, the model will be deployed on Digital Ocean.

Testing and Validation

The ML model will be trained and evaluated on a dataset split into an 80:20 ratio for training and validation, respectively, with cross-validation applied during training.

User Roles

Wasail has the following user roles:

- Grocery Store: A user who is looking for a product for their store.
- Vendor: A user who is selling the products to the grocery store.
- Site Admin: A user who is part of the developer team

User Stories

User Story: Placing the Order

As a grocery store, I would want to place an order for a product while receiving a recommended amount to order that would ensure maximum profit (as seen below in Figure 5.2).

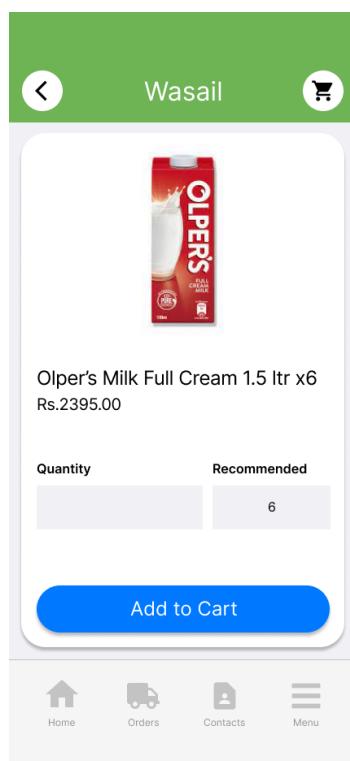


Figure 5.2

Acceptance Criteria:

- Once I have clicked on the vendor's profile, the app should display the vendor's details which should include the contact information, the product listings, and an option to connect (*future improvement: ratings and reviews*).
- While I'm on the vendor's profile, the app displays the product that I have searched for under the "Searched Product" section, and the other products the vendors sell under the "Popular Products" section.
- I can click on the product I want to order, and the app will take me to another screen where I will be able to select the quantity of the product.

- A recommended amount to order would be displayed.
- I should be able to order the recommended amount or enter the amount to order myself.
- Once I have filled out the aforementioned criteria, I would select the “Add to Cart” button and my order would be added to the cart.
- Then, I can open the cart and place the order.

User Story: Searching for a Product

As a grocery store, I would like the ability to look up products online (as seen below in Figure 5.3 and 5.4).

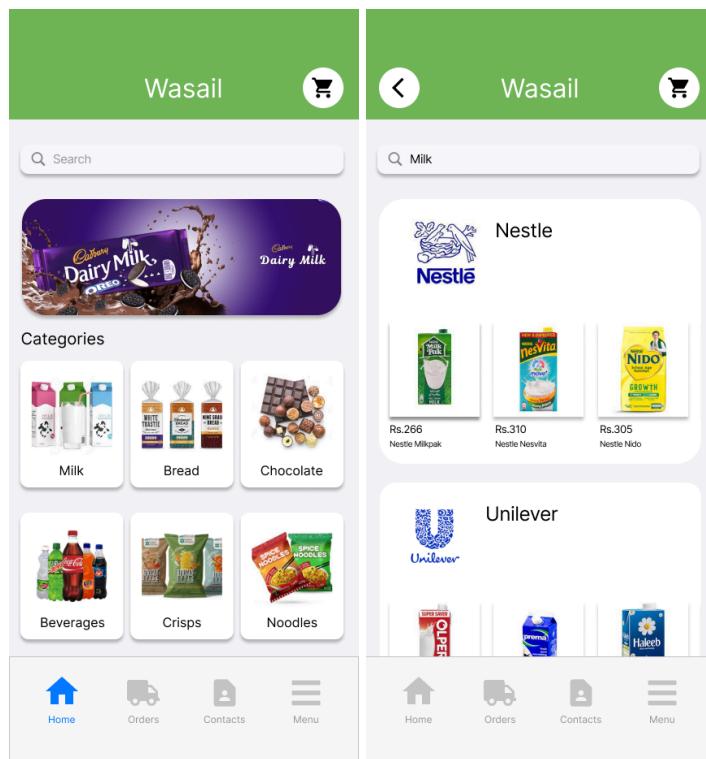


Figure 5.3 & 5.4

Acceptance Criteria:

- When I log into the app, I should see a search bar on the main screen along with popular categories.
- I can search the product by the name of the product (e.g. Milkpak), category of the product (e.g. Milk), and other suggestions as alternatives.
- After entering a search term, the app should present a list of vendors offering the product I need (along with the products), taking into account their ability to deliver to my store's location.
- The vendors on my connection list who fulfil the aforementioned criteria should be displayed at the top, and the rest of the vendors should be displayed after them.
- I can click on the vendor's profile to view more details.

- (Future improvement: Sorting by price low to high, ratings, orders completed, etc)
- (Future improvement: Be able to filter products based on quantity such as a pack, half a pack or individual items, amount of litres of bottles, etc)

Functional Requirements

This section details functional requirements for:

- Grocery Store Mobile App
- Vendor Mobile App
- Admin Web App

Grocery Store and Vendor (FR1)

FR1.1: Language Selection

- **Description:** The system should allow the user to select a language (Fig 5.5).
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user has not specified their preferred language for displaying the text in the app.
- **Postcondition:** The user has specified their preferred language.
- **Details:**
 1. The user is given the option to select a language.
 2. The available options are English, Roman Urdu, and Urdu.
 3. The user's choice for language is saved.

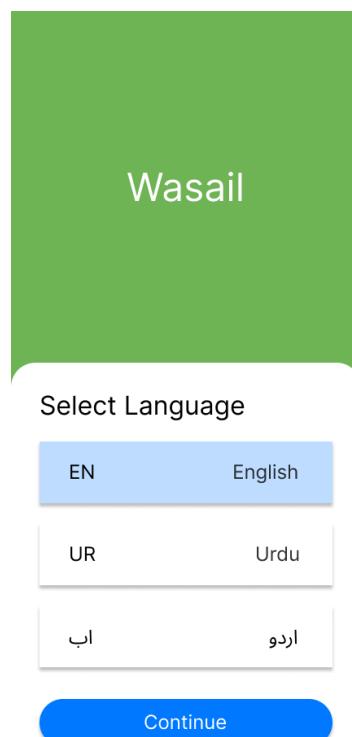


Figure 5.5

FR1.2: Phone Registration

- **Description:** The system should allow users to create an account using their phone number (Figure 5.6).
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user has opened the app.
- **Postcondition:** The user is directed to the phone number confirmation screen.
- **Details:**
 1. User would provide a valid phone number.
 2. The system validates the phone number format ensuring that it has been entered correctly.

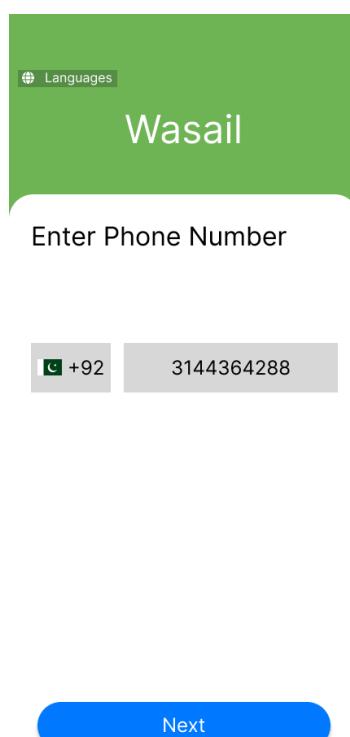


Figure 5.6

FR1.3: Phone Number Confirmation

- **Description:** The system should allow users to confirm if they entered the number correctly (Figure 5.7).
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user has entered their phone number.
- **Postcondition:** The user is directed to the one-time password (OTP) screen.
- **Details:**
 1. The user is asked to confirm if their phone number has been entered correctly.
 2. The system gives the user the option to edit their phone number in case it has not been entered correctly.
 3. Upon confirmation, the system should ask the user to create an account.

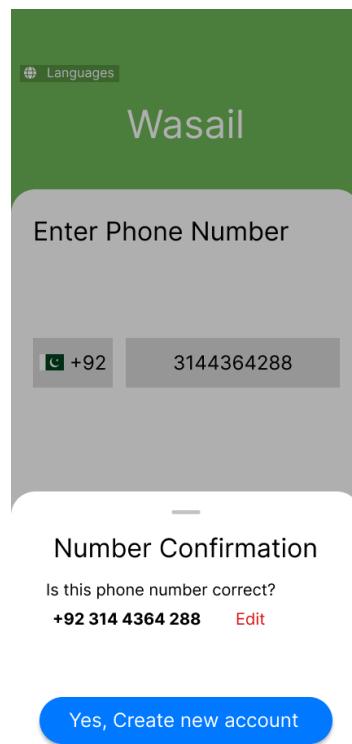


Figure 5.7

FR1.4: Phone Number Exists

- **Description:** The system should check if the phone number exists in the database.
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user has confirmed their phone number.
- **Postcondition:** The user is redirected to either the login or the registration page.
- **Details:**
 1. After phone number confirmation, the system checks if it exists in the database.
 2. If the phone number exists in the database, the user is redirected to the login page.
 3. If the phone number does not exist, that means that it is a new user and the system redirects the user to the registration page.

FR1.5: OTP Code Generation and Delivery

- **Description:** The system should send an OTP code to the user's phone number (Figure 5.8 & 5.9).
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user has confirmed their phone number.
- **Postcondition:** The user has entered the OTP and is directed to the account details page.
- **Details:**
 4. The system generates a 4-digit unique OTP code.
 5. The system sends the OTP code to the user's phone number via SMS.
 6. The OTP code can be resent after 60 seconds and the timer is being shown to the user.
 7. The user is asked to enter the OTP.
 8. The system verifies the validity of the OTP code by comparing it to the generated code sent to the given phone number.
 9. Upon verification, the user shall be directed to the account details page.

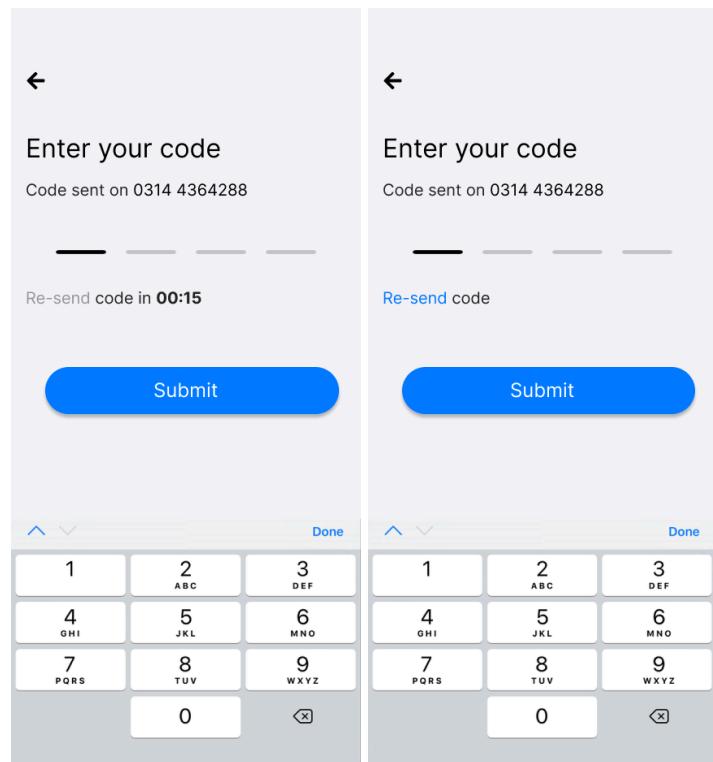


Figure 5.8 & 5.9

FR1.6: Login

- **Description:** The system should allow the registered users to log in using their credentials (Figure 5.10 & 5.11).
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user is not logged in.
- **Postcondition:** The user is logged in.
- **Details:**
 1. The user is asked to enter their phone number.
 2. If the phone number is registered already with the system, the user is asked to enter their password.
 3. The system validates the user's credentials.
 4. Upon successful validation, the user is granted access to their profile.

The figure consists of two screenshots of a mobile application interface. Both screenshots have a green header bar with a globe icon and the word 'Languages'. Below the header is the 'Wasail' logo. The left screenshot shows a text input field labeled 'Enter Phone Number'. The right screenshot shows a text input field labeled 'Phone Number' containing '+923144364288'. Below the phone number input is a password input field labeled 'Password' with the placeholder 'Enter Password'. At the bottom of each screenshot is a blue button labeled 'Next' or 'Login'.

Figure 5.10 & 5.11

FR1.7: Logout

- **Description:** The system should allow the user to logout
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user is logged in.
- **Postcondition:** The user is logged out.
- **Details:**
 1. The system shall give the user the option to log out
 2. The user can log out of the system by using the option

FR1.8: Reset Password

- **Description:** The system should allow the user to reset their password
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user is registered.
- **Postcondition:** The user's password is successfully reset
- **Details:**
 1. The system should give the user the option for resetting their password
 2. In order to reset their password, the system generates a 4 digit unique OTP code.
 3. The system sends the OTP code to the user's phone number via SMS.
 4. Upon verification, the user shall enter their new password.
 5. The user shall re enter their password for confirmation.
 6. After confirmation, the user shall save the password.
 7. The system updates the password in the database.

FR1.9: View Profile

- **Description:** The system should allow the user to view their own profile
- **Actor:** Grocery Store, Vendor
- **Precondition:** The user is on their own profile.
- **Postcondition:** The user is able to view their profile.
- **Details:**
 1. The system displays the user's profile.
 2. The user can view their profile to see their account details.

Grocery Store (FR2)

FR2.9: Account Details

- **Description:** The system should allow the user to enter their account details (Figure 5.12).
- **Actor:** Grocery Store
- **Precondition:** The user has verified their phone number.
- **Postcondition:** The user has registered.
- **Details:**
 1. The user enters their account details including their name, store's name, store's address, password, and location.
 2. User account information is stored in the database.

The image shows a mobile application registration screen titled "Register Now". At the top left is a back arrow icon. The form consists of several input fields: "Mobile Number" with a placeholder "+92 3144364288" and a flag icon; "Password" with a placeholder "Enter Password"; "Confirm Password" with a placeholder "Re-Enter Password"; "Name" with a placeholder "Enter your full name"; "Shop Name" with a placeholder "Enter your shop name"; "Shop Address" with a placeholder "Enter your shop address"; and "Shop Location" with a placeholder "Please enter your shop's current location".

Figure 5.12

FR2.10: Search Product

- **Description:** The system should allow the users to search products based on product name and category (Figure 5.13 & 5.14).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and on the home page.
- **Postcondition:** The system displays the list of matching products along with the vendors that sell them.
- **Details:**
 1. The user can enter the search criteria such as product name or category.
 2. The system retrieves and displays the list of matching products along with the vendors that sell them.
 3. If no matches are found, the system provides appropriate feedback.

