

LAUNDRY ORDERING APPLICATION

Shuceyb Ali Barise

Abukar Mohamud Elmi

Abdirahim Ahmed Abukar

Abdikani Mohamed Hassan



**SUBMISSION OF GRADUATION PROJECT FOR PARTIAL
FULFILLMENT OF THE DEGREE OF BACHELOR OF SCIENCE IN
COMPUTER APPLICATION**

**JAMHURIYA UNIVERSITY OF SCIENCE AND TECHNOLOGY
(JUST)**

**FACULTY OF COMPUTER APPLICATION AND INFORMATION
TECHNOLOGY**

August 2023

Original Literary Work Declaration

NO:	Name of Candidates	ID No
1	Abukar Mohamud Elmi	C119926
2	Shuceyb Ali Barise	C119942
3	Abdirahim Ahmed Abukar	C119289
4	Abdikani Mohamed Hassan	C119021

Name of Degree: **Bachelor of Computer Application**

Title of Project Paper/Research Report/Dissertation/Thesis (“this work”);

Laundry Ordering Application

Field of Study: **Computer Applications**

We, the undersigned, solemnly and sincerely declare that:

Candidate 1’s Signature _____ Candidate 2’s Signature _____

Candidate 3’s Signature _____ Candidate 4’s Signature _____

Date: _____

Subscribed and solemnly declared before,

Supervisor’s Signature: _____ Date: _____

Name: _____ Designation: _____

Abstract

The Laundry Order App is a mobile-friendly application designed to streamline the process of scheduling and managing laundry services for customers. With the ever-increasing demand for convenient services in our fast-paced society, this app aims to provide an efficient and seamless solution for individuals looking for laundry-free assistance. Customers can define their laundry preferences, such as washing instructions, detergent preferences, and special handling requirements, ensuring their status such as ongoing and completion. To make technology more age and current situation friendly, it is important to understand the advantages that disabled persons and senior citizens perceive in using it. To assist laundry ordering with users placing their own booking without any problems. Overall, the Laundry Ordering App offers a modern and efficient approach to laundry management, saving customers valuable time and effort. By connecting users with trusted laundry service providers, it contributes to a more seamless and convenient laundry experience in today's fast-paced world.

Dedication

We are happy to thank our parents for all the help they have given us and for their sacrifice to support us to reach this great milestone especially the completion of this project, In addition to all our relatives and everyone who assisted us to succeed in this project, we would like to thank you all.

Acknowledgment

First we would like to thank **Allah** the creator of this universe for giving us good health and well-being to complete this long journey as well as coordinating this book.

Secondly we are thankful to our mother and father who have been the backbone of our lives especially the foundation of our education, all our brothers and sisters.

Thirdly we would like to express our sincere gratitude our teacher and supervisor **Eng.** Mohamed Abdi Karin for his constant encouragement, guidance, and commitment to the success of our research and we would like to than **Eng.** Abdunur Abdulkadir Hashi as an assist supervisor. We would also like to thank **Eng.** Sharma'arke Ali Kahie, Head of Research at the Faculty of Computer Science and IT, has always been very supportive and has provided useful advice. We would also like to thank the honorable

UNIVERSITY OF SCIENCE AND TECHNOLOGY. (J U S T) Who gave us excellent opportunity to finish our bachelor degree of computer application, it is chance to take this opportunity to thank all the lectures of the faculty of computer application and IT for their loyalty and guidance toward the bright future, especially the dean of faculty **Eng.** Mohamed Abdullahi Ali. Finally, we would like to deliver our special thanks to all those who have helped us, both directly and indirectly. All praise is due to **Allah**.

Abbreviations

ABSs	Alkyl Benzene Sulphonates
APIs	Application programming interface
BIG-SF	Business Internet Group San Francisco
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart.
CBD	Central Business District
CSS	Cascading Style Sheets
E-business	electronic business,
E-records	electronic records
GCP	Google Cloud Platform
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
LMS	laundry management system
MCC	Mobile Cloud Computing
PHP	Hypertext Preprocessor
SSL	Secure Sockets Layer
UI	User Interface
UML	Unified Modeling Language
Wi-Fi	Wireless Fidelity

CONTENTS

Original Literary Work Declaration.....	ii
Abstract	iii
Dedication	iv
Acknowledgment	v
Abbreviations	vi
1.0 INTRODUCTION	1
1.1 Background of the study	2
1.2 Statement of the problem	4
1.3 Motivation of the study	4
1.4 Objective of the study	4
General objective.....	4
Specific objectives.....	4
1.5 Research of the questions.....	5
1.6 Scope of the study	5
1.7 Significance of the study.....	5
1.8 Structure of the thesis.....	6
CHAPTER II: LITERATURE REVIEW	7
2.0 Introduction.....	7
2.1 History of Laundry	7
2.2 Types of laundries	9
2.2.1 Industrial Laundry	9
2.2.2 Commercial laundry service	10
2.2.2.1 Non-self-service laundry.....	10
2.2.2.2 Self-service laundry	10
2.2.2.3 Pickup and delivery services.....	11
2.3 Mobile application	12
2.4 Impact of laundry system	14
2.4.1 Environmental impact	14
2.4.2 Technological impact	15
2.5 Advantages of laundry system.....	15
2.6 Scheduling and door-step servicing through the mobile app.	16
2.7 Related work	18

CHAPTER III: METHODOLOGY	22
3.0 Introduction of laundry ordering app	22
3.1 overview of laundry ordering app	22
3.2 system features of laundry ordering app	23
3.3 software requirements	24
3.4 Hardware requirements	25
3.5 user requirement of laundry ordering app	25
CHAPTER IV: SYSTEM ANALYSIS AND DESIGN	27
4.0 System analyses	27
4.1 Existing system/approach	27
4.2 Proposed system.....	28
4.3 System requirement:	29
4.3.1 Functional Requirements	29
4.3.2 Non-functional requirement	29
4.4 Feasibility Study	30
4.4.1. Schedule feasibility	30
4.4.2 Operational feasibility	31
4.4.3 Technical feasibility	31
4.5 System Design.....	32
4.5.1 System Maintenance	32
4.6 System Architecture	33
4.6.1 Architecture Diagram.....	33
4.7 UML Diagram Types	33
4.7.1 Use case Diagram.....	34
4.7.2 Activity Diagram.....	35
4.8 Database Design.....	36
4.9 System algorithms	36
CHAPTER V: IMPLEMENTATION & TESTING	37
5.0 Introduction	37
5.1 overview	37
5.2 Snapshots of the System	37
5.2.1 Login screen	37
5.2.2 Register user screen	39
5.2.3 Cart Screen.....	42