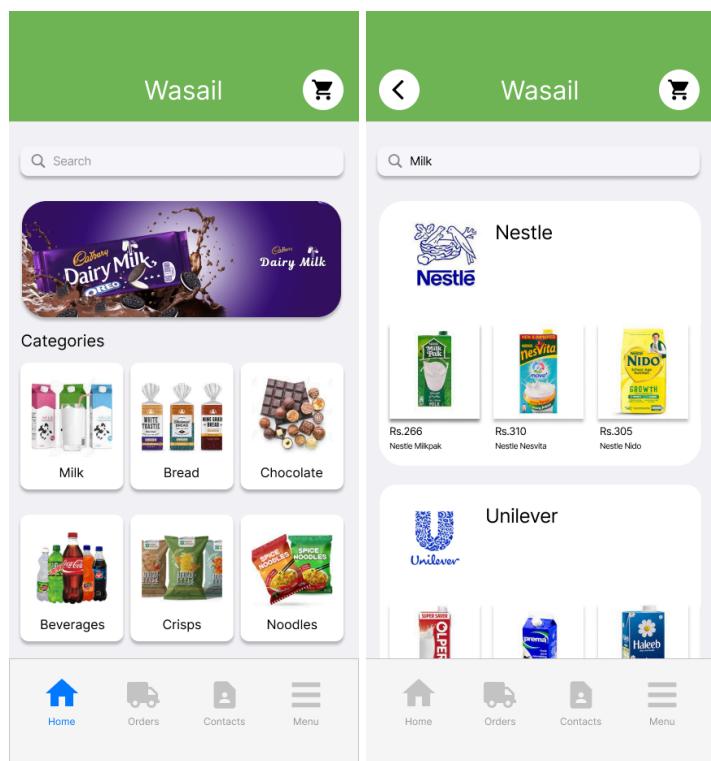


Figure 5.13 & 5.14

FR2.11: Search Vendor

- **Description:** The system should allow the users to search vendors based on their name (Figure 5.15).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and on the home page.
- **Postcondition:** The system displays a list of vendors that deliver in their area based on the search.
- **Details:**
 1. The user can enter the search criteria such as vendor name.
 2. The system retrieves and displays the vendor along with its popular products.
 3. If no matches are found, the system provides appropriate feedback.

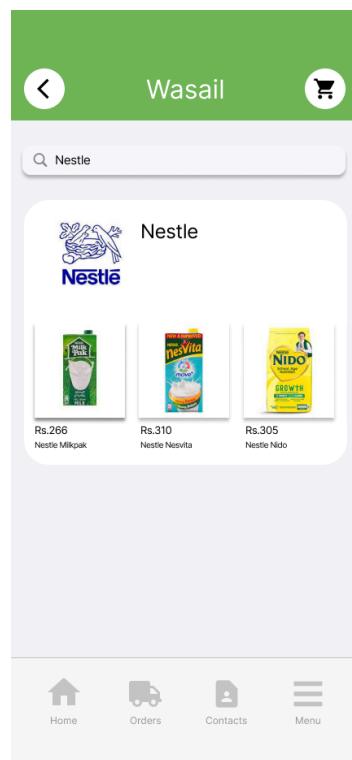


Figure 5.15

FR2.12: Browse Category

- **Description:** The system should allow the user to select a category from the home page directly (Figure 5.16 & 5.17).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and on the home page.
- **Postcondition:** The system displays a list of products based on category selection.
- **Details:**
 1. The user can select the categories that are being displayed to them.
 2. The system retrieves and displays products that fall under the category along with the vendors who sell them.
 3. If no match is found, the system provides appropriate feedback.

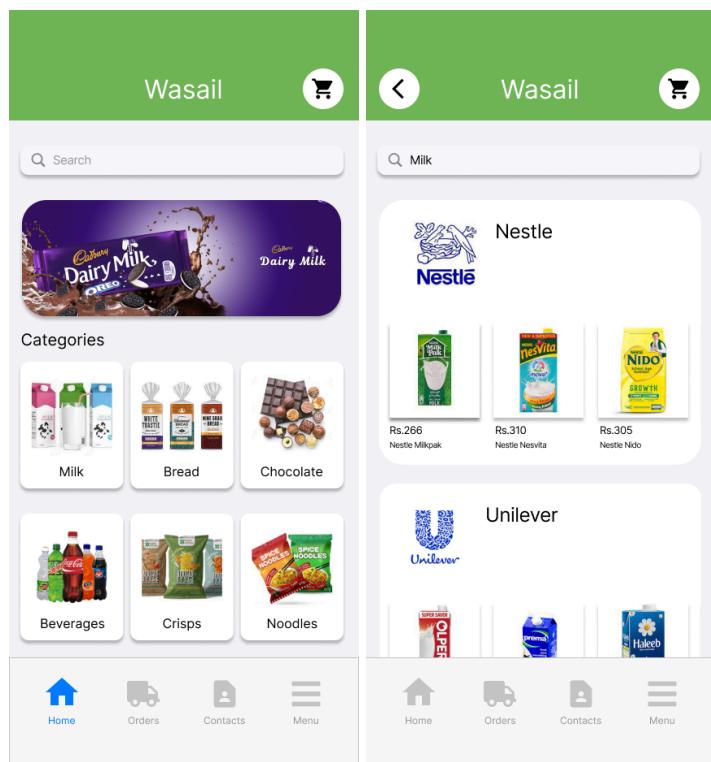


Figure 5.16 & 5.17

FR2.13: View Vendor Profile

- **Description:** The system should allow the user to view the vendor's profile (Figure 5.18).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and has searched the vendor or is on the vendor list page.
- **Postcondition:** The system displays the vendor's profile.
- **Details:**
 1. The user can select the vendor's profile in order to view it.
 2. The system retrieves the vendor's information including their profile picture, name, product listing, and displays it for the user.

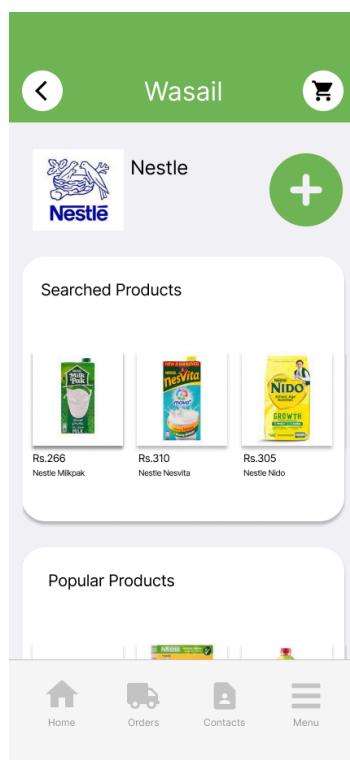


Figure 5.18

FR2.14: Add Vendor to Vendor List

- **Description:** The system should allow the user to add the vendor to their vendor list (Figure 5.19).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and on the vendor's profile.
- **Postcondition:** The vendor is added to the vendor list.
- **Details:**
 1. The user can select the option to add the vendor.
 2. The vendor is added to the vendors list of the user.

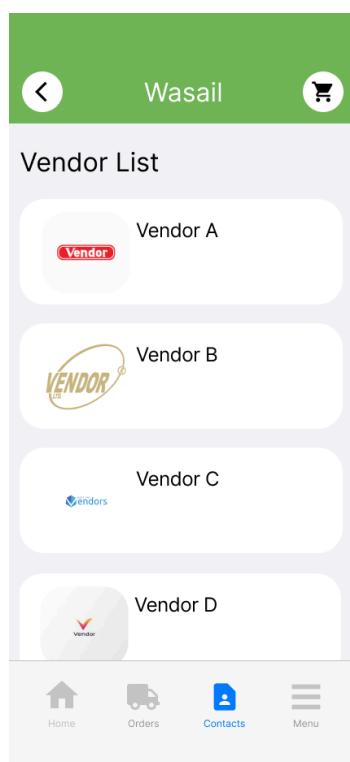


Figure 5.19

FR2.15: Contact Vendor

- **Description:** The system should allow the user to contact the vendor (Figure 5.20).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and on the vendor's profile.
- **Postcondition:** The user has contacted the vendor.
- **Details:**
 1. The system displays the vendor's phone number on the profile.
 2. The user can contact the vendor via message or phone call.

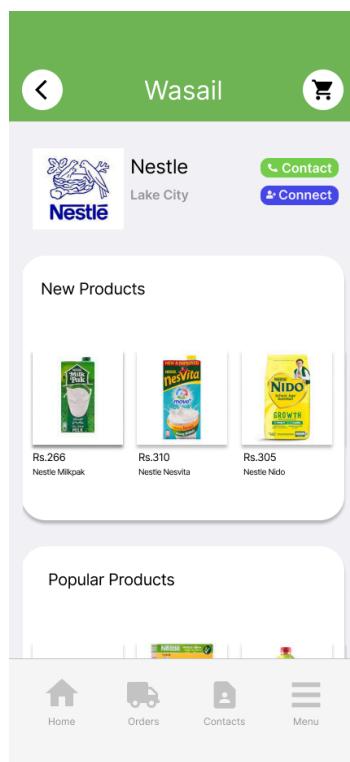


Figure 5.20

FR2.16: View Products on the Vendor's Profile

- **Description:** The system should allow the user to view all the products that the vendors sell on their profile (Figure 5.21).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and on the vendor's profile.
- **Postcondition:** The user has viewed all the products.
- **Details:**
 1. The system retrieves all the products that the vendor sells and displays it to the user.
 2. The user can view the products and scroll through them.

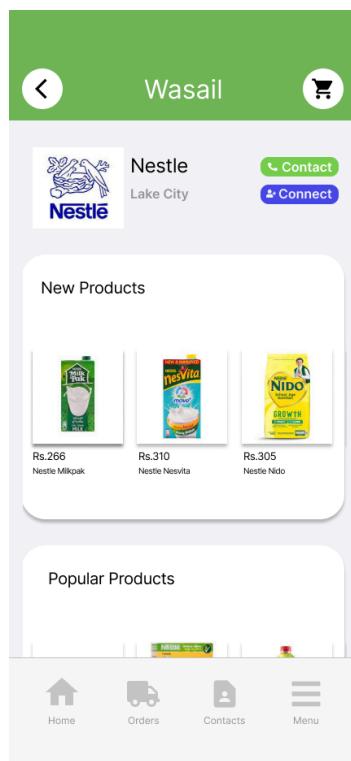


Figure 5.21

FR2.17: View Searched Product

- **Description:** The system should allow the user to view the searched product on the vendor's profile (Figure 5.22).
- **Actor:** Grocery Store
- **Precondition:** The user has searched the product and is on the vendor's profile.
- **Postcondition:** The user has viewed the searched product on the vendor's profile.
- **Details:**
 1. The system displays the searched product on the vendor's profile.
 2. The user can view the products along with their prices and scroll through them.

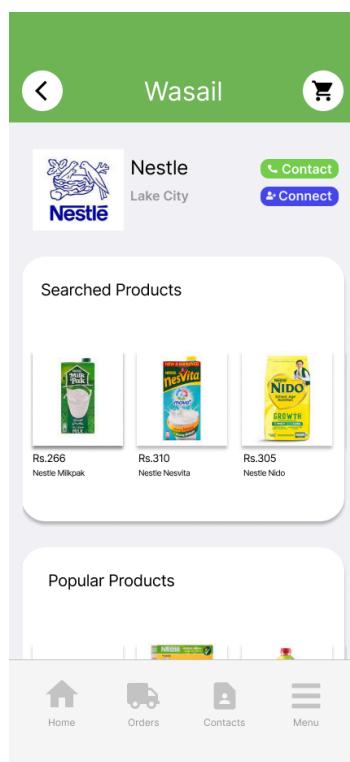


Figure 5.22

FR2.18: Select Products

- **Description:** The system should allow the user to select the product that they want to order (Figure 5.23).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and on the vendor's profile.
- **Postcondition:** The user is directed to the order placement page.
- **Details:**
 1. The system should display products on the vendor's profile.
 2. The user can select from the product list that is being displayed so they can order it.
 3. The user can proceed to place the order.

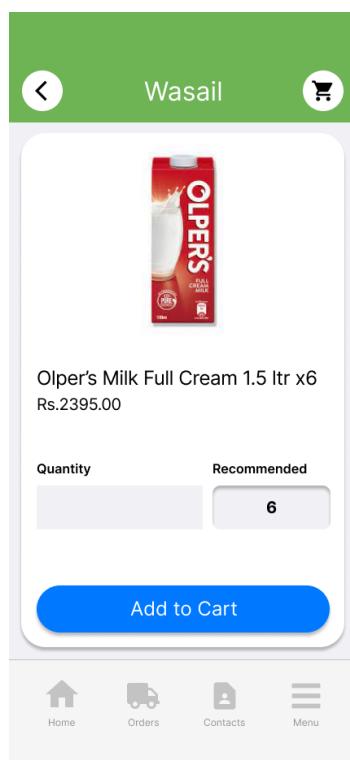


Figure 5.23

FR2.19: Order Recommendation

Description: The system should allow the user to view the recommended amount to order that would ensure maximum profit (Figure 5.24).

- **Actor:** Grocery Store
- **Precondition:** The user is logged in and has selected the product.
- **Postcondition:** The user is given a recommendation for the amount of product to order.
- **Details:**
 1. After product selection, the user is recommended an amount to order.
 2. To generate recommendations, the ML model will be provided with real-time data - product name, price, date, and store's location from the system and then holiday, exchange rate, and weather through APIs (such as [Open Weather API](#)).

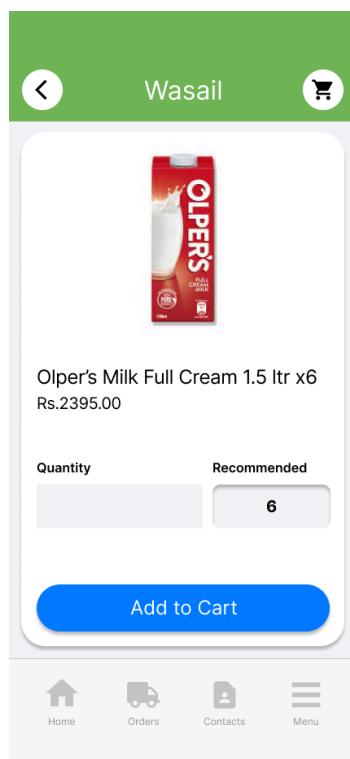


Figure 5.24

FR2.20: Quantity Selection

- **Description:** The system should allow the user to add their own amount to order (Figure 5.25).
- **Actor:** Grocery Store
- **Precondition:** The user is logged in and has selected the product.
- **Postcondition:** The user has entered the quantity of the product.
- **Details:**
 1. The user is given the option to enter the amount of product they want to order.
 2. The user enters the amount of product.
 3. The user adds the product to the cart.

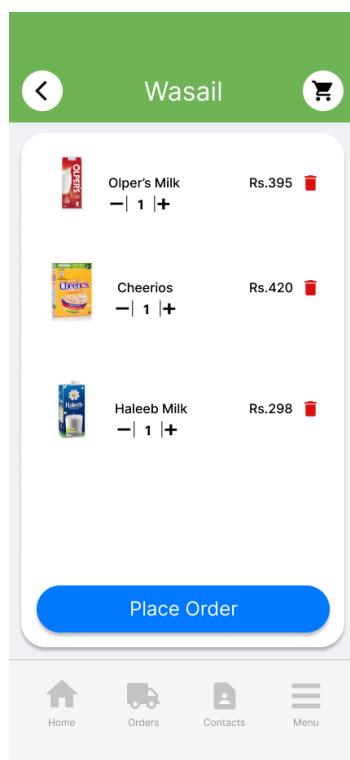


Figure 5.25

FR2.21: Remove Product

- **Description:** The system should allow the user to remove the product (Figure 5.26).
- **Actor:** Grocery Store
- **Precondition:** The product is added to the cart.
- **Postcondition:** The product is removed from the cart.
- **Details:**
 1. The system gives the option to the user to remove the product from the cart.
 2. The user removes the product they do not want to order from the cart.
 3. The system stops displaying that product in their cart.

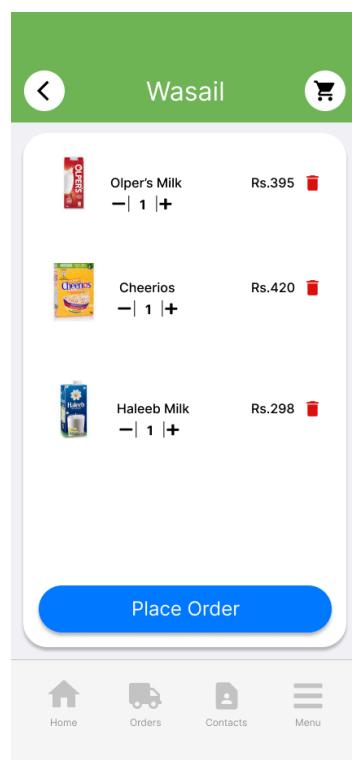


Figure 5.26

FR2.22: Order Placement

- **Description:** The system should allow the user to place the order (Figure 5.27).
- **Actor:** Grocery Store
- **Precondition:** The product is added to the cart.
- **Postcondition:** The order is placed.
- **Details:**
 1. After the user has added the products to their cart, the system gives the option to place the order.
 2. The order is placed to the vendor.
 3. The vendor is notified.
 4. The order is stored in the database.

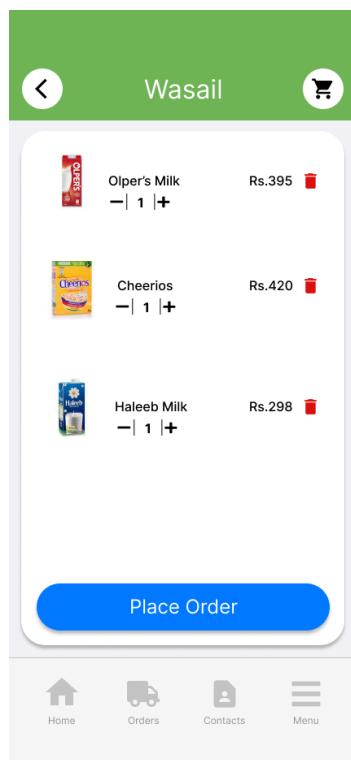


Figure 5.27

FR2.23: View Order

- **Description:** The system should allow the users to view the current orders that they have placed (Figure 5.28).
- **Actor:** Grocery Store
- **Precondition:** The user is on the orders page.
- **Postcondition:** The user has viewed the orders that they have placed.
- **Details:**
 1. The system should display the orders that the user has placed.
 2. The user can view their orders.

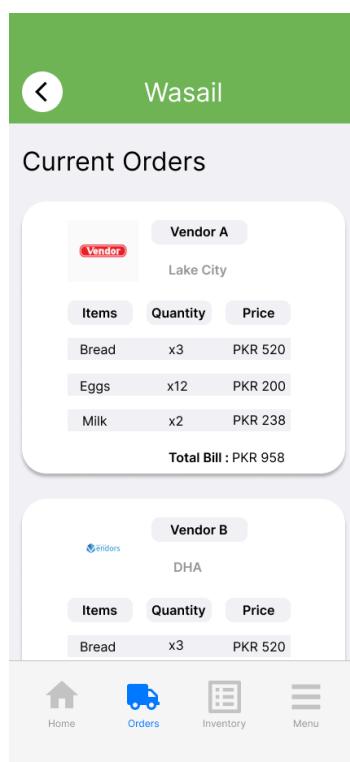


Figure 5.28

FR2.24: Order Tracking

- **Description:** The system should allow the user to track the order (Figure 5.29, 5.30 & 5.31).
- **Actor:** Grocery Store
- **Precondition:** The user is on the orders page.
- **Postcondition:** The user is able to track orders.
- **Details:**
 1. The system displays all the orders that are placed by the user.
 2. The user can select the order which they want to track.
 3. Upon order selection, the user can see which process the order is in. The three options are in process, on its way, and delivered.
 4. The system notifies the user when the order is delivered
 5. The order is marked as delivered in the system

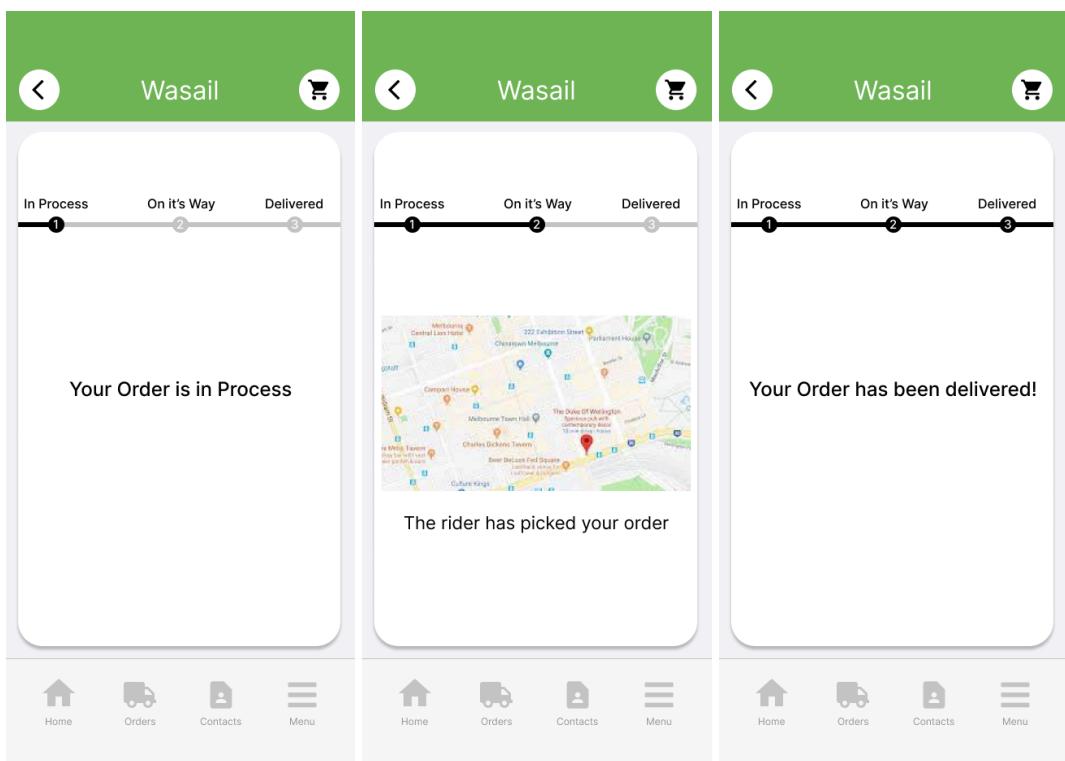


Figure 5.29, 5.30 & 5.31

FR2.25: View Vendor List

- **Description:** The system should allow the user to view the vendors in the vendor list (Figure 5.32).
- **Actor:** Grocery Store
- **Precondition:** The user is on the vendor page.
- **Postcondition:** The user is able to view the vendors that they have added.
- **Details:**
 1. The system displays all the vendors that the user has added.
 2. The user can view the vendors in the vendors list.

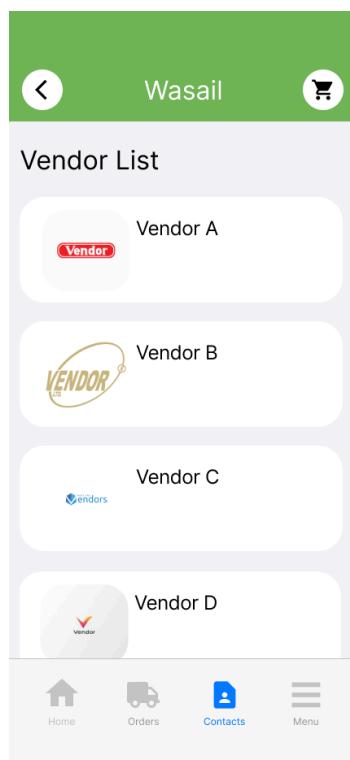


Figure 5.32

FR2.26: View Order History

- **Description:** The system should allow the user to view their previous orders.
- **Actor:** Grocery Store
- **Precondition:** The user is on the orders history page.
- **Postcondition:** The user is able to view their previous orders.
- **Details:**
 1. The system displays all the previous orders the user has placed.
 2. The user can view all their previous orders.

FR2.27: Edit Profile

- **Description:** The system should allow the user to edit their profile details.
- **Actor:** Grocery Store
- **Precondition:** The user is on their own profile.
- **Postcondition:** The user's profile is edited.
- **Details:**
 1. The system displays the profile to the user.
 2. The user can edit the account details that include name, store name, address.
 3. The system updates the information in the database.

FR2.28: Notification

- **Description:** The system should provide notification for important events
- **Actor:** Grocery Store
- **Precondition:** The user is logged in.
- **Postcondition:** The user receives notifications for relevant events
- **Details:**
 1. User receives notification for order being processed, when the order is on its way to be delivered and when the order has been delivered

Vendor (FR3)

FR3.9: Vendor Registration

- **Description:** The system should allow the user to enter their account details
- **Actor:** Vendor
- **Precondition:** The user has verified their phone number.
- **Postcondition:** The user has registered.
- **Details:**
 1. The user enters their account details including their password, full name, username, and delivery areas.
 2. User account information is stored in the database.

FR3.10: Valid Password

- **Description:** The system should allow the user to enter password
- **Actor:** Vendor
- **Precondition:** The user is on the registration page.
- **Postcondition:** The user has entered a valid password.
- **Details:**
 1. Once the user has entered their password, the system checks whether it is valid (against the criteria) or not.
 2. The criteria for checking the password is that the password's length should be a minimum of eight characters, it should have a special character.
 3. If the password meets the criteria, the system indicates this to the user.
 4. If the password does not meet the criteria, the system prompts the user to enter the password according to the criteria.

FR3.11: Username Exists

- **Description:** The system should check if the username exists in the database.
- **Actor:** Vendor
- **Precondition:** The user is on the registration page.
- **Postcondition:** The user has entered a username.
- **Details:**
 1. Once the user has entered their username, the system checks if it exists in the database.
 2. If the username already exists in the database (belongs to another user), the system asks the user to enter a different username.
 3. If the username does not exist, the system allows the user to use that username.

FR3.12: Search Product in Inventory

- **Description:** The system should allow the user to search for a product from their inventory (Figure 5.33 & 5.34).
- **Actor:** Vendor
- **Precondition:** The user is logged in and is on the home page.
- **Postcondition:** The system retrieves the matching product's page.
- **Details:**
 1. The user can enter the search criteria namely product name.
 2. The system retrieves the matching product's page.
 3. If no matches are found, the system provides appropriate feedback.

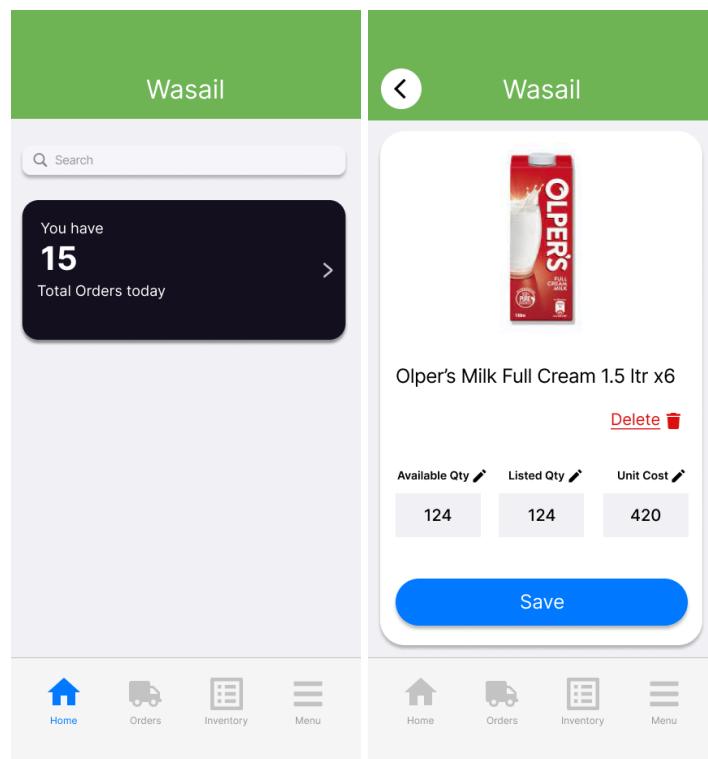


Figure 5.33 & 5.34

FR3.13: Add Product to Inventory

- **Description:** The system should allow the user to add a new product to their inventory (Figure 5.35).
- **Actor:** Vendor
- **Precondition:** The user is logged in and on the inventory page.
- **Postcondition:** The system has directed the user to the all products search page.
- **Detail**
 1. The user wants to add a new product to their inventory.
 2. The system gives the user an option to add a new product.
 3. The user selects the option to add a new product to the inventory.

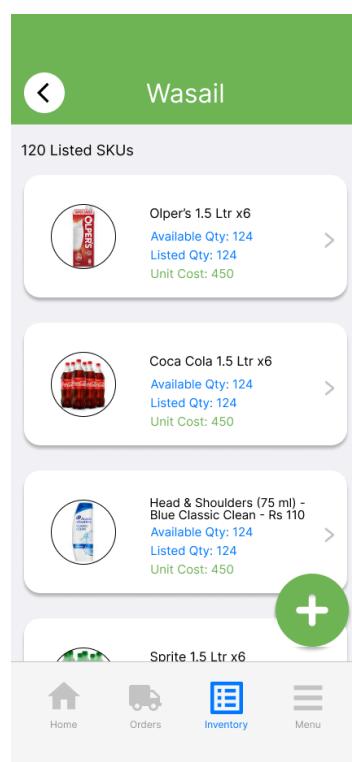


Figure 5.35

FR3.14: All Products Search

- **Description:** The system should allow the user to search all products from the system's product listing (Figure 5.36, 5.37 & 5.38).
- **Actor:** Vendor
- **Precondition:** The user is on the all products search page.
- **Postcondition:** The product is added to their inventory.
- **Details:**
 1. The user searches for an item from among all products.
 2. The system is prompted to generate a list of item suggestions.
 3. The user selects an item from the suggestions.
 4. The item is added to the inventory list.
 5. The user is redirected to update details of the selected item namely its listed quantity, available quantity, and unit cost.