5.2.4	Customer feedback.....	43
CHAPTER VI: DISCUSSION OF RESULTS		45
6.0	Introduction:.....	45
6.1	Benefits of Laundry Order Service Apps:.....	45
6.2.1	Convenience and Time-Saving:	45
6.2	Customization and Personalization:	45
6.2.2	Transparent Tracking and Updates:	45
6.2.3	Payment Options and clarity:.....	45
6.3	Challenges and Areas for Improvement:	46
6.2.4	Delivery Time Accuracy	46
6.2.5	Potential Clothing Damage	46
6.2.6	Privacy and Security	46
6.2.7	Handling Special Garments or Requirements	46
6.2.8	User Feedback and Issue Resolution:	46
7.0	Introduction	47
REFERENCES		49
Appendix A // Connection Firebase.....		52
Appendix B // Signup User		55
Appendix C // Login Screen.....		65
Appendix D // Forgot Password.....		75
Appendix E // Cart Items Screen.....		79

List of figure

Figure	Page
Figure 2. 1 Commercial laundry service 1	10
Figure 2. 2 self-service laundry 1.....	11
Figure 2. 3: Wash-house homepage2 1	12
Figure 2. 4 : Laundry Mama	18
Figure 2. 5 Just Clean.....	19
Figure 2. 6 Maurice Dry Cleaners.....	20
Figure 2. 7:LaunApp mobile application	21
figure 4. 1:Architecture diagram level	33
figure 4. 2:Use Case Diagram	34
figure 4. 3:Activity Diagram 1	35
figure 4. 4:Firebase Collections	36
Figure 5. 1: Login screen	38
Figure 5. 2:Signup Screen.....	39
Figure 5. 3:Home page Screen.....	40
Figure 5. 4:Order Screen.....	42
Figure 5. 5:Cart Screen	43
Figure 5. 6:feedback screen.....	44

CHAPTER I: INTRODUCTION

1.0 INTRODUCTION

Laundry is a basic necessity of life that requires a significant amount of time and effort. Overall, this Chapter aims to provide a comprehensive overview of the laundry ordering app industry and its potential for growth in the future. By doing so, it will provide valuable insights for individuals and businesses interested in the laundry service industry and the technology that is transforming it. Laundry also easy knows that potential customers want to order laundry service. In addition prospective customers will also be more interested in knowing laundry can be reached and can provide pickup and delivery service. (Jamgade et al., 2016).

As everyone is busy with their professional life, getting online laundry service at their doorstep makes life hassle free. Clean clothes, enhances personality of person. The secret to clean clothes is simple; to take advantages of an online laundry service at your doorstep, which takes care of your garment with soft and Eco friendly detergents, which gives your clothes a brand new look along with freshness to them. Professionals, who are running their business or are doing corporate jobs, have a time crunch always. For them a laundry service is bliss. It would be convenient, fast, and easy to get connected with. Doorstep laundry services provide garments washed, ironed and neatly delivered in a convenient bag (Olawar et al., 2017).

A significant part of the operation of any laundry firm involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves; customer personal information and clothing records history, user information, price of delivery and retrieval period, users scheduling as regards customers details and dealings in service rendered, also our products package waiting list. All of this information must be managed in an efficient and cost wise fashion so that the organization resources may be effectively utilized (Shoewu et al., 2016)

In existing days, laundry service is time consuming Process. Customers have to often visit Laundry service Centre, after the order is placed they have to collect their cloth by their self and they don't get any notification about completion of work. Although in some of the laundry Centre there is probability that they will serve good service to customer and will

deliver the order at home but there may be possibilities that customer will not be present at home, so the customer in turn miss his delivery (Einav et al., 2014).

The laundry ordering system permits business owners to trail large amounts of specific laundry items faster with high efficiency. By automating the business procedures, one can deal with one's time better and make enlightened business decisions. By using the laundry management system, it is easy to automatically keep tabs on clothing, linens, rags, and other resources, automating the inventory process and accounting (Adekola, et el 2021.)

1.1 Background of the study

Every day we will change the clothes we wear. Clothes that are dirty will be washed. Laundry is a daily and time-consuming home duty. However, research has seldom investigated the laundry sector, and the means through which its business processes have been redesigned via digital applications. Developing a mobile application for bringing laundry services to a customer's doorstep will be effective and beneficial for both customers and services provider (Priambodo, B. and Ani , 2016).

The study's objectives are to determine whether the organization level influenced e-records implementation, whether the technology level influenced e-records implementation, whether the environment level influenced e-records implementation, and whether adoption of e-records implementation influenced the organization, technology, and environment. As in the current scenario every business opportunity is getting digitalized wherein it has been easy access to anybody from any part of the world to connect and organize any form of services. These systems are the pillars providing a company its much needed information and compiling their day to day services (Soni, 2020).

People nowadays do not have sufficient time to complete home duties after work. The development of information technology and mobile technologies has led to the development of electronic business (E-business), and has extended such business's competitive advantages. Organizations intend to redesign their ordinary processes, and to change their business processes, in order to remain competitive, to obtain success, and to keep up with technology innovations. Therefore many business sectors have developed mobile applications designed to market their business services and goods (Bamasoud et al., 2018).

The previous systems in our indigenous environment require numerous paper forms, with data stores spread all over the laundry management infrastructure. Information in this type of system is often incomplete, inaccurate, lost in transit during computation and sometimes duplicated. Therefore, the business workflow is at low ebb. problems such as mix-up with customer clothing, loss of customer clothing, late delivery, inability to account for certain customer items, inefficient organization methods (cataloging), insufficient collation of reports for managerial or company use, To attend to these issues, (Adekola et al., 2021).

As a records management tool, many organizations have chosen to implement E-Records. E-Records has changed the way organizations interact with their records. The purpose of implementing E-Records is to obtain an accurate and easy to access information anywhere as well as to reduce the risk of information loss. In addition, this E-Records can improve employment services and increase efficiency in records management. To identify relevant information, human intervention is still needed to handle these records because the nature of the information and records stored in the E-Records. (Norolazmi et al., 2022)

There are several constraints that occur while developing this system which among them are the authors had to change the initial system proposals that had been made because the flow of the system was difficult to understand and somewhat complicated to build. So it took a few days to get the idea for a new system that is more organized and easy to understand by users and authors. In addition, authors who had different time schedules made group discussions a little difficult to do. So sometimes there is a bit of a misunderstanding between the authors. In the future, this system can be further improved by adding features of a laundry pick -up service which allows users to use the service so that they do not have to go to the laundry to send clothes to be washed (Omar et al., 2022).