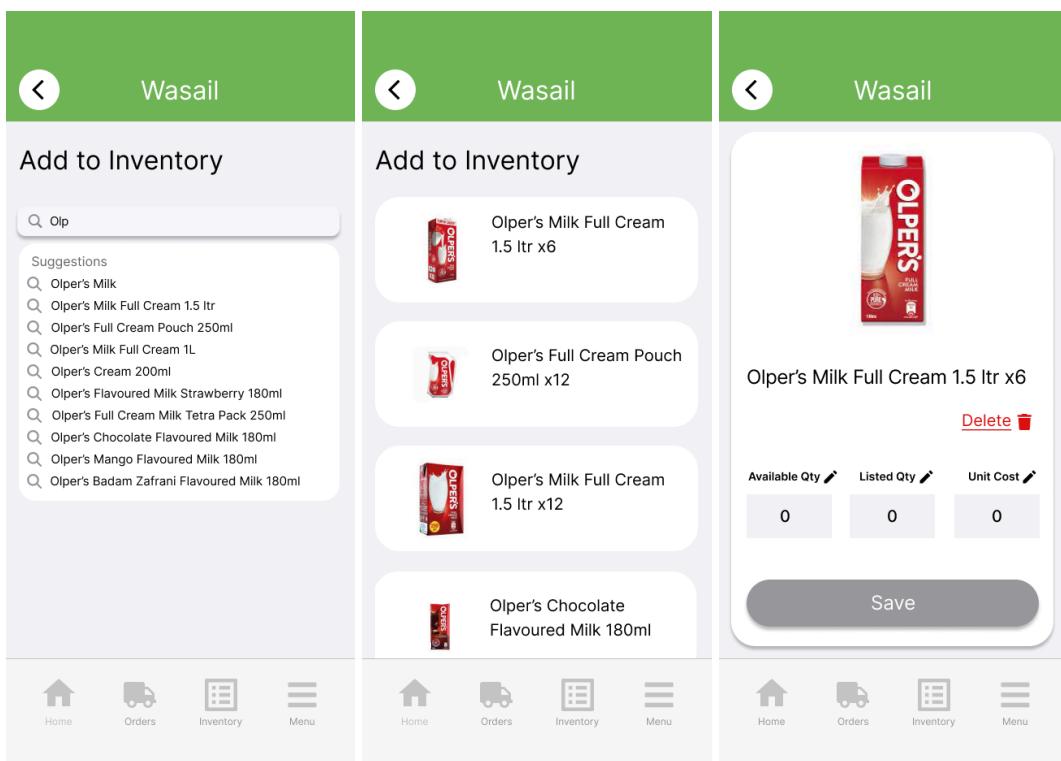


Figure 5.36, 5.37 & 5.38

FR3.15: Remove Product

- **Description:** The system should allow the user to remove a product from the inventory (Figure 5.39).
- **Actor:** Vendor
- **Precondition:** The user is on the product's page.
- **Postcondition:** The product has been removed from their inventory.
- **Details:**
 1. The system gives the option to the user to remove the product from the inventory.
 2. The user removes the product they do not want to keep in the inventory.
 3. The system stops displaying that product in the inventory.

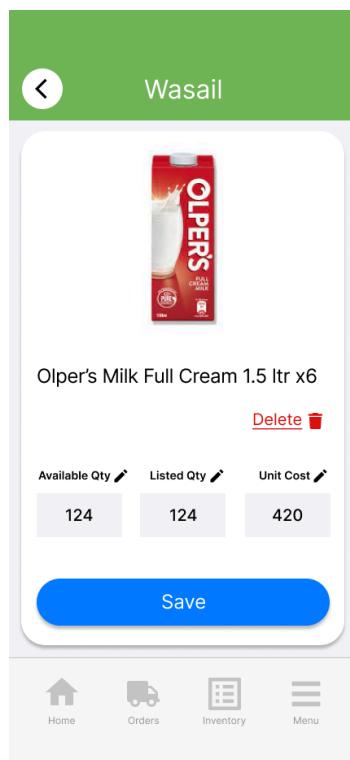


Figure 5.39

FR3.16: Edit Details of Product

- **Description:** The system should allow the user to edit the details of products already in their inventory (Figure 5.40).
- **Actor:** Vendor
- **Precondition:** The user has added the product to their inventory
- **Postcondition:** The details of the product have been edited.
- **Details:**
 1. The system displays the product page for the user.
 2. The system gives the user the option to edit the product details.
 3. The user edits the product details namely its listed quantity, available quantity, and unit cost.
 4. The system will save the edited product details.

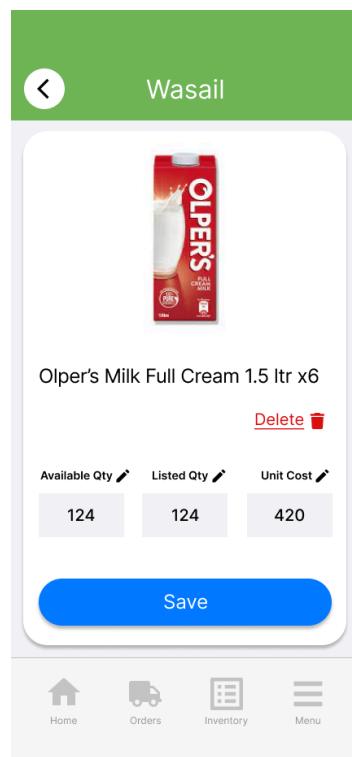


Figure 5.40

FR3.17: View Inventory

- **Description:** The system should allow the user to view their inventory i.e. product listings (Figure 5.41)
- **Actor:** Vendor
- **Precondition:** The user is logged in and on the inventory page.
- **Postcondition:** The user is able to view the inventory.
- **Details:**
 1. The system should display the inventory of the user.
 2. The user can view their inventory.

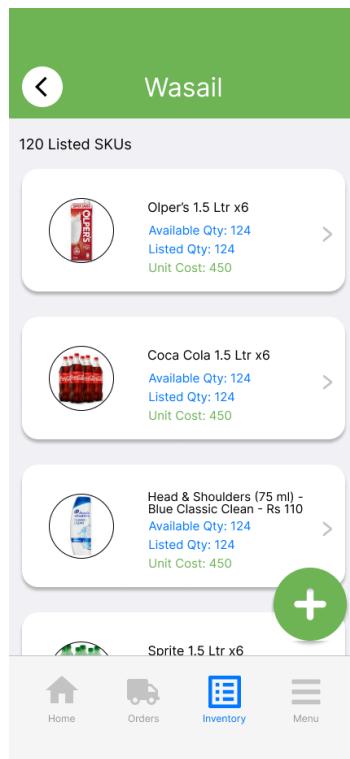


Figure 5.41

FR3.18: View Current Orders

- **Description:** The system should allow the user to view the total number of current orders from all grocery stores (Figure 5.42).
- **Actor:** Vendor
- **Precondition:** The user is logged in and on the orders page.
- **Postcondition:** The user can view all the orders received.
- **Details:**
 1. The system displays all the orders that the user has received and has to deliver.
 2. The user can view all the orders.

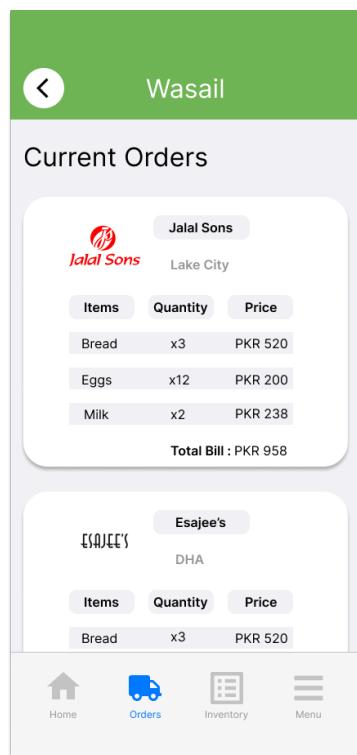


Figure 5.42

FR3.19: View Grocery Stores List

- **Description:** The system should allow the user to view the grocery store list (Figure 5.43).
- **Actor:** Vendor
- **Precondition:** The user is logged in and on the store list page.
- **Postcondition:** The user is able to view the list of grocery stores that placed orders.
- **Details:**
 1. The system displays all the grocery stores that have placed orders.
 2. The user can view the grocery stores in the stores list.

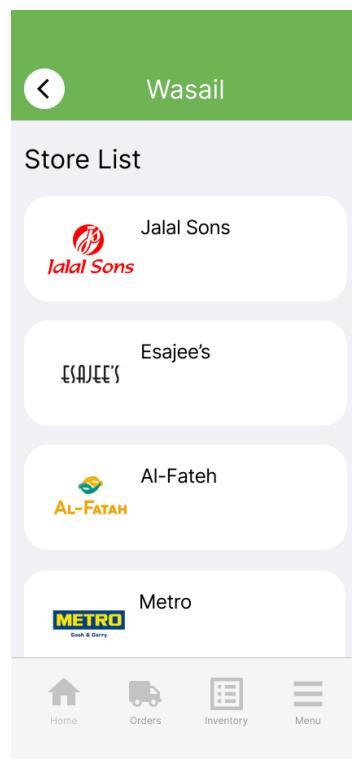


Figure 5.43

FR3.20: View Grocery Store Profile

- **Description:** The system should allow the user to view the grocery store profile (Figure 5.44).
- **Actor:** Vendor
- **Precondition:** The user is on the store list page
- **Postcondition:** The system displays the grocery store's profile.
- **Details:**
 1. The user can select the grocery store's profile to view it.
 2. The system retrieves the grocery store's information including their profile picture, name, and area in which they are, and displays it for the user.

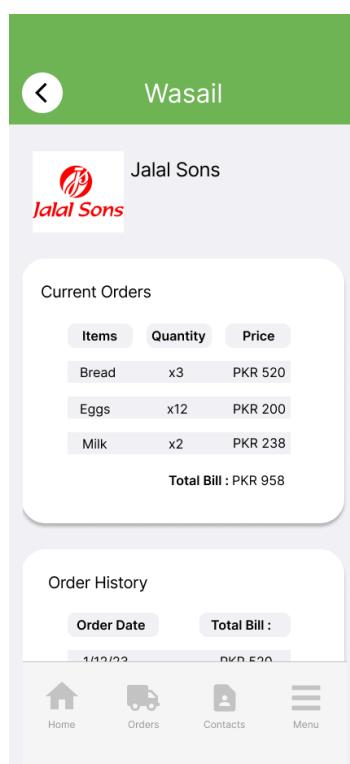


Figure 5.44

FR3.21: View Grocery Store Current Order

- **Description:** The system should allow the user to view the current order placed by the grocery store (profile), (Figure 5.45).
- **Actor:** Vendor
- **Precondition:** The user is on the grocery store's profile page.
- **Postcondition:** The user has viewed the current order that they have received from the grocery store.
- **Details:**
 1. The system should display the current orders that the user has received from the grocery store.
 2. The user can view their current orders received.



Figure 5.45

FR3.22: View Orders History

- **Description:** The system should allow the user to view the orders they have already completed delivering (Figure 5.46).
- **Actor:** Vendor
- **Precondition:** The user is on the orders history page.
- **Postcondition:** The user is able to view the delivered orders.
- **Details:**
 1. The system displays all the orders the user has delivered.
 2. The user can view all their delivered orders.

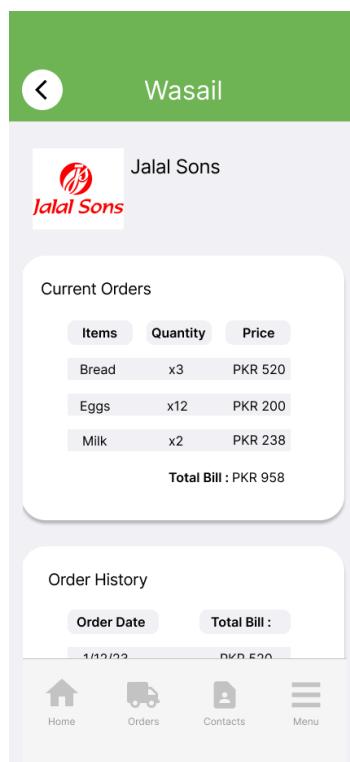


Figure 5.46

FR3.23: Order Dispatch Tracking

- **Description:** The system should allow the user to track the dispatched order (Fig 5.47 & 5.48).
- **Actor:** Vendor
- **Precondition:** The user is on the orders page.
- **Postcondition:** The user is able to track order dispatches.
- **Details:**
 1. The system displays all the orders that are dispatched by the user.
 2. The user can select the order in which they want to track dispatch progress.
 3. Upon order selection, the user can update which process the order is in. The three options are in process, on their way, and delivered.
 4. The user updates the system on the progress of order delivery.
 5. The user successfully completes the delivery and updates the system.
 6. The system notifies of completed delivery.

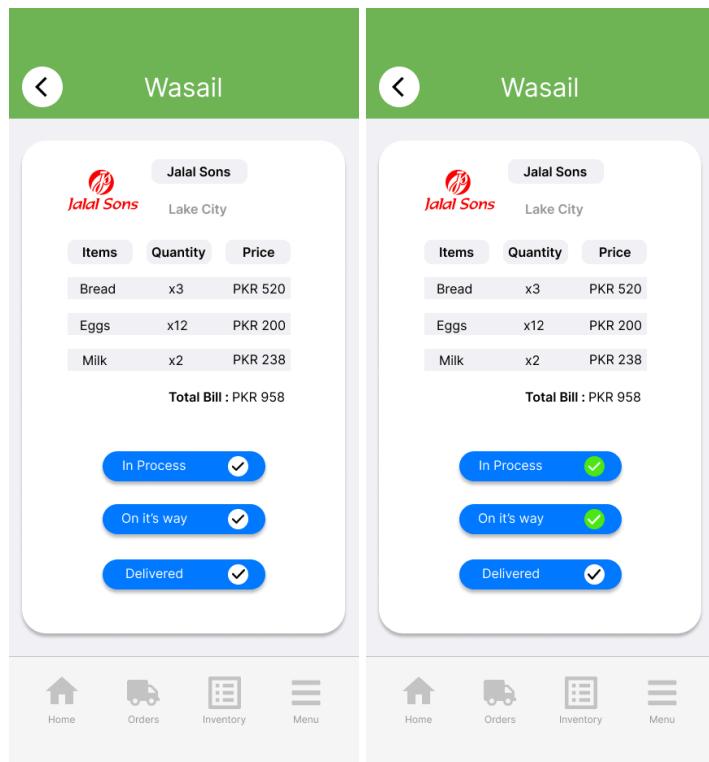


Figure 5.47 & 5.48

FR3.24: Edit Profile

- **Description:** The system should allow the user to edit their profile details
- **Actor:** Vendor
- **Precondition:** The user is on their own profile.
- **Postcondition:** The user's profile is edited.
- **Details:**
 1. The system displays the profile to the user
 2. The user can edit the account details including name, store name, and address.
 3. The system updates the information in the database

FR3.25: Restrict Product Duplication

- **Description:** The system should restrict duplication of products.
- **Actor:** Vendor
- **Precondition:** The user is attempting to add an already existing product to their inventory.
- **Postcondition:** The system prevents the addition of duplicate products and alerts the user.
- **Details:**
 1. The user tries to add a product with the same name or category as an existing product in their inventory.
 2. The system will alert the user that the product already exists in their inventory and instead offer to update the existing product.
 3. The system does not allow duplication of an already existing product within the inventory.

Admin Portal (FR4)

FR4.1: Add New User

- **Description:** The system should allow the existing user to add a new user (Figure 6.1).
- **Actor:** Admin
- **Precondition:** The existing user is logged in.
- **Postcondition:** The existing user is directed to the add details page.
- **Details:**
 1. The system shall display the option to add a new user to the existing user.
 2. After selecting the option to add a new user, the system shall direct the user to the add details page.

The screenshot shows the WASAIL Admin Portal interface. On the left, there is a sidebar with various management options: User Management (selected), Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. The main content area is titled "Users" and has a "Add New" button. A success message "New User Created. [Edit User](#)" is displayed. Below this is a search bar and a table listing four users. The table columns are "Username", "Name", and "Email". The data is as follows:

Username	Name	Email
fatima	Fatima Ali	fatimaalitirmizi12@gmail.com
irtaza	Irtaza Ahmad	yrrebeere@gmail.com
fizza	Fizza Adeel	fizza.adeel19@gmail.com
malaika	Malaika Sultan	malaikasultant@gmail.com

Figure 6.1

FR4.2: Add New User Details

- **Description:** The system should allow the existing user to add details of the new user (Figure 6.2).
- **Actor:** Admin
- **Precondition:** The existing user is on the add details page.
- **Postcondition:** A new user is added to the Admin Portal.
- **Details:**
 1. The existing user adds details of the new user which includes username, email, first name, last name, and password.
 2. The user selects the add new user option after adding the details.
 3. The system shall notify the new user via email.
 4. The system shall add the new user to the database.