The demand for application-based laundry services have triggered the customers to register and receive their laundry services with simple clicks. This insisted building a mobile application for laundry services. Also, the need for cross-platform enabled application for providing the services have insisted the use of Flutter. Developers of mobile apps prefer Flutter because the applications developed using Flutter provides a near native performance. This focuses on establishing a healthy connection between the customers and the laundry service providers to get mutually benefitted, in provisioning and availing the on-demand laundry services respectively. (Gupta et al., 2018)

1.2 Statement of the problem

A laundry firm currently uses a manual system for the management and maintenance of critical information. The current system requires numerous paper forms, with data stores spread throughout the laundry firms' management infrastructure. Often, information (in written form) is incomplete or does not follow management standards. Forms are often lost in transit between departments, requiring a comprehensive auditing process to ensure that vital information is not lost. This has resulted in inconsistencies in various data due to the large volume of contrasting customer details, resulting in clothing mix-ups in laundry firms, resulting in delays in collecting the clothes back. It is also useful for washing clothes for people who don't have time to wash their clothes specially laundry.

1.3 Motivation of the study

Laundry ordering app could be driven by the increasing demand for convenient and efficient laundry services in today's fast-paced world. With the rise of digital technology and the growing preference for online transactions, there is a need for laundry business to adapt to change consumer behavior and preferences.

1.4 Objective of the study

General objective

To design and run as a mobile laundry application, which requires an internet connection, and provides a unified platform for people to get their cleaned clothes.

Specific objectives

1. To provides auto calculation of the payment.
2. To provide a fast and comfortable service for customers by considering the factors like process, performance, and delivery.
3. To Scheduled and door-step servicing through the mobile app.

1.5 Research of the questions

- How to provide auto calculation of the payment?
- How to optimize business processes by offering them in the form of a mobile application that handles the activities step by step?
- How to provide a fast and comfortable service for customers by considering the factors like process, performance, and delivery?

1.6 Scope of the study

This project scope is divided into three which are known as the customers, staff, admin and the system.

Customer

- i. Customers need to speak their and what services they want.
- ii. Customers can check their status by mentioning the reference number.

Staff

- i. Staff will collect and check the data.
- ii. Staff also can change the status when done

Admin

- i. Admin will add, delete, or update services and staff
- ii. Admin can view the report.

1.7 Significance of the study

- For individual a laundry mobile app can help users to book their laundry orders online, on the tap on a few buttons.
- For business mobile application also enables laundry businesses to automate their whole process (booking, payment, pickup & delivery) and help them work more efficiently on a large scale too.

- Moreover, studying a laundry app can provide insights into the broader trend of digitization and automation of everyday tasks, and how these technologies can improve and business models in the laundry industry, which can lead to job creation and economic growth.

1.8 Structure of the thesis

This study will cover six main chapters.

Chapter One: Introduction this chapter we will talk about introduction, background study, problem statement, objectives, research question, significant of the study, scope of the study, Structure of the study, motivation of the study.

Chapter Two: Literature Review This chapter reviews previous literature and research related to your field of study. Literature reviews are comprehensive and should include recent publications.

Chapter Three: Methodology This chapter describes the research methods used in the study. Sub-topics in this chapter include key survey questions, survey design, and accepted survey procedures.

Chapter Four: System Analysis and Design: this chapter is talking about system analyses and designing of the system.

Chapter Five: Implementation and Testing: this chapter is also talk about implementation and testing of the system

Chapter Six: Conclusion and Recommendation: this is the final chapter of the research explains conclusion of the research, discussion and recommendation.

CHAPTER II: LITERATURE REVIEW

2.0 Introduction

Laundry is the washing of clothing and linens. Laundry is a basic necessity of life that requires a significant amount of time and effort. While some people enjoy doing laundry, others find it tedious and time-consuming. Additionally, with the busy lifestyles of today's society, finding the time to do laundry can be challenging. Fortunately, technology has made it possible to simplify the process of doing laundry through the use of laundry ordering apps. The evolution of a technology-focused life has helped solve most problems faced by individuals and has harnessed technology to help serve them and accomplish their goals. Our research in this chapter emphasized the importance of service innovations to customers, by including an addition to the current service mix, or by changing the existing services offered. Laundry is a daily and time-consuming home duty. However, research has seldom investigated the laundry sector, and the means through which its business processes have been redesigned via digital applications. Developing a mobile application for bringing laundry services to a customer's doorstep will be effective and beneficial for both customers and services provider (Priambodo et al ,2016).

An essential part of many industries, clean and hygienic laundry services help businesses to improve quality, control costs, and provide customer satisfaction. Industrial laundry operations vary in size and scope, and are common in the hospitality, leisure, and healthcare industries. Other industries that use laundry services can be listed as sea travel for leisure and military purposes, fire departments, airlines, food industries, sport centers, schools, and apartments. Laundry services exhibit similar characteristics of businesses operating in complex, heterogeneous and dynamic environments. One of the primary purposes of a laundry management system (LMS) is to organize and monitor different laundry processes at various levels, to ensure accurate and correct laundry activities that increase in productivity, quality, customer loyalty, and satisfaction (Celikkan & Kurtel, 2017).

2.1 History of Laundry

Before the 1900s, people used to wash clothes with bare hands requiring more hard work and energy, which was cumbersome and time consuming. People experienced it as a burden. They brought their laundry to lakes and rivers to wash and used rocks

or boards as scrubbing surfaces. Women used rocks to beat the wet clothes as that was the best way to remove dirt and stains from the clothes. The process of drying was done in the open in the sunshine. The sun was also utilized as to bleach the whitening of fabrics. Removing of stains and dirt from clothes required lots of energy. This was a hectic and tiring task for the people. Later on, for a few years large hand-turned drums came into existence in every household. Though it required muscle power, many clothes could be washed and cleansed at the same time instead of washing only one garment at a time (Upadhyaya & Kavatekar, n.d.)