The screenshot shows the 'Add New User' page of the WASAIL Admin Portal. On the left, there is a sidebar with icons for User Management (selected), Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. The main area has a title 'Add New User' and a subtitle 'Create a brand new user and add them to this site.' It contains five input fields: 'Username (required)' with value 'fatima', 'Email (required)' with value 'fatimaalitirmizi12@gmail.com', 'First Name' with value 'Fatima', 'Last Name' with value 'Ali', and 'Password' (empty). Below these is a checkbox labeled 'Send User Notification' which is unchecked. At the bottom is a blue 'Add New User' button.

Figure 6.2

FR4.3: User Login

- **Description:** The system should allow the user to login using their credentials (Figure 6.3).
- **Actor:** Admin
- **Precondition:** The user is not logged in.
- **Postcondition:** The user is logged in.
- **Details:**
 1. User provides their valid email and password.
 2. The system validates the user's credentials.
 3. Upon successful validation, the user is granted access.

WASAIL

Username

Password

[Reset](#) [Password](#) Log In

Figure 6.3

FR4.4: Edit User Profile

- **Description:** The system should allow the user to edit their profile (Figure 6.4).
- **Actor:** Admin
- **Precondition:** The user is logged in.
- **Postcondition:** The user's profile has been edited.
- **Details:**
 1. The system shall display the option to edit the profile.
 2. The user can edit their credentials including username, email, first name, last name, and password.
 3. After editing the user shall save the updated version.
 4. The system shall update the database.

The screenshot shows the WASAIL application interface. On the left, there is a sidebar with various management options: User Management (selected), Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. The main area is titled "Users" and contains a button "Add New". A success message "New User Created. [Edit User](#)" is displayed. Below this is a search bar and a table listing four users:

Username	Name	Email
fatima	Fatima Ali	fatimaalitirmizi12@gmail.com
irtaza	Irtaza Ahmad	yrrebeere@gmail.com
fizza	Fizza Adeel	fizza.adeel19@gmail.com
malaika	Malaika Sultan	malaikasultant@gmail.com

Each row in the table includes "Edit", "Delete", and "View" links.

Figure 6.4

FR4.5: Delete User Profile

- **Description:** The system should allow the user to delete their profile (Figure 6.5).
- **Actor:** Admin
- **Precondition:** The user is logged in.
- **Postcondition:** The user's profile has been deleted.
- **Details:**
 1. The system shall display the option to delete the profile.
 2. After successful deletion, the system shall remove it from the database.

The screenshot shows the WASAIL application interface. On the left, there is a sidebar with various management options: User Management (selected), Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. The main area is titled "Users" and contains a button "Add New". Below this, a message "New User Created. [Edit User](#)" is displayed. To the right is a search bar and a "Search Users" button. A table lists four users with the following data:

Username	Name	Email
fatima	Fatima Ali	fatimaalitirmizi12@gmail.com
irtaza	Irtaza Ahmad	yrrebeere@gmail.com
fizza	Fizza Adeel	fizza.adeel19@gmail.com
malaika	Malaika Sultan	malaikasultant@gmail.com

Figure 6.5

FR4.6: View Grocery Store's Profile

- **Description:** The system should allow the user to view the grocery store's profile (Figure 6.6).
- **Actor:** Admin
- **Precondition:** The user is logged in.
- **Postcondition:** The system displays the grocery store's profile.
- **Details:**
 1. The user can select the grocery store's profile to view it.
 2. The system retrieves the grocery store's information including name, shop name, mobile number, address, and location.

The screenshot shows the WASAIL application interface. On the left, there is a sidebar with the following menu items:

- User Management
- Grocery Management** (highlighted)
- Vendor Management
- ML Configuration
- Analytics
- Content Management

On the right, the main area is titled "Grocery Management". It contains a search bar labeled "Search Users" and a table listing four grocery stores:

Icons	Name	Phone Number	Actions
	Jalal Sons	0300-9876543	Delete View
	Esajee's	0312-8766542	Delete View
	Al-Fateh	0314-6247966	Delete View
	Metro	0305-7654836	Delete View

Figure 6.6

FR4.7: Disable Grocery Store's Profile

- **Description:** The system should allow the user to delete the grocery store's profile (Figure 6.7).
- **Actor:** Admin
- **Precondition:** The user is logged in.
- **Postcondition:** The system has disabled the grocery store's profile.
- **Details:**
 1. The user can select the grocery store's profile to disable it.
 2. The system disables the grocery store's profile.

The screenshot shows the WASAIL platform interface. On the left, there is a sidebar with the following menu items:

- User Management
- Grocery Management** (highlighted)
- Vendor Management
- ML Configuration
- Analytics
- Content Management

On the right, the main area is titled "Grocery Management". It displays a table of grocery store profiles:

Icons	Name	Phone Number	Actions
	Jalal Sons	0300-9876543	Delete View
	Esajee's	0312-8766542	Delete View
	Al-Fateh	0314-6247966	Delete View
	Metro	0305-7654836	Delete View

Figure 6.7

FR4.8: View Vendor's Profile

- **Description:** The system should allow the user to view the vendor's profile (Figure 6.8).
- **Actor:** Admin
- **Precondition:** The user is logged in.
- **Postcondition:** The system displays the vendor's profile.
- **Details:**
 1. The user can select the vendor's profile in order to view it.
 2. The system retrieves the vendor's information including name, mobile number, areas in which they deliver.

The screenshot shows the WASAIL application interface. On the left is a sidebar with the following menu items:

- User Management
- Grocery Management
- Vendor Management** (highlighted)
- ML Configuration
- Analytics
- Content Management

The main area is titled "Vendor Management". It features a search bar with placeholder text "Search Users" and a table displaying vendor information:

Icons	Name	Phone Number	Actions
	Vendor A	0300-9876543	Delete View
	Vendor B	0312-8766542	Delete View
	Vendor C	0314-6247966	Delete View
	Vendor D	0305-7654836	Delete View

Figure 6.8

FR4.9: Disable Vendor's Profile

- **Description:** The system should allow the user to delete the vendor's profile (Figure 6.9).
- **Actor:** Admin
- **Precondition:** The user is logged in.
- **Postcondition:** The system has disabled the vendor's profile.
- **Details:**
 1. The user can select the vendor's profile to disable it.
 2. The system disables the vendor's profile.

The screenshot shows the WASAIL application interface. On the left is a sidebar with navigation links: User Management, Grocery Management, Vendor Management (which is selected and highlighted in grey), ML Configuration, Analytics, and Content Management. The main area is titled "Vendor Management". It features a search bar with placeholder text "Search Users" and a table listing four vendors. The table columns are Icons, Name, Phone Number, and Actions. Each vendor row includes a small thumbnail icon, the vendor's name, their phone number, and two action buttons: "Delete" and "View".

Icons	Name	Phone Number	Actions
	Vendor A	0300-9876543	Delete View
	Vendor B	0312-8766542	Delete View
	Vendor C	0314-6247966	Delete View
	Vendor D	0305-7654836	Delete View

Figure 6.9

FR4.10: Select Grocery Store's ML Model

- **Description:** The system should allow the user to select the ML model for the grocery store (Figure 6.10).
- **Actor:** Admin
- **Precondition:** The user is logged in.
- **Postcondition:** The system has allowed the user to select the ML model for the grocery store.
- **Details:**
 1. The user can select a ML model from a list generated by the system.
 2. The system will save the option selected by the user.

The screenshot shows the WASAIL application interface. On the left, there is a sidebar with the following menu items:

- User Management
- Grocery Management
- Vendor Management
- ML Configuration** (highlighted in grey)
- Analytics
- Content Management

The main content area is titled "ML Configuration". It contains a search bar with placeholder "Search Users" and a table with the following data:

Icons	Name	Model	RMSLE
	Jalal Sons	Jalal Sons Model A	0.23
	Esajee's	Esajee's Model B	0.5
	Al-Fateh	Al-Fateh Model A	1.2
	Metro	Metro Model C	0.95

Figure 6.10

FR4.11: Display RMSLE value

- **Description:** The system should display the RMSLE value to the user (Figure 6.11).
- **Actor:** Admin
- **Precondition:** The user has selected a ML model.
- **Postcondition:** The system has displayed the RMSLE value.
- **Details:**
 1. The system calculates the RMSLE value against the selected ML model.
 2. The system displays the RMSLE value for the selected ML model.

The screenshot shows the WASAIL application interface. On the left, there is a sidebar with the following menu items: User Management, Grocery Management, Vendor Management, ML Configuration (which is highlighted in grey), Analytics, and Content Management. The main content area is titled "ML Configuration". It features a search bar with placeholder text "Search Users" and a table displaying four ML models. The table has columns for Icons, Name, Model, and RMSLE. The data is as follows:

Icons	Name	Model	RMSLE
	Jalal Sons	Jalal Sons Model A	0.23
	Esajee's	Esajee's Model B	0.5
	Al-Fateh	Al-Fateh Model A	1.2
	Metro	Metro Model C	0.95

Figure 6.11

FR4.12: Display Analytics

- **Description:** The system should display the analytics to the user (Figure 6.12).
- **Actor:** Admin
- **Precondition:** The user is logged in and on the analytics page.
- **Postcondition:** The system has displayed the analytics to the user.
- **Details:**
 1. The systems shall display the option to view analytics.
 2. Once selected, the system shall display different analytics including the total number of groceries stores that have registered, the total number of vendors that have registered, the total number of SKUs that are present.
 3. The user shall be able to view these analytics.

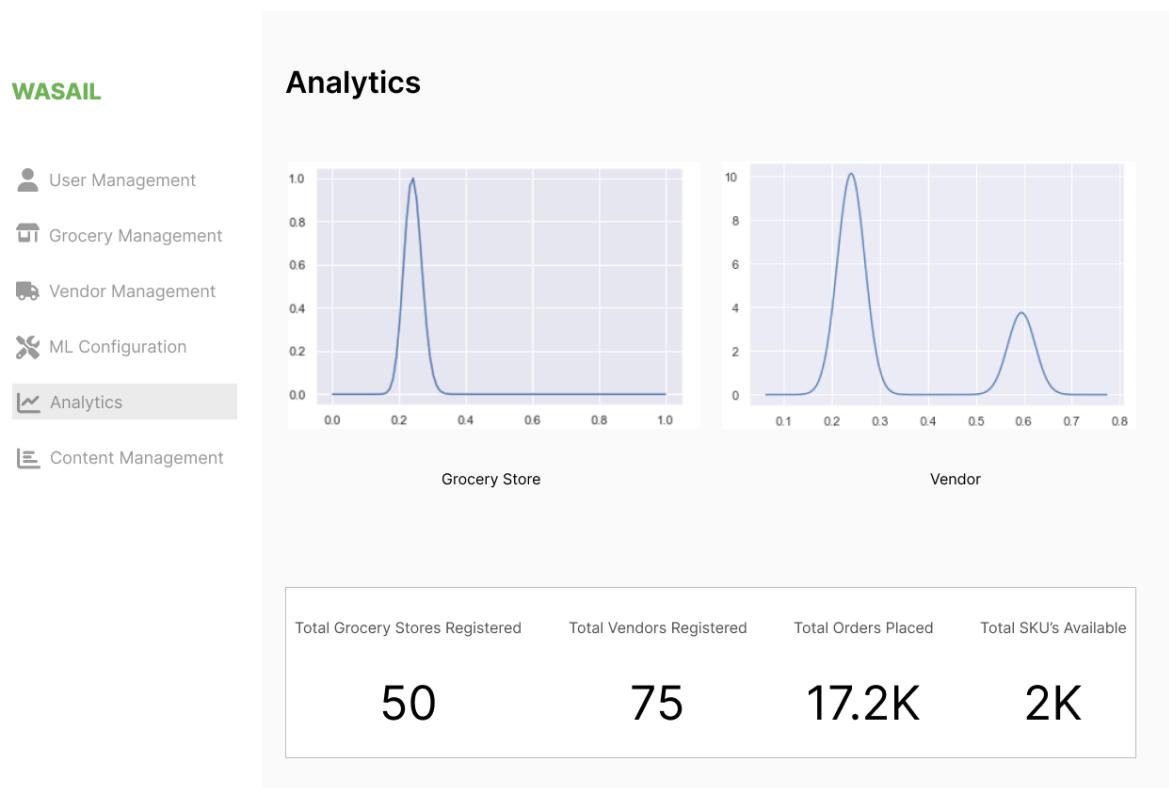


Figure 6.12

FR4.13: Add Category

- **Description:** The system should allow the user to add a category of products (Figure 6.13).
- **Actor:** Admin
- **Precondition:** The user is logged in and on the products listings page.
- **Postcondition:** The user has added a category.
- **Details:**
 1. The system shall display an option to add a category of the products for the database (through which the vendors would be able to add products to their own inventory)
 2. The user can add the category name for example dairy, vegetables, condiments, meat etc.
 3. The system shall save them in the database.

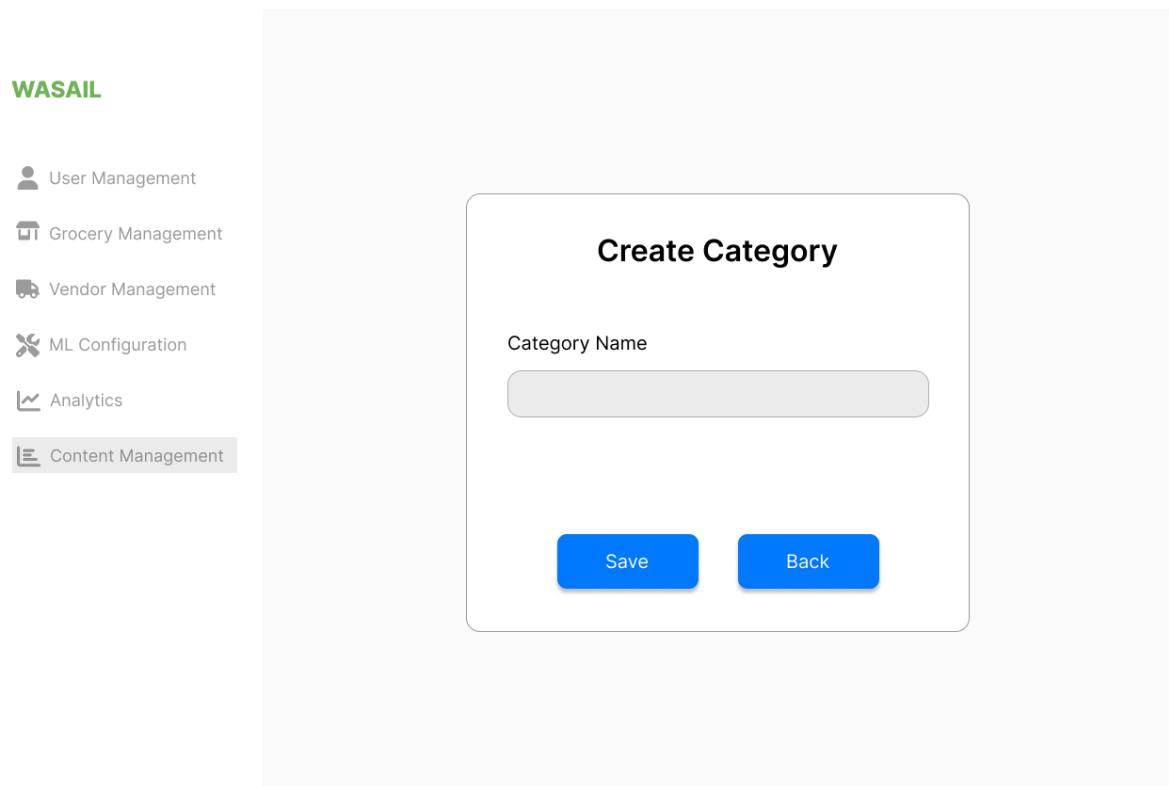


Figure 6.13

FR4.14: Update Category

- **Description:** The system should allow the user to update the category (Figure 6.14).
- **Actor:** Admin
- **Precondition:** The user is logged in and on the products listings page.
- **Postcondition:** The user has updated the category.
- **Details:**
 1. The system shall give the option to update the category name.
 2. The user can select the option and update the category name.
 3. The system shall save the changes in the database.