In the mid-1960s concern arose over the environmental fate of complex phosphates due to their implication in eutrophication of waterways. Since then, the detergent industry has devoted considerable energy to finding cost effective replacement. The surfactants used in early synthetic detergents were prepared by reacting benzene with propylene tetramer to form the alkyl aryl group, which was then sulfonated. These materials are so called hard ABSs were highly branched and non-biodegradable. Overtime they accumulated in the environment to such an extent that foaming resulted in some sewage treatment plants and waterways. In 1965, the U. S. detergent industry voluntarily withdrew hard ABSs from the market, replacing them with biodegradable linear chain analogs. In the mid in 1970s, the introduction of ion exchange materials and zeolites as detergents builders led to a gradual movement away from phosphate technology. (Bajpai & Tyagi, 2007)

The study of Australian women in unpaid and paid work has changed over the last thirty years. In 1975, Kingston examined, for the first time, the experiences and value to society of women working in their own homes. *Women at Work*, a project by a collective of writers, examined individuals and groups of women in their endeavors to succeed in leadership and at work. They placed women in a broad context of work in both the private and public spheres.⁸ In the same year, Ryan and Conlon focused their research on women in the paid workforce. They challenged other researchers to follow them and explore the position of working women. They wrote, "This book may be regarded as a primer which we hope will inspire others to enlarge upon and to dig deeper into the vast territory left to be explored. They succeeded. Since then, the body of literature written on women and work has expanded greatly. Gender must always be at the core of all debate on the role of women in society

because, as Game and Pringle (1983) explained, 'Gender is fundamental to the way work is organized' (TeachCert, n.d. 1998)

In 1982, the contributors to Worth her salt broadened the scope of studies and recognized that social change shaped women's position in society within the parameters of the gender division of labor. Pringle (1988) showed how the gender division of labor was accentuated by computers and information technology. Yet O'Donnell and Hall (1988) and then Probert (1989) argued that, despite being better educated, with wider employment opportunities and greater economic freedom, women were still trapped in the world of economic, political and sexual inequality. Alongside the general analyses of women and work have been critiques of specific influences. Many writers have criticized the system of arbitration and unions because they have failed to protect the lowest paid - female - workers. The arbitration system's reinforcement of the 'male breadwinner' concept in wage indexations has been held responsible for inequalities in the labor market, and the lack of female protection and women's subsequent hardship. Gill failed to address the gender inequity of the basic wage when he defended the arbitration court's indexation of the basic wage movement to inflation and consumer prices (Connell RW, et al, 1991).

2.2 Types of laundries

There are several types of laundry available in the world. Commonly used systems are manual only. Therefore, user data is difficult to record. Laundry is categorized into different type due to their size, scale, products offered, Store Format and Trends While people use the terms "Industrial", "Commercial" and "Residential" interchangeably to refer to laundry services, industry watchers offer more specific guidelines about different types of Laundry. "Industrial type" is on the larger end of this spectrum and carry a diverse mix of machines and general merchandise. Nomenclature is not always uniform Financial Institutions Fund places Wal-Mart in the same category as supermarkets and as well running laundry services. (Ibrahim, n.d.)

2.2.1 Industrial Laundry

Industrial laundry this type is for the big guys. Usually utilizes the use of a tunnel washer and/or a heavy duty front load washer with big capacity, about 50kgs up per machine. It would require big investment. Clients to look for if you have this type of

laundry would be institutional like hospital, hotel, and motel. Spa and etc. (Yung et,el., 2006)

2.2.2 Commercial laundry service

There are upwards of 200,000 commercial laundry facilities in the United States today. These can be freestanding laundry plants, located in hotels, hospitals, nursing homes, health clubs, resorts, as well as in penal institutions, athletic facilities, in coin-operated laundromats, and in dry-cleaning establishments. These locations are using, in toto, tens of billions of gallons of water and massive amounts of energy each year. (Rice et al., 2009)

2.2.2.1 Non-self-service laundry

The first system is a non-self-service laundry system. It helps to wash the user's clothes by following the store's procedures. Users only need to send and leave their clothes in the store. However, users will never know the current status of their clothes until they are called by the laundry worker. (Omar et al., 2022)



Figure 2. 1 Commercial laundry service 1

2.2.2.2 Self-service laundry

A self-service laundry, coin laundry, or coin wash, is a facility where clothes are washed and dried without much personalized professional help. They are known in the United Kingdom as launderettes or launderettes, and in the United States, Canada, Australia and New Zealand as laundromats. Users need to use coins or

tokens provided by the laundry to start the washer and dryer. The user has to wait until the washing and drying process is over. It takes a long time and is a lot of time. Laundromats are an essential business in urban communities. Laundromat owners may employ someone to oversee and maintain the general laundromat throughout the day. Some laundries employ staff to provide service for the customers. Minimal service centers may simply provide an attendant behind a counter to provide change, sell laundry detergent, and watch unattended machines for potential theft of clothing. If the business is big enough, the owner may employ a plumber to constantly maintain the machines and other workings. Others allow customers to drop off clothing to be washed, dried, and folded. This is often referred to as fluff & fold, wash-n-fold, drop off, bachelor bundles, a service wash or full-service wash. Some staffed laundry facilities also provide dry cleaning pick-up and drop-off (According to Wikipedia, 22 April 2012.).



Figure 2. 2 self-service laundry 1

2.2.2.3 Pickup and delivery services

A laundry service is a convenient way to wash clothes without leaving home. You simply schedule a pickup, and someone will come to your house to pick up your dirty clothes. They will then take them to the service and return them to your doorstep once they are clean and folded.

WashHouse

The third system is Washhouse. This system's functions are provide pickup and delivery services. Users can choose the time they want. It only covers the area in Kuala Lumpur and Petaling. However, everybody can access this website <https://www.washhouse.my/> which means the security of this system might look a bit low. (Omar et al., 2022)