The screenshot shows the 'WASAIL' application interface. On the left, there is a vertical sidebar with icons and labels for User Management, Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. 'Content Management' is highlighted with a gray background. The main area is titled 'List Category'. It features a blue 'Add Category' button at the top left. Below it is a search bar with a placeholder 'Search Category' and a clear button. A table lists four categories: Dairy, Vegetables, Snacks, and Meat. Each row has an 'Actions' column with 'Update' and 'Delete' links. At the bottom of the table is a 'List Product' link.

Category Name	Actions
Dairy	Update Delete
Vegetables	Update Delete
Snacks	Update Delete
Meat	Update Delete

Figure 6.14

FR4.15: Delete Category

- **Description:** The system should allow the user to delete the category (Figure 6.15).
- **Actor:** Admin
- **Precondition:** The user is logged in and on the products listings page.
- **Postcondition:** The user has deleted the category.
- **Details:**
 1. The system shall give the option to delete the category.
 2. Once the user has deleted the category, the system shall remove the category from the database as well.

The screenshot shows the 'WASAIL' application interface. On the left, there is a sidebar with the following navigation options: User Management, Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. The 'Content Management' option is currently selected and highlighted with a grey background. The main content area is titled 'List Category'. At the top right of this area is a blue button labeled 'Add Category'. Below the title, there is a search bar consisting of a text input field and a 'Search Category' button. The main content is a table titled 'Category Name' with four rows. Each row contains a category name ('Dairy', 'Vegetables', 'Snacks', 'Meat') and two actions: 'Update' and 'Delete', represented by blue and red text respectively. At the bottom of the table, there is a link labeled 'List Product'.

Category Name	Actions
Dairy	Update Delete
Vegetables	Update Delete
Snacks	Update Delete
Meat	Update Delete

Figure 6.15

FR4.16: Add Product

- **Description:** The system should allow the user to add product details (Figure 6.16).
- **Actor:** Admin
- **Precondition:** The user is logged in and on the products listings page.
- **Postcondition:** The user has added a product.
- **Details:**
 1. The system shall display an option to add a product to the database (through which the vendors would be able to add products to their own inventory)
 2. The user can add the product name, can upload an image, and can select from the category.
 3. The system shall save them in the database.

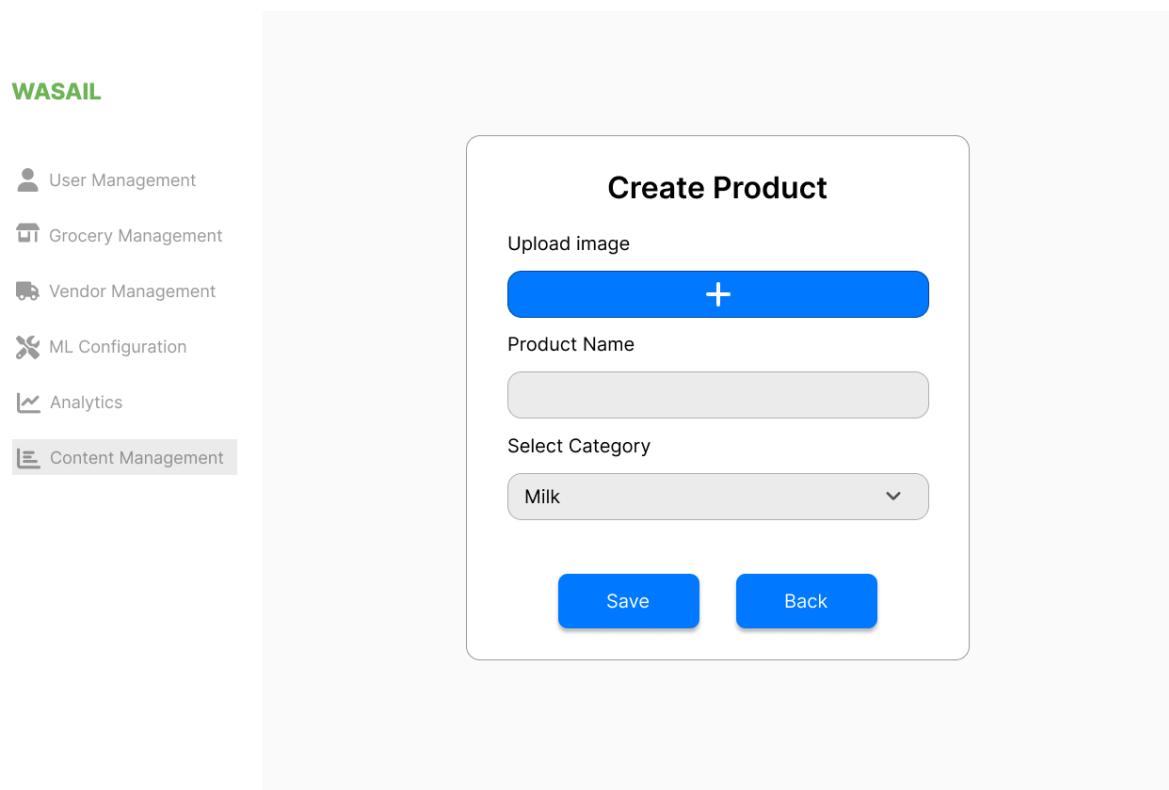


Figure 6.16

FR4.17: Update Product

- **Description:** The system should allow the user to update the product (Figure 6.17).
- **Actor:** Admin
- **Precondition:** The user is logged in and on the products listings page.
- **Postcondition:** The user has updated the product.
- **Details:**
 1. The system shall give the option to update the product name, the picture or the category.
 2. The user can select the option and update the product name, the picture or the category.
 3. The system shall save the changes in the database.

The screenshot shows the 'WASAIL' application interface. On the left, there is a sidebar with the following navigation options: User Management, Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. The 'Content Management' option is currently selected and highlighted with a grey background. The main area is titled 'List Product' and contains a blue 'Add Product' button. Below the button is a search bar with a placeholder 'Search Product'. A table displays four product entries:

Image	Product Name	Category	Actions
	Olper's Milk Pack	Dairy	Update Delete
	Carrots	Vegetable	Update Delete
	Cookies	Snacks	Update Delete
	Minced Beef	Meat	Update Delete

At the bottom of the main area, there is a link labeled 'List Category'.

Figure 6.17

FR4.18: Delete Product

- **Description:** The system should allow the user to delete the product (Figure 6.18).
- **Actor:** Admin
- **Precondition:** The user is logged in and on the products listings page.
- **Postcondition:** The user has deleted the product.
- **Details:**
 3. The system shall give the option to delete the product.
 4. Once the user has deleted the product, the system shall remove the product from the database as well.

The screenshot shows the WASAIL application interface. On the left, there is a sidebar with the following menu items: User Management, Grocery Management, Vendor Management, ML Configuration, Analytics, and Content Management. The Content Management item is highlighted with a light gray background. The main area is titled "List Product". It features a blue "Add Product" button at the top left. Below it is a search bar with a placeholder and a "Search Product" button. A table displays four product entries:

Image	Product Name	Category	Actions
	Olper's Milk Pack	Dairy	Update Delete
	Carrots	Vegetable	Update Delete
	Cookies	Snacks	Update Delete
	Minced Beef	Meat	Update Delete

At the bottom left of the main area, there is a link labeled "List Category".

Figure 6.18

Non-Functional Requirements

- **Performance**
 - **Scalability:** The system should be able to handle a growing number of users, both grocery stores and vendors, without a significant decrease in performance.
- **Security**
 - **Access Control:** Role-based access control should be in place to restrict unauthorised access to sensitive functionalities and data.
- **Usability**
 - **Multilingual Support:** The system should support multiple languages (English, Roman Urdu, Urdu) as per FR1.1 to cater to a diverse user base.
 - **User-Friendly Interface:** The user interface should be intuitive and easy to use, ensuring a seamless experience for both grocery stores and vendors.
- **Data Storage**
 - **Data Capacity:** The system should be capable of handling a large volume of user data, product listings, and order history efficiently.
- **Notification System**
 - **Real-Time Notifications:** The notification system should deliver real-time updates on orders, deliveries, and important events to users.
- **Error Handling**
 - **Error Messages:** Clear and informative error messages should be provided to assist users in troubleshooting issues.
- **Reporting and Logging**
 - **Activity Logging:** The system should maintain logs of user activities, including login attempts, order changes, and product updates, to facilitate auditing and troubleshooting.
- **Integration**
 - **Third-Party Integrations:** The system should seamlessly integrate with SMS services for OTP delivery, and other third-party services as required.
- **Mobile Responsiveness**
 - **Cross-Platform Compatibility:** The system should be accessible and user-friendly on various mobile devices.

Test Cases

Test Case 1: Language Selection (FR1.1)

- **Test Scenario:** The user wants to select their preferred language.
- **Preconditions:** The user has not specified their preferred language.
- **Test Steps:**
 1. Open the app.
 2. Navigate to the language selection option.
 3. Select English as the preferred language.
- **Expected Result:** The app's text is displayed in English. English is saved as the preferred language.
- **Alternate Flow 1:** User selects Urdu as the preferred language.
- **Expected Result:** The app's text is displayed in Urdu. Urdu is saved as the preferred language.

Test Case 2: Phone Registration (FR1.2)

- **Test Scenario:** The user wants to create an account using their phone number.
- **Preconditions:** The user has opened the app.
- **Test Steps:**
 1. Open the app.
 2. Navigate to the phone registration option.
 3. Enter a valid phone number.
 4. Tap the "Register" button.
- **Expected Result:** The user is directed to the phone number confirmation screen. The system validates the phone number format.
- **Alternate Flow 1:** User enters an invalid phone number.
- **Expected Result:** The system displays an error message indicating that the phone number format is incorrect and prompts the user to enter a valid phone number.

Test Case 3: Phone Number Confirmation (FR1.3)

- **Test Scenario:** The user wants to confirm their entered phone number.
- **Preconditions:** The user has entered their phone number.
- **Test Steps:**
 1. Enter the phone number confirmation screen.
 2. Confirm if the entered phone number is correct.
 3. If incorrect, edit the phone number.
 4. Upon confirmation, the system asks the user to create an account

- **Expected Result:** The user is directed to the OTP screen if the phone number is confirmed.
- **Alternate Flow:** User's phone number is not confirmed
- **Expected Results:** The system displays an error message indicating that the phone number has not been confirmed and prompts the user to try checking their internet connection again.

Test Case 4: OTP Code Generation and Delivery (FR1.4)

- **Test Scenario:** The user receives an OTP code for phone number verification.
- **Preconditions:** The user has confirmed their phone number.
- **Test Steps:**
 1. The system generates a 4-digit unique OTP code.
 2. Sends the OTP code to the user's phone number via SMS.
 3. User is asked to enter the OTP.
 4. The system verifies the validity of the OTP code.
- **Expected Result:** The user is directed to the account details page upon successful verification.
- **Alternate Flow 1:** User enters an incorrect OTP code.
- **Expected Result:** The system displays an error message indicating that the OTP code is incorrect and prompts the user to re-enter it.

Test Case 5: Login (FR1.5)

- **Test Scenario:** Registered users want to log in using their credentials.
- **Preconditions:** The user is not logged in.
- **Test Steps:**
 1. Enter the phone number.
 2. If the phone number is registered, enter the password.
 3. The system validates the user's credentials.
- **Expected Result:** Upon successful validation, the user is granted access to their profile.
- **Alternate Flow 1:** User enters an incorrect password.
- **Expected Result:** The system displays an error message indicating that the password is incorrect.

Test Case 6: Logout (FR1.6)

- **Test Scenario:** The user wants to log out of the system.
- **Preconditions:** The user is logged in.

- **Test Steps:**
 1. The system gives the user the option to log out.
 2. The user logs out of the system using the option.
- **Expected Result:** The user is logged out of the system.
- **Alternative Flow 1:** The user attempts to log out but fails to do so and the system displays a possible reason for the error.
- **Expected Result:** The user is able to resolve the issue and then log out.

Test Case 7: Reset Password (FR1.7)

- **Test Scenario:** The user wants to reset their password.
- **Preconditions:** The user is registered.
- **Test Steps:**
 1. The system gives the user the option to reset their password.
 2. The system generates a 4-digit unique OTP code.
 3. Sends the OTP code to the user's phone number via SMS.
 4. Upon verification, the user enters their new password.
 5. The user re-enters their password for confirmation.
 6. After confirmation, the user saves the password.
- **Expected Result:** The user's password is successfully reset, and the system updates it in the database.

Test Case 8: View Profile (FR1.8)

- **Test Scenario:** The user wants to view their own profile.
- **Preconditions:** The user is on their own profile.
- **Test Steps:**
 1. The system displays the user's profile.
 2. The user can view their account details.
- **Expected Result:** The user is able to view their profile.

Test Case 9: Account Details (FR2.9)

- **Test Scenario:** The user wants to enter their account details.
- **Preconditions:** The user has verified their phone number.
- **Test Steps:**
 1. The user enters their account details, including their name, store's name, store's address, password, and location.
 2. User account information is securely stored in the database.
- **Expected Result:** The user's account details are successfully saved in the database.

Test Case 10: Search Product (FR2.10)

- **Test Scenario:** The user wants to search for products based on product name and category.
- **Preconditions:** The user is logged in and on the home page.
- **Test Steps:**
 1. The user enters search criteria, such as a product name or category.
 2. The system retrieves and displays the list of matching products along with the vendors that sell them.
- **Expected Result:** The user sees a list of matching products and vendors. If no matches are found, the system provides appropriate feedback.

Test Case 11: Search Vendor (FR2.11)

- **Test Scenario:** The user wants to search vendors based on their name.
- **Preconditions:** The user is logged in and on the home page.
- **Test Steps:**
 1. The user enters the search criteria, such as a vendor name.
 2. The system retrieves and displays the vendor along with its popular products.
- **Expected Result:** The user sees a list of vendors that deliver in their area based on the search. If no matches are found, the system provides appropriate feedback.

Test Case 12: Browse Category (FR2.12)

- **Test Scenario:** The user wants to select a category from the home page directly.
- **Preconditions:** The user is logged in and on the home page.
- **Test Steps:**
 1. The user selects the categories that are being displayed to them.
 2. The system retrieves and displays products that fall under the category along with the vendors who sell them.
- **Expected Result:** The user views products and vendors based on the selected category. If no match is found, the system provides appropriate feedback.

Test Case 13: View Vendor Profile (FR2.13)

- **Test Scenario:** The user wants to view the vendor's profile.

- **Preconditions:** The user is logged in and has searched the vendor or is on the vendor list page.
- **Test Steps:**
 1. The user selects the vendor's profile to view it.
 2. The system retrieves the vendor's information, including their profile picture, name, area in which they deliver, and displays it for the user.
- **Expected Result:** The user is able to view the vendor's profile.

Test Case 14: Add Vendor to Vendor List (FR2.14)

- **Test Scenario:** The user wants to add the vendor to their vendor list.
- **Preconditions:** The user is logged in and on the vendor's profile.
- **Test Steps:**
 1. The user selects the option to add the vendor.
 2. The vendor is added to the vendors list of the user.
- **Expected Result:** The vendor is successfully added to the user's vendor list.

Test Case 15: Contact Vendor (FR2.15)

- **Test Scenario:** The user wants to contact the vendor.
- **Preconditions:** The user is logged in and on the vendor's profile.
- **Test Steps:**
 1. The system displays the vendor's phone number on the profile.
 2. The user can contact the vendor via message or phone call.
- **Expected Result:** The user successfully contacts the vendor through the provided contact information.

Test Case 16: View Products on the Vendor's Profile (FR2.16)

- **Test Scenario:** The user wants to view all the products that the vendor sells on their profile.
- **Preconditions:** The user is logged in and on the vendor's profile.
- **Test Steps:**
 1. The system retrieves all the products that the vendor sells and displays them to the user.
 2. The user can view the products and scroll through them.
- **Expected Result:** The user can view all the products offered by the vendor on their profile.

Test Case 17: View Searched Product (FR2.17)

- **Test Scenario:** The user wants to view the searched product on the vendor's profile.
- **Preconditions:** The user has searched for a product and is on the vendor's profile.
- **Test Steps:**
 1. The system displays the searched product on the vendor's profile.
 2. The user can view the products along with their prices and scroll through them.
- **Expected Result:** The user can view the searched product on the vendor's profile, including prices and details.

Test Case 18: Select Products (FR2.18)

- **Test Scenario:** The user wants to select the product they want to order.
- **Preconditions:** The user is logged in and on the vendor's profile.
- **Test Steps:**
 1. The system displays products on the vendor's profile.
 2. The user can select products from the product list to order them.
 3. The user can proceed to place the order.
- **Expected Result:** The user successfully selects products and proceeds to place an order.

Test Case 19: Order Recommendation (FR2.19)

- **Test Scenario:** The user wants to view the recommended amount to order.
- **Preconditions:** The user has selected a product.
- **Test Steps:**
 1. After product selection, the user is recommended an amount to order.
 2. The recommendation is based on..
- **Expected Result:** The user receives a recommendation for the amount of the selected product.

Test Case 20: Quantity Selection (FR2.20)

- **Test Scenario:** The user wants to add their own quantity to order.
- **Preconditions:** The user is logged in and has selected a product.
- **Test Steps:**
 1. The user is given the option to enter the amount of the product they want to order.

2. The user enters the desired quantity.
 3. The user adds the product to the cart.
- **Expected Result:** The user successfully enters the quantity and adds the product to the cart.

Test Case 21: Remove Product (FR2.21)

- **Test Scenario:** The user wants to remove a product from the cart.
- **Preconditions:** The product is added to the cart.
- **Test Steps:**
 1. The system gives the option to the user to remove the product from the cart.
 2. The user removes the product they do not want to order from the cart.
 3. The system stops displaying that product in their cart.
- **Expected Result:** The user successfully removes the product from the cart.

Test Case 22: Order Placement (FR2.22)

- **Test Scenario:** The user wants to place an order.
- **Preconditions:** The product is added to the cart.
- **Test Steps:**
 1. After the user has added the products to their cart, the system gives the option to place the order.
 2. The order is placed to the vendor.
 3. The vendor is notified.
 4. The order is stored in the database.
- **Expected Result:** The order is successfully placed, and the vendor is notified.

Test Case 23: View Order (FR2.23)

- **Test Scenario:** The user wants to view the current orders they have placed.
- **Preconditions:** The user is on the orders page.
- **Test Steps:**
 1. The system should display the orders that the user has placed.
 2. The user can view their orders.
- **Expected Result:** The user can view their current orders.

Test Case 24: Order Tracking (FR2.24)

- **Test Scenario:** The user wants to track the order.

- **Preconditions:** The user is on the orders page.
- **Test Steps:**
 1. The system displays all the orders that are placed by the user.
 2. The user can select the order they want to track.
 3. Upon order selection, the user can see which process the order is in: in process, on its way, or delivered.
 4. The system notifies the user when the order is delivered.
 5. The order is marked as delivered in the system.
- **Expected Result:** The user can track the progress of their order and receive notifications for the order's status.

Test Case 25: View Order History (FR2.25)

- **Test Scenario:** The user wants to view their previous orders.
- **Preconditions:** The user is on the orders history page.
- **Test Steps:**
 1. The system displays all the previous orders the user has placed.
 2. The user can view all their previous orders.
- **Expected Result:** The user can view their order history.

Test Case 26: View Vendor List (FR2.26)

- **Test Scenario:** The user wants to view the vendors in the vendor list.
- **Preconditions:** The user is on the vendor page.
- **Test Steps:**
 1. The system displays all the vendors that the user has added.
 2. The user can view the vendors in the vendor list.
- **Expected Result:** The user can view the vendors they have added to their list.

Test Case 27: Edit Profile (FR2.27)

- **Test Scenario:** The user wants to edit their profile details.
- **Preconditions:** The user is on their own profile.
- **Test Steps:**
 1. The system displays the profile to the user.
 2. The user can edit the account details, including name, store name, and address.
 3. The system updates the information in the database.
- **Expected Result:** The user's profile details are successfully edited and updated in the database.

Test Case 28: Notification (FR2.28)

- **Test Scenario:** The user wants to receive notifications for important events.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. The user receives notifications for the order being processed, when the order is on its way to be delivered, and when the order has been delivered.
- **Expected Result:** The user receives timely notifications for relevant events.

Test Case 29: Search Product in Inventory (FR3.9)

- **Test Scenario:** The vendor wants to search for a product in their inventory.
- **Preconditions:** The vendor is logged in and has products in their inventory.
- **Test Steps:**
 1. Open the app.
 2. Go to the "Inventory" section.
 3. Enter search criteria, such as product name or category.
 4. Tap the "Search" button.
- **Expected Result:** A list of matching products in the vendor's inventory is displayed, sorted by relevance.

Test Case 30: Add Product to Inventory (FR3.10)

- **Test Scenario:** The vendor wants to add a new product to their inventory.
- **Preconditions:** The vendor is logged in and on the inventory page.
- **Test Steps:**
 1. In the "Inventory" section, select the option to add a new product.
 2. Enter the details of the new product, quantity, and unit cost.
 3. Confirm the addition of the product to the inventory.
- **Expected Result:** The new product is successfully added to the vendor's inventory.

Test Case 31: Remove Product (FR3.12)

- **Test Scenario:** The vendor wants to remove a product from their inventory.
- **Preconditions:** The vendor is on the product's page.
- **Test Steps:**
 1. The system gives the option to the vendor to remove the product from their inventory.
 2. The vendor removes the product they no longer want in the inventory.

3. The system stops displaying that product in their inventory.
- **Expected Result:** The vendor successfully removes the product from their inventory.

Test Case 32: Edit Details of Product (FR3.13)

- **Test Scenario:** The vendor wants to edit the details of a product in their inventory.
- **Preconditions:** The vendor has added the product to their inventory.
- **Test Steps:**
 1. The system displays the product page for the vendor.
 2. The system gives the vendor the option to edit the product details, including listed quantity, available quantity, and unit cost.
 3. The vendor edits the product details.
 4. The system saves the edited product details.
- **Expected Result:** The product details in the inventory are successfully edited.

Test Case 33: View Inventory (FR3.14)

- **Test Scenario:** The vendor wants to view their inventory (product listings).
- **Preconditions:** The vendor is logged in.
- **Test Steps:**
 1. The vendor goes to the "Inventory" section.
 2. The system should display the inventory of the vendor.
 3. The vendor can view their inventory.
- **Expected Result:** The vendor can view their product listings in the inventory.

Test Case 34: View Current Orders (FR3.15)

- **Test Scenario:** The vendor wants to view the total number of current orders from all grocery stores.
- **Preconditions:** The vendor is logged in and has pending orders.
- **Test Steps:**
 1. The vendor goes to the "Orders" section.
 2. The system displays all the current orders that the vendor has received and needs to deliver.
 3. The vendor can view all the current orders.
- **Expected Result:** The vendor can view the list of current orders they need to fulfil.

Test Case 35: View Grocery Stores List (FR3.16)

- **Test Scenario:** The vendor wants to view the grocery store list.
- **Preconditions:** The vendor is logged in.
- **Test Steps:**
 1. The vendor goes to the "Stores List" section.
 2. The system displays a list of grocery stores that have placed orders with the vendor.
 3. The vendor can view the grocery stores in the list.
- **Expected Result:** The vendor can view the list of grocery stores that have placed orders.

Test Case 36: View Grocery Store Profile (FR3.17)

- **Test Scenario:** The vendor wants to view the profile of a specific grocery store.
- **Preconditions:** The vendor is on the store list page.
- **Test Steps:**
 1. The vendor selects a grocery store's profile to view.
 2. The system retrieves the grocery store's information, including their profile picture, name, and delivery area.
 3. The information is displayed for the vendor.
- **Expected Result:** The vendor can view the selected grocery store's profile.

Test Case 37: View Grocery Store Current Order (FR3.18)

- **Test Scenario:** The vendor wants to view the current order placed by a specific grocery store.
- **Preconditions:** The vendor is on the grocery store's profile page.
- **Test Steps:**
 1. The vendor selects the grocery store's profile.
 2. The system should display the current orders that the vendor has received from the grocery store.
 3. The vendor can view the current order placed by that grocery store.
- **Expected Result:** The vendor can view the current order details placed by the grocery store.

Test Case 38: View Orders History (FR3.19)

- **Test Scenario:** The vendor wants to view their order history, including previously completed deliveries.

- **Preconditions:** The vendor is on the orders history page.
- **Test Steps:**
 1. The system displays all the orders the vendor has delivered.
 2. The vendor can view all their delivered orders.
- **Expected Result:** The vendor can access and review their order history, including successfully delivered orders.

Test Case 39: Edit Profile (FR3.20)

- **Test Scenario:** The vendor wants to edit their profile details.
- **Preconditions:** The vendor is on their own profile.
- **Test Steps:**
 1. The system displays the vendor's profile.
 2. The vendor can edit the account details, including name, store name, and address.
 3. The system updates the information in the database.
- **Expected Result:** The vendor's profile details are successfully edited and updated in the database.
- **Alternative Flow 1:** Vendor encounters validation error editing profile (e.g., incorrect address, special characters). System prompts correction, the correct data format is used and profile edit is resubmitted.
- **Expected Result:** Validation requirements are not fulfilled so correct and valid data is given which will allow successful editing of the profile details.

Test Case 40: Order Dispatch Tracking (FR3.21)

- **Test Scenario:** The vendor wants to track the progress of order dispatch.
- **Preconditions:** The vendor is on the orders page.
- **Test Steps:**
 1. The system displays all the orders that are dispatched by the vendor.
 2. The vendor can select the order they want to track the dispatch progress for.
 3. The vendor updates the system on the progress of order delivery (in process, on its way, or delivered).
 4. The system notifies the vendor of the completed delivery.
- **Expected Result:** The vendor can track the status of order dispatch and receive notifications.
- **Alternative Flow 1:** Vendor faces a tech glitch updating order delivery progress, gets an error, retries, and successfully updates.
- **Expected Result:** The vendor overcomes the technical issue, successfully re-updates the progress of order delivery, and continues tracking.

- **Alternative Flow 2:** The system spots a vendor's delay, notifies them to update delivery progress, vendor promptly complies.
- **Expected Result:** The system proactively alerts the vendor about a potential delay, the vendor responds promptly, and the order dispatch tracking continues without issues.

Test Case 41: Restrict Product Duplication (FR3.22)

- **Test Scenario:** The vendor tries to add a product that already exists in their inventory.
- **Preconditions:** The vendor is adding a product to their inventory.
- **Test Steps:**
 1. The vendor tries to add a product with the same name or category as an existing product in their inventory.
- **Expected Result:** The system should alert the vendor that the product already exists in their inventory and offer to update the existing product. The system should not allow duplication of an existing product.
- **Alternative Flow 1:** The system becomes aware of duplication but instead of alerting, it automatically adds the existing product with the new information provided by the vendor.
- **Expected Result:** The system identifies duplication and updates the existing product with new information from the vendor, ensuring data consistency.

Test Case 42: Add New User (FR4.1)

- **Test Scenario:** Adding a new user by the existing admin.
- **Preconditions:** The existing user is logged in.
- **Test Steps:**
 1. Go to the Admin Portal.
 2. Find and select the "Add New User" option.
 3. Fill in the user details (username, email, first name, last name, password).
 4. Save the details.
- **Expected Result:** The system successfully adds a new user and directs to the details page.
- **Alternate Flow 1: Missing Information**
- **Test Steps:**
 1. Enter incomplete user details.
 2. Save the information.
- **Expected Result:** The system prompts an error message for incomplete data entry and prevents saving until all necessary fields are filled.

Test Case 43: Add New User Details (FR4.2)

- **Test Scenario:** Adding details of the new user by the existing admin.
- **Preconditions:** The existing user is on the "Add Details" page.
- **Test Steps:**
 1. Enter user details including username, email, first name, last name, and password.
 2. Save the details.
 3. Confirm the prompt to notify the new user via email.
- **Expected Result:** The new user is added to the Admin Portal and a notification is sent.
- **Alternate Flow 1:** Email Notification Failure
- **Test Steps:**
 1. Save the details but the email notification fails.
- **Expected Result:** The user is added to the system but a failure message is generated due to unsuccessful email notification.

Test Case 44: User Login (FR4.3)

- **Test Scenario:** Logging in as a user using valid credentials.
- **Preconditions:** The user is not logged in.
- **Test Steps:**
 1. Enter the valid email and password.
 2. Click on the login button.
- **Expected Result:** The system validates the user's credentials and grants access upon successful validation.
- **Alternate Flow 1:** Invalid Credentials
- **Test Steps:**
 1. Enter invalid email and/or password.
 2. Click on the login button.
- **Expected Result:** The system denies access and prompts an error message for invalid credentials.

Test Case 45: Edit User Profile (FR4.4)

- **Test Scenario:** Editing the user's profile details.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. Find and select the option to edit the profile.
 2. Modify the desired credentials (username, email, first name, last name, or password).
 3. Save the updated details.

- **Expected Result:** The system successfully updates the user's profile and database.
- **Alternate Flow 1:** Connection Failure
- **Test Steps:**
 1. Modify the profile details.
 2. Attempt to save changes without an internet connection.
- **Expected Result:** The system displays an error message, unable to save changes due to the lack of internet connectivity.

Test Case 46: Delete User Profile (FR4.5)

- **Test Scenario:** Deleting the user's profile.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. Locate and select the option to delete the profile.
 2. Confirm the deletion.
 3. System prompts for confirmation.
- **Expected Result:** The system successfully removes the user's profile from the database.
- **Alternate Flow 1:** Accidental Deletion
- **Test Steps:**
 1. Initiate profile deletion.
 2. Hesitate and cancel the deletion.
- **Expected Result:** The system confirms and prevents the profile deletion upon user confirmation of cancellation.