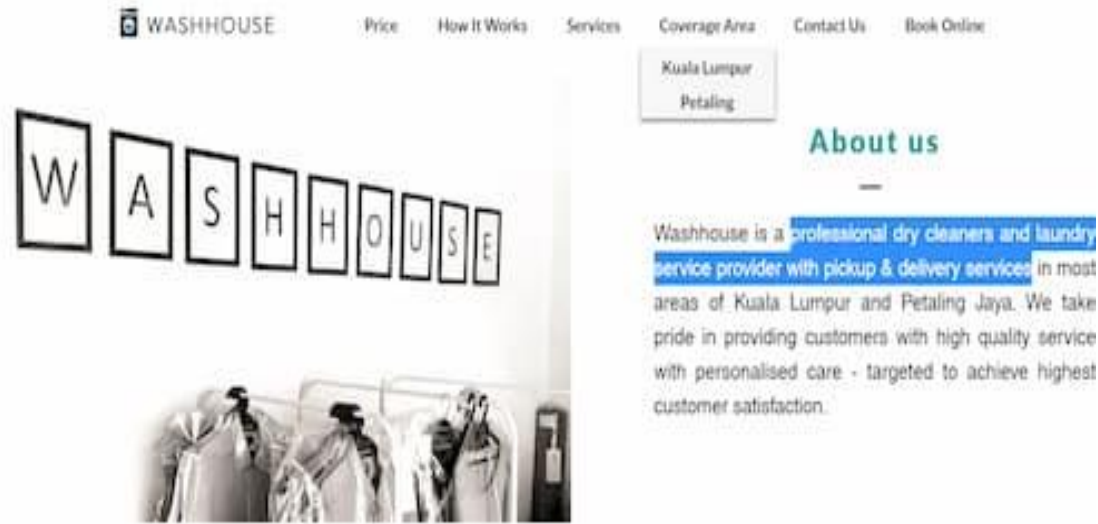


Figure 2. 3: Wash-house homepage2 1

2.3 Mobile application

Mobile applications are consist of software/set of program that runs on a mobile device and perform certain tasks for the user. Mobile application is a new and fast developing Segment of the global Information and Communication Technology. Mobile application is easy, user friendly, inexpensive, downloadable and run able in most of the mobile phone including inexpensive and entry level phone. The mobile application has wide uses for its vast functioning area like calling, messaging, browsing, chatting, social network communication, audio, video, game etc. There is a global positive impact of mobile application. Using mobile application developed country are becoming facilitate and people, society of developing country are upgrading themselves and making a new type of IT infrastructure. Mobile applications are running on a small hand hold mobile device which is moveable, easy to use and accessible from anywhere and any place. Now a day, so many people are using mobile

application to contact friends, browse internet, file content management, document creating and handling, entertainment etc (Islam et al., 2010).

2.3.1 MOBILE APPS AND PLATFORMS

2.3.1.1 Native applications

Native apps are developed using an Integrated Development Environment that provides the necessary development tools for building and debugging the applications. Native applications refer to applications that are specifically written and developed for a specific mobile operating system. The three leading mobile operating systems are Google's Android, Apple's iOS, and Windows Phone. In order to create true, native applications, the Java programming language must be used for Android, the Objective C programming language for iOS, and the .NET framework for Windows Phone. Common, key characteristics of native applications are that these applications have unhindered access to device hardware and support all user interface and interactions available in the respective mobile operating environment (Jobe, 2013).

2.3.1.2 Web or connected applications (Web Apps)

The web has had a significant impact on all aspects of our society, from business, education, government, entertainment sectors, industry, to our personal lives. The main advantages of adopting the web for developing software products include (1) no installation costs, (2) automatic upgrade with new features for all users, (3) Universal access from any machine connected to the Internet and (4) being independent of the operating system of clients. On the downside, the use of server and browser technologies make web applications particularly error-prone and challenging to test, causing serious dependability threats. A 2003 study conducted by the Business Internet Group San Francisco (BIG-SF, 2003) reported that approximately 70% of websites and web applications contain defects. In addition to financial costs, defects in web applications result in loss of revenue and credibility. The difficulty in testing web applications is many-fold. First, web applications are

distributed through a client/server architecture, with (asynchronous) HTTP request/response calls to synchronize the application state. Second, they are heterogeneous, i.e., web applications are developed using different programming languages, for instance, HTML, CSS, JavaScript on the client-side and PHP, Ruby, Java on the server-side. And third, web applications have a dynamic nature; in many scenarios they also possess nondeterministic characteristics (Doğan et al., 2014).

2.3.1.3 Hybrid or synchronized applications

Hybrid apps try to combine the advantages of web and native apps. Hybrid apps are primarily built using HTML5 and JavaScript, and a detailed knowledge of the target platform is not required. Hybrid apps embed HTML5 apps inside a thin native container (UIWebView in iOS and Web View in Android). Like web apps, the source code is still executed by a browser that is part of the final application and can be packaged with the application unlike web apps where the source code is downloaded from the web. Hybrid apps are installed on the device and access to the underlying device hardware and data is feasible through specialized APIs. An example of the most popular container for creating hybrid mobile apps is Phone Gap. The code implementation of the hybrid apps can be done using various technologies and development platforms but in order to achieve a native look and feel it is necessary to use specific development libraries, such as JQuery. (Xanthopoulos & Xinogalos, 2013)

2.4 Impact of laundry system

The impact of laundry system have a significant impact on our daily lives and the environment and Technology.

2.4.1 Environmental impact

Laundry detergent is widely used a household chemical product all over the world. In Mogadishu this is also popular for washing and cleaning clothes. The production of laundry detergent comprises of the combination of various chemicals. There are some common chemicals required for the production of laundry detergents. Besides these common chemicals, different industries use different types of other chemicals which are responsible for the comparative quality of detergents produced by various

companies. Production of detergents includes various chemicals; of them some are costly and need to be used in right proportion for the effective performance. Some of these costly chemicals are specially imported in Mogadishu because they are not available in the local chemical market (Afreen, 2012).

Impact of detergents – detergents have to be considered in assessment of environmental impact because they may become part of waste water. Unless the released waste water is properly processed, it may contain harmful substances, such as ammonia and phosphorus. These substances are toxic not only to plants and animals but also to humans. In addition to release of water with toxic substances, biological waste is also discharged and it may be a significant source of bacteria. Optimization of detergents dosing is a key method for elimination of impact of the laundry on environment. Reduction of detergents concentration at the source directly reduces difficulties in subsequent purification of waste water. Besides this measure, purified waste water containing detergents may also be reused in a bath (Hoinkis et al. 2008).

2.4.2 Technological impact

People have started to use technology to serve themselves. Technology has invaded all aspects of our lives and has become substantial for everyone. Therefore technology has greatly influenced people's lifestyles. Presently, great progress has been witnessed in smartphone technology, and smartphones are no longer just a means of voice communication. Rather, they provide various capabilities which help transform traditional services into electronic ones. With the rapid development of the internet and mobile technologies, these technologies have led to the development of mobile business, which is becoming increasingly innovative and diverse (Bamasoud et al., 2018)

2.5 Advantages of laundry system

The laundry service industry is widespread in big cities in Indonesia, such as Jakarta. The reason for using the laundry service was because of the busy schedule of the customers, which made them not have time to clean their own clothes. Therefore, urban customers often look for laundry services to help them clean their clothes in the midst of their busy lives (Hussein et al., 2014 as cited by Jack, 2013).

Chairman of the Indonesian Migrant Workers Association, Wasono Raharjo said if you want to start a laundry business with a small capital, laundry-kilo is the right choice. With a capital of about 15 million up to 20 million rupiahs, entrepreneurs can build laundry service business in strategic locations such as housing, campus or boarding house (Novita Intan Sari, 2015 as cited by Jack, 2013)

In the situations described above, the laundry service industry requires adequate electronic laundry service applications. Electronic laundry service applications can be used to simplify and speed up administrative work to be more effective and efficient. Electronic laundry service applications that will be implemented in the laundry service industry can also be used to manage all necessary data, ranging from customer registration data to managing customer data, raw material data to be used, supplier data, and all other operational data used by the laundry service industry (Rukmanasari, E.Y. 2013).

2.6 Scheduling and door-step servicing through the mobile app.

Laundry ordering apps are mobile applications that allow users to order laundry services from their smartphones or other mobile devices. These apps provide a convenient way for people to get their laundry done without the hassle of going to a physical laundromat or hiring a personal laundry service. The laundry ordering app industry has grown significantly in recent years, with a wide range of apps available for download on various app stores. The ability of users to organize events using mobile devices is a defining characteristic of today's social network systems. With the advancement of mobile technology, more and more people are digitally connected, which makes the analysis of group event planning process and decision making for event organizers critically important. There is a rich history of UbiComp research concentrating on individual user behavior analysis, which treats individual user's data as a singleton. However, social interactions among group members are often ignored, and relatively scant research to date has explored the subject of group event scheduling. Some early work has been confined to in-lab surveys and has not studied the real-world event scheduling process by groups of users, nor the factors that would impact group event decision-making. More recently, a study of university groups using mobile phones was presented (Kasthuri Jayarajah et al 2015).

What has been missing to date is a detailed understanding of the process of how groups make a decision to visit a particular place at a particular time using their mobile devices. What factors influence a group's final decision? This paper provides detailed novel insights in the event scheduling process of social groups. We believe that this is an exciting area ripe for exploration by the ubiquitous computing research community. With ever-increasing popularity of smartphones, we expect that mobile computing will be used extensively to assist groups of people in event planning in terms of when and where to rendezvous. Consider a group of friends out on a weekend evening trying to decide what movie to see or where to eat, or consider a group of professional colleagues trying to decide where to go for lunch. Given the frequency with which people schedule collocated events, we believe mobile applications for group event scheduling can provide significant help. However, despite this considerable potential, today's technology offers limited help when it comes to coordinating group events in online and offline scenarios. Currently there are few group event organization applications on the market. The most commonly used services are Meetup, Facebook Events and Evite (Alon Elgar et al. 2004).

Modern mobile devices continuously generate and process sensory data about the surrounding environment, and hence enable users' contextual awareness. Due to the computational intensity of processing such sensory data which could be audio, images, or biometric measurements, mobile devices tend to be operated in a MCC environment, in which the local computation workloads are offloaded to the remote cloud via cellular networks. The involvement of mobile clouds hence augments local devices' capabilities and prolongs their battery lifetime. (Tong & Gao, 2016)

Doodle is an online event scheduling service which supports groups in finding a mutually agreeable meeting time. Participants are able to vote for their time preferences. But all the meeting time options are pre-selected by the host. Group members have no permission to suggest new options. In addition, group members cannot suggest or vote for meeting locations. To address the limitations of existing services, we developed Out WithFriendz, a mobile application that enables groups of people to decide together through a voting process the date/time the group would like to meet as well as the location where they would like to meet. The app will be implemented as a client-server architecture that is comprised of both Android based clients that communicate with a server implemented as a Java Web

application. The main elements of our mobile application to start using it, a user may create a new invitation acting as a host. During this process, she can specify the details of this invitation including a title, a list of suggested dates, a list of suggested locations and invited participants. After this host submits a new invitation to the server, all invited participants receive it and can view the detailed invitation was found (Doodle 2017)

2.7 Related work

Among the applications similar to our application the following:

Laundry Mama: Laundry Management System and Laundry-On-Demand Mobile Applications are presented in this paper. Using conventional laundry service method, customer is not informed about the laundry process stage, does not have option to arrange the preferred laundry pick up time for the deliveryman to pick up the unwashed laundry from the address provided by customer and the laundry ordering paper forms are often lost in transit between customer and admin. Therefore, a laundry management system software and laundry on demand mobile application is demanded to solve the problems. The software development is performed using an open source developing platform Android Studio IDE and Firebase Real-time Database, Authentication, Cloud Messaging and Cloud Storage. The method used to develop the software is waterfall modelling and two characters are involved, which is admin and customer. (Mei et al., 2020)



Figure 2. 4 : Laundry Mama

Aknan Laundries: Aknan application for the client allows the user to book a laundry, ironing or home services request from the center. The user can add his data easily and locate him on the map through specific selection processes that facilitate the process of completing the application data first from a city, neighborhood and real estate, then write his home data. They have surveyed different aspects on the basis of age group, marital status, smart phone users (android or iOS), employment and many more things which is analyzed and different results have been from that survey. This would help to target the customer who desperately need this type of laundry app for their personal benefits as well as it will help laundry sector to grow in coming years. This type of data analysis gives an idea about people and their requirements according to their needs which can be fulfilled by this laundry app.

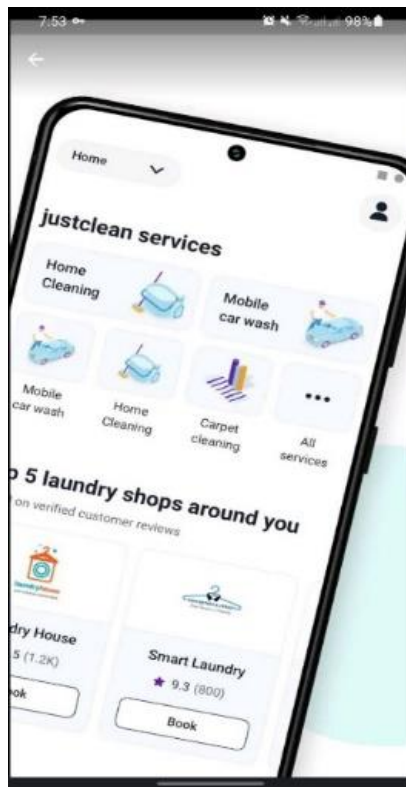


Figure 2. 5 Just Clean

Justclean: Just Clean, the Gulf's premier laundry and dry clean app. Choose the pick-up and delivery time from your favorite laundry in your area and you will get clean clothes in record time. justClean is the leading Laundry on Demand service provider.