Test Case 47: View Grocery Store's Profile (FR4.6)

- **Test Scenario:** Viewing details of a grocery store.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. Select the grocery store's profile.
 2. Review the information displayed, including name, shop name, mobile number, address, and location.
- **Expected Result:** The system accurately retrieves and displays the grocery store's information.
- **Alternate Flow 1:** Missing Information
- **Test Steps:**
 1. Access the grocery store's profile.
 2. Confirm if any required details are missing or incorrectly displayed.
- **Expected Result:** The system indicates missing or incomplete information, providing an error message or a blank field for the missing data.

Test Case 48: Disable Grocery Store's Profile (FR4.7)

- **Test Scenario:** Disabling a grocery store's profile.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. Find and select the option to disable the grocery store's profile.
 2. Confirm the action
- **Expected Result:** The system disables the grocery store's profile and removes it from the active listings.
- **Alternate Flow 1:** Inadvertent Disabling
- **Test Steps:**
 1. Initiate the disabling process for a grocery store.
 2. Cancel the process before confirmation.
- **Expected Result:** The system confirms cancellation of the disabling action and keeps the grocery store's profile active.

Test Case 49: View Vendor's Profile (FR4.8)

- **Test Scenario:** Accessing details of a vendor's profile.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. Click to access the vendor's profile.
 2. Review the displayed information including name, mobile number, and delivery areas.
- **Expected Result:** The system accurately retrieves and displays the vendor's information.
- **Alternate Flow 1:** Unavailable Details
- **Test Steps:**
 1. Review the vendor's profile.
 2. Check for any missing or incomplete information.
- **Expected Result:** The system identifies and flags any missing or incomplete details, providing an error message or a blank field for the missing data.

Test Case 50: Disable Vendor's Profile (FR4.9)

- **Test Scenario:** Disabling a vendor's profile.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. Find the vendor's profile to disable.
 2. Initiate the disabled action and confirm.
- **Expected Result:** The system disables the vendor's profile and removes it from the active vendor listings.

- **Alternate Flow 1:** Accidental Disabling
- **Test Steps:**
 1. Initiate the disabling process for a vendor.
 2. Cancel the action before confirmation.
- **Expected Result:** The system confirms the cancellation of the disabling action and retains the vendor's profile in the active listings.

Test Case 51: Select Grocery Store's ML Model (FR4.10)

- **Test Scenario:** Selecting an ML model for a grocery store.
- **Preconditions:** The user is logged in.
- **Test Steps:**
 1. Navigate to the ML model selection section.
 2. Choose a specific ML model for a grocery store.
 3. Confirm the selection.
- **Expected Result:** The system stores the user's ML model selection for the chosen grocery store.
- **Alternate Flow 1:** Model Selection Conflict
- **Test Steps:**
 1. Attempt to select a model that's incompatible with the grocery store's requirements.
 2. Confirm the selection.
- **Expected Result:** The system alerts the user about the incompatibility and advises the selection of a suitable model.

Test Case 52: Display RMSLE Value (FR4.11)

- **Test Scenario:** Displaying the RMSLE value for a selected ML model.
- **Preconditions:** The user has selected an ML model.
- **Test Steps:**
 1. Access the section to view RMSLE values.
 2. Review the displayed RMSLE value for the selected ML model.
- **Expected Result:** The system accurately calculates and displays the RMSLE value for the selected ML model.
- **Alternate Flow 1:** Calculation Error
- **Test Steps:**
 1. Access the RMSLE value section.
 2. Cross-check the calculated RMSLE value against the expected value for the selected ML model.
- **Expected Result:** If the displayed RMSLE value differs significantly from the expected value, the system might have encountered a calculation error.

Test Case 53: Display Analytics (FR4.12)

- **Test Scenario:** Displaying analytics data.
- **Preconditions:** The user is logged in and on the analytics page.
- **Test Steps:**
 1. Navigate to the analytics section.
 2. Review the displayed analytics data, including the number of registered grocery stores, vendors, and SKUs.
- **Expected Result:** The system correctly showcases the analytics data for review.
- **Alternate Flow 1:** Analytics Data Discrepancy
- **Test Steps:**
 1. Access the analytics section.
 2. Compare the displayed analytics with the actual counts from the database.
- **Expected Result:** Any discrepancy between the displayed analytics and actual counts should prompt a system recheck or possible data update.

Test Case 54: Add Category (FR4.13)

- **Test Scenario:** Adding a category of products.
- **Preconditions:** The user is logged in and on the products listings page.
- **Test Steps:**
 1. Access the section to add a product category.
 2. Input the new category name and confirm the addition.
- **Expected Result:** The system adds the new category to the product database.
- **Alternate Flow 1:** Duplicate Category Addition
- **Test Steps:**
 1. Attempt to add a category that already exists in the database.
 2. Confirm the addition.
- **Expected Result:** The system prevents the addition of a duplicate category and alerts the user about the existing entry.

Test Case 55: Update Category (FR4.14)

- **Test Scenario:** Updating a product category.
- **Preconditions:** The user is logged in and on the products listings page.
- **Test Steps:**
 1. Find the section to update a product category.
 2. Choose a category and input the updated name.
 3. Confirm the update.

- **Expected Result:** The system updates the selected category name in the database.
- **Alternate Flow 1:** Invalid Category Update
- **Test Steps:**
 1. Attempt to update a non-existent category.
 2. Confirm the update.
- **Expected Result:** The system flags an error, notifying the user that the category doesn't exist for update.

Test Case 56: Delete Category (FR4.15)

- **Test Scenario:** Deleting a product category.
- **Preconditions:** The user is logged in and on the products listings page.
- **Test Steps:**
 1. Navigate to the section to delete a product category.
 2. Select a category for deletion and confirm the action.
- **Expected Result:** The system removes the selected category from the product database.
- **Alternate Flow 1:** Delete Default Category
- **Test Steps:**
 1. Attempt to delete a system-defined/default category.
 2. Confirm the deletion.
- **Expected Result:** The system prohibits the deletion of system-defined categories and notifies the user about this restriction.

Test Case 57: Add Product (FR4.16)

- **Test Scenario:** Adding product details.
- **Preconditions:** The user is logged in and on the products listings page.
- **Test Steps:**
 1. Access the option to add a new product.
 2. Enter the product name, upload an image, and choose a category.
 3. Confirm the addition.
- **Expected Result:** The system successfully adds the new product details to the database.
- **Alternate Flow 1:** Incomplete Product Addition
- **Test Steps:**
 1. Attempt to add a product with missing essential information.
 2. Confirm the addition.
- **Expected Result:** The system rejects the addition and prompts the user to input all mandatory product details.

Test Case 58: Update Product (FR4.17)

- **Test Scenario:** Updating product details.
- **Preconditions:** The user is logged in and on the products listings page.
- **Test Steps:**
 1. Access the section to update a product.
 2. Choose a product to modify, update the name, image, or category, and confirm the changes.
- **Expected Result:** The system successfully updates the selected product details in the database.
- **Alternate Flow 1:** Update Nonexistent Product
- **Test Steps:**
 1. Attempt to update a product that doesn't exist.
 2. Confirm the update.
- **Expected Result:** The system notifies the user about the absence of the selected product for updating.

Test Case 59: Delete Product (FR4.18)

- **Test Scenario:** Deleting a product.
- **Preconditions:** The user is logged in and on the products listings page.
- **Test Steps:**
 1. Navigate to the section to delete a product.
 2. Select a product for deletion and confirm the action.
- **Expected Result:** The system removes the selected product from the product database.
- **Alternate Flow 1:** Delete Unavailable Product
- **Test Steps:**
 1. Attempt to delete a product that's already been removed or doesn't exist.
 2. Confirm the deletion.
- **Expected Result:** The system prompts an error indicating the non-existence of the selected product for deletion.

Future Improvements

1. Convert "Grocery Store" actors into two actors: "Owner" and "Employee"
2. Allow "Vendor" to add multiple people (delivery person) under one account
3. Allow "Site Admins" to have different roles (Admin, Editor, Viewer)

References

- [1] [L. X. Lu and J. M. Swaminathan, "Advances in Supply Chain Management," SSRN Electronic Journal, January 2013. DOI: 10.2139/ssrn.2758860.](#)
- [2] [H. A. Khawaja, "What are the problems faced by the retail sector in Pakistan?" Online Blog Article, 2021.](#)
- [3] [G. Ondiek, "Inventory management automation and performance of supermarkets in western Kenya," 2015.](#)
- [4] [A. Sharma, "AI and Robotics for Reducing Waste in the Food Supply Chain: Systematic Literature Review, Theoretical Framework and Research Agenda," 2020.](#)
- [5] [S. K. Panda and S. N. Mohanty, "Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis," 2023.](#)
- [6] [Y. Liu, "Grocery Sales Forecasting," 2022.](#)
- [7] [R. A. Carboneau, R. Vahidov, K. Laframboise, "Machine Learning-Based Demand Forecasting in Supply Chains," 2007.](#)
- [8] [S. Gull, I. Sarwar, Dr. W. Anwar, R. Rashid, "Smart eNose Food Waste Management System," 2021.](#)
- [9] [J. Parfitt, M. Barthel, S. Macnaughton, "Food Waste within Food Supply Chains: Quantification and Potential for Change to 2050," 2010.](#)
- [10] [L. Rieseegger, A. Hübner, "Reducing Food Waste at Retail Stores—An Explorative Study," 2022.](#)
- [11] [A. M. Nascimento, F. S. Meirelles, "Applying Artificial Intelligence to Reduce Food Waste in Small Grocery Stores," 2022.](#)
- [12] [M. Ulrich, H. Jahnke, R. Langrock, R. Pesch, R. Senge, "Distributional regression for demand forecasting in e-grocery," 2021.](#)
- [13] [E. Gul, A. Lim, J. Xu, "Retail Store Layout Optimization for Maximum Product Visibility," 2021.](#)
- [14] [N. Nassibi, H. Fasihuddin, L. Hsairi, "A Proposed Demand Forecasting Model by Using Machine Learning for the Food Industry," 2023.](#)
- [15] [C.-H. Wang, Y.-W. Gu, "Sales Forecasting, Market Analysis, and Performance Assessment for US Retail Firms: A Business Analytics Perspective," 2022.](#)
- [16] [V. Prabhakar, D. Sayiner, U. Chakraborty, T. Nguyen, M. A. Lanham, "Demand forecasting for a large grocery chain in Ecuador."](#)

- [17] [J. H. Friedman](#), “Greedy function approximation: A gradient boosting machine,” *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
- [18] [G. Ke](#), [Q. Meng](#), [T. Finley](#), [T. Wang](#), [W. Chen](#), [W. Ma](#), and [T.-Y. Liu](#), “LightGBM: A highly efficient gradient boosting decision tree,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.
- [19] [T. Chen](#) and [C. Guestrin](#), “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2016, pp. 785–794.
- [20] [A. V. Dorogush](#), [V. Ershov](#), and [A. Gulin](#), “CatBoost: Gradient boosting with categorical features support,” 2018, arXiv:1810.11363.
- [21] [S. Hochreiter](#) and [J. Schmidhuber](#), “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997
- [22] [H. Hewamalage](#), [C. Bergmeir](#), and [K. Bandara](#), “Recurrent neural networks for time series forecasting: Current status and future directions,” *Int. J. Forecasting*, vol. 37, no. 1, pp. 388–427, 2021.
- [23] [H. Xu](#) and [C. Y. Wang](#), “Demand prediction of chain supermarkets based on LSTM neural network,” *China Logistics Purchasing*, vol. 3, pp. 42–43, 2021
- [24] [J. Kim](#) and [N. Moon](#), “BiLSTM model based on multivariate time series data in multiple field for forecasting trading area,” *J. Ambient Intell. Hum. Comput.*, pp. 1–10, Jul. 2019
- [25] [G. Kechyn](#), [L. Yu](#), [Y. Zang](#), [S. Kechyn](#), "Sales Forecasting using WaveNet within the Framework of the Kaggle Competition," arXiv:1803.04037v1 [cs.LG], Mar. 11, 2018.

Appendix

Interview Transcripts

Z Mart

Q.1: How many of the items are imported?

A: Very less, out of 100 percent only 20 percent of the items are imported.

Q.2: What do you think can be solved if your work gets digitised?

A: It's easier to work using the software systems as they can scan items, 2 hours of work gets done in 15 minutes.

Q.3 Are there any issues you face with the software?

A: Not really. Nowadays, iPos software is being used in markets. If we encounter any issue, we call them and they fix it.

Q.4: Have you faced any issues with fake products?

A: Yes, Mobilers. Companies like Shangrila, Lipton, Supreme and Delmonte, their sales come from the company itself. For other companies, wholesalers in Shah Almi get their stuff from the companies in large quantities and get a discount. There are a few things in them that are fake, usually 'bubbles', 'tulsi', 'bombay' etc.

Q.5: How do you place your orders, do you have automatic systems or do you see for yourself if the items are less?

A: We observe how many items get sold in a day and for the next few days we buy the same amount.

Q.6: How does the order for the imported items get placed?

A: We get imported items for less quantities so they do not get wasted. There is 'Monthly Auditing' done too. If a certain item does not get sold before the expiry then we decrease the quantity for the next time. If the items have 3-4 months until their expiry, we sell them for discounted prices. Quantity of canned foods with a shelf life of 2-3 years is large therefore we put it on discount so it gets sold and fresh items get placed and for the next time we order a lesser quantity. We guess the quantity of items on a daily basis, our ordering is based on the sales of items from the previous day mostly.

Q.7: How does the ordering differ on special occasions?

A: On special occasions like eid, we buy more items than usual. For example, if we buy 4-5 breads each day, we would buy 20-25 breads on these occasions.

Q.8: How often do you update inventory?

A: Whatever comes from the company (vendor) gets added to the inventory first and expired items are removed from the inventory.

Q.9: Is there something that takes a lot of time to do that needs to be improved?

A: The iPos system does everything but sometimes if we forget to add something, something gets stolen, or 20 items are on the shelf but 24 are in the inventory, these types of issues do occur so we have to manually audit them.

Q.10: If you want to sell some new product, how do you find it?

A: The companies (vendors) ask us to display the new products and if it gets sold then they get paid. Otherwise, if the products do not get sold they take it back.

Express Store

Q.1: How do you run the grocery store and what significant issues do you confront?

A: People have expectations, therefore when imports ceased and demand for a product called stevia increased, we requested sellers (vendors) for it so we could continue to sell it. People once assumed that our store would carry items that were unavailable at other supermarkets, but they no longer do so because imports are either scarce or non-existent.

Q.2: How do you verify identification of personnel delivering your orders?

A: It's challenging to verify the authenticity because there have been, and still are, instances where individuals falsely claim to represent these organisations, selling stock with products nearing their expiry dates. Within the community, these individuals are commonly referred to as 'mobilers.'

Q.3: How do you manage imported goods?

A: Imported items have dealers here as well and they keep a follow-up as well. Nowadays there's an issue with the budget, since inflation the stock purchasing has decreased significantly, assuming that before 10 items were being purchased now just 3-4 items are being purchased. Then there are also a few things that get restricted by some stores just for themselves.

Q.4: Are there any issues regarding the current software that is being used in the store?

A: If electricity goes away then we are unable to use the software therefore we have to write down the purchases on a piece of paper which sometimes could get lost and then later we have to manually enter the data. There is no backup, for 8 hours there is no electricity, and sometimes if someone (buyer) is in a hurry then nothing gets entered into the system.