Just download the app and with a few taps, your life becomes a lot easier. Our representative picks up your clothes from your doorstep, washes them, dries them, bends them, squeezes them and returns them to you within 24 hour. All of these applications operate under one scenario, which is one laundry that provides services, unlike our expanding application that provides services to the nearest laundries located in the neighborhood in which the user lives (Ibrahim Mohammed Saati, 2020).

Maurice Dry Cleaners their pride themselves refer to figure 2.2 above from (<http://www.mauricedrycleaners.com.au/>) on being leaders in garment cleaning technology servicing throughout the Sydney CBD. Their service to the CBD of Sydney is second to none

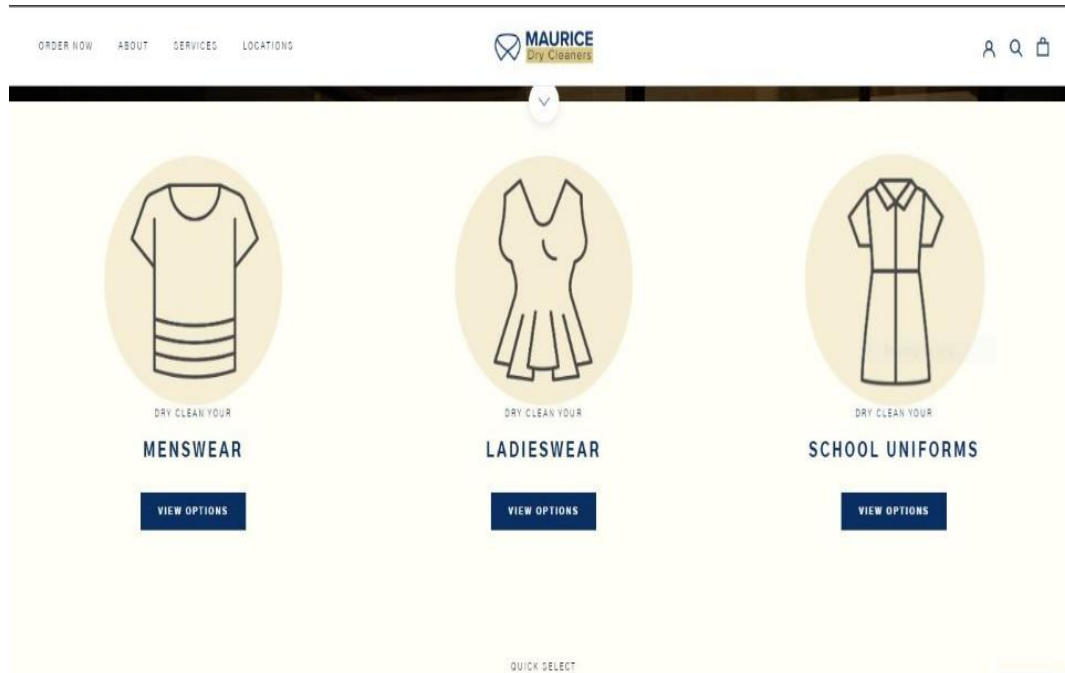


Figure 2. 6 Maurice Dry Cleaners

with their highest quality cleaning and same-day delivery always our priority. This is only achieved by 10 handy locations and fleet of vehicles that allow us to keep our prices competitive and delivered on time without fuss. From their high-end designer wear through to their crisp white business shirt service, no piece of clothing you wear is left out. At Maurice, with over 30 years of experience, we are committed to obtaining the most up to date of machinery with the greatest of consideration to our environment. Their all-new hydrocarbon dry cleaning machines along with our many states of the art shirt machines allows us to be leaders in the dry clean industry in the Sydney metro area.

The demand for application-based laundry services have triggered the customers to register and receive their laundry services with simple clicks. This insisted building a mobile application for laundry services. laundry mobile application to ease the processes involved in managing laundry including record keeping, payments, invoices, scheduling delivery, and pick up of laundry. By embracing this LaunApp mobile application, businesses and customers will be able to effectively deal with the process of managing laundry in a scheduled manner, without any interrupts in the routine activities of the customers (R & S, 2021).

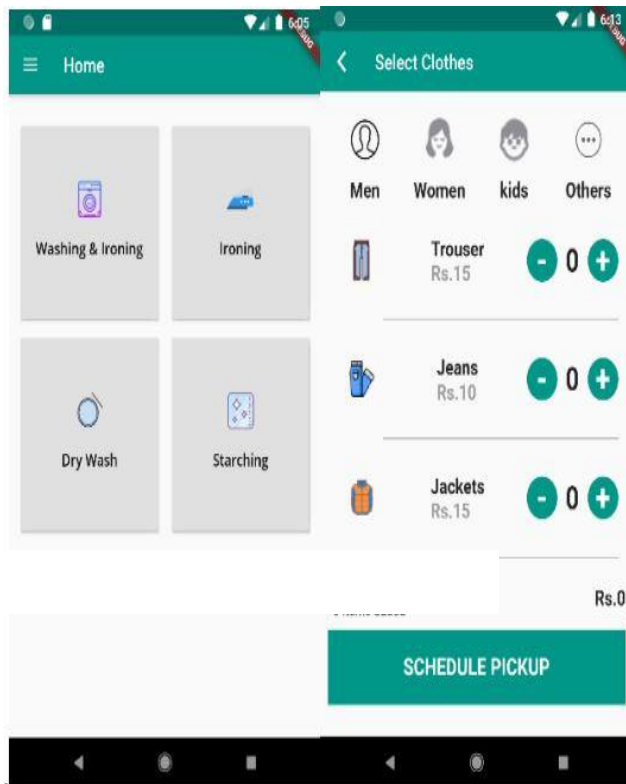


Figure 2. 7:LaunApp mobile application

CHAPTER III: METHODOLOGY

3.0 Introduction of laundry ordering app

Laundry ordering app is mobile application designed to simplify the process of ordering laundry services. With the laundry ordering app, users can select the laundry services they need, choose a pick-up and delivery time, pay for the service all through mobile device. The app connects users with local laundry service providers who handle the laundry, providing a convenient and time-saving solution for busy individuals who don't have time or inclination to do their own laundry.

Some laundry ordering apps also offer additional features like customization of wash setting dry cleaning and ironing services.

Overall, a laundry ordering app is great way to simplify the laundry process and make life a little easier.

3.1 overview of laundry ordering app

Laundry ordering app is a mobile application that enables users to order laundry services online. This app can be used to schedule pickup and delivery times, select the desired laundry service, and make payments electronically. The app can also provide users with real-time updates on the status of their laundry, and enable them to track their orders.

Typically, laundry ordering apps are used by busy individuals who don't have time or inclination to do their laundry themselves. These apps are particularly popular in urban areas where people have busy schedules and limited access to laundry facilities.

Laundry ordering apps may be offered by independent laundry services or as feature of larger, multi-service apps. They can be designed for both android and iOS devices and may have different features depending on the app. Some common features include the ability to schedule laundry pickup and delivery, select specific laundry services such as dry cleaning, and choose from a variety of payment options.

Overall, laundry ordering apps are convenient and time-saving for busy individuals who need their laundry done but don't have the time or desire to do it themselves.

3.2 system features of laundry ordering app

Some of the common system features of a laundry ordering app may include: -

User registration and authentication: The app should provide a user registration system where users can create an account and login to access the app's features.

Laundry ordering: The app should allow users to order laundry services online. This includes selecting the type of service, such as wash and fold or dry cleaning, scheduling the pickup and delivery times, and specifying any special requests.

Payment processing: The app should offer various payment options such as credit card, debit card, online wallets, or other electronic payment methods. The app should be secure and ensure that all financial transactions are processed safely.

Order tracking: The app should allow users to track their orders in real-time, from the time the laundry is picked up to the time it is delivered. The app should also provide users with notifications and updates on the status of their orders.

Customer support: The app should provide customer support, including a contact information, and chat support.

Reviews and ratings: The app should allow users to rate and review the laundry services they receive. This can help other users make informed decisions and provide feedback to the service provider.

Service provider management: The app should provide a platform for the service provider to manage their services, including managing laundry orders, scheduling pickups and deliveries, and managing customer feedback.

Overall, a laundry ordering app should be user-friendly, secure, and provide a convenient and efficient way to order laundry services.

3.3 software requirements

The software requirements of a laundry ordering app may include:

Platform: The app should be developed for a specific platform, such as Android or iOS. The development process may involve using different programming languages, tools, and frameworks specific to each platform.

Backend technology: The app should be built on a reliable and scalable backend technology such as firebase or GCP. The backend technology should include a database, API, and server infrastructure to support the app's features.

User interface design: The app should have a user-friendly interface that is intuitive and easy to use. The user interface should be designed with the target audience in mind, and the app should be optimized for different screen sizes and device types.

Payment gateway integration: The app should integrate with payment gateway systems such as Evc-plus to facilitate electronic payments.

Push notification system: The app should have a push notification system to notify users about order updates, promotional offers, and other relevant information.

Security features: The app should be developed with security in mind, including SSL encryption for secure data transfer, two-factor authentication, and secure payment processing.

Customer support: The app should include a customer support feature, such as a chatbot, helpdesk, or support email, to provide users with quick and efficient assistance.

Overall, a laundry ordering app should have a robust software architecture that can support its features, offer a user-friendly interface, integrate with payment and other relevant systems, and be secure and reliable.

3.4 Hardware requirements

The hardware requirements of a laundry ordering app may include:

Smartphones or tablets: The app should be developed to run on smartphones or tablets with the required operating system, such as Android or iOS. The hardware requirements may vary depending on the device type and version.

Internet connectivity: The app requires a stable internet connection to access the app's features and communicate with the backend server. Users may need to have access to 3G/4G or Wi-Fi connectivity.

Overall, the hardware requirements for a laundry ordering app are relatively low and should be compatible with most modern smartphones or tablets. Users only need access to stable internet connectivity, while service providers may need payment processing hardware for in-person payments.

3.5 user requirement of laundry ordering app

Here are some common user requirements for a laundry ordering app:

1. **User-Friendly Interface:** Users expect a clean and intuitive user interface that makes it easy to navigate the app and place an order. This includes features such as a simple ordering process, clear pricing information, and the ability to easily edit or cancel orders.
2. **Secure Payment Processing:** Users want the assurance that their payment information is secure and protected when making a payment through the app. The app should use industry-standard encryption and offer secure payment gateways such as credit card payment or PayPal integration.
3. **Order Tracking and Notifications:** Users expect to be able to track the progress of their order and receive notifications when it's ready for pickup or delivery. This includes real-time tracking of the delivery driver and estimated delivery times.
4. **Personalized Settings:** Users should be able to customize their profile settings, including their preferred pickup and delivery times, detergent preferences, and other relevant details.

5. Customer Support: Users expect responsive customer support, either through in-app messaging or a customer support hotline, to address any questions or concerns that they may have.

6. Flexible and Convenient Delivery Options: Users expect flexible delivery options, including the ability to choose pickup or delivery times, and to schedule recurring or one-time services.

7. Affordable Pricing: Users expect fair and competitive pricing, with transparent pricing information that helps them understand the cost of each service.

Overall, the key user requirement for a laundry ordering app is to provide a seamless and convenient experience that simplifies the process of ordering laundry services and provides a reliable and efficient service.

CHAPTER IV: SYSTEM ANALYSIS AND DESIGN

4.0 System analyses

The analysis of the system involves a laundry ordering app a convenient and hassle-free way to have your laundry done. With our app, you can easily place an order for laundry services from the comfort of your own home, and have your laundry delivered right to your doorstep it is a digital platform that allows customers to order laundry services from a laundry service provider. Our app also offers a range of features to make your laundry experience even more convenient. You can track the status of your orders in real-time, pay for your services securely and conveniently through the app, and get in touch with customer support if you have any questions or concerns. Here is a systematic analysis of a laundry ordering app:

4.1 Existing system/approach

There are several laundry ordering apps available on the market today. Some of the most popular ones include:

- **Rinse:** Rinse is a laundry and dry-cleaning app that offers pickup and delivery services in select cities in the United States. Users can schedule pickups and drop-offs, track the status of their orders, and receive notifications when their laundry is ready.
- **Laundry heap:** Laundry heap is an on-demand laundry and dry-cleaning app that operates in several countries, including the UK, the US, and Australia. Users can schedule pickups and drop-offs, select their preferred cleaning options, and track the progress of their orders.
- **Cleanly:** Cleanly is a laundry and dry-cleaning app that offers pickup and delivery services in select cities in the United States. Users can schedule pickups, manage their orders, and receive notifications when their laundry is ready.
- **Washio:** Washio was a laundry and dry-cleaning app that operated in several US cities until it was acquired by a competitor in 2016. Users could schedule

pickups and drop-offs, select their preferred cleaning options, and track the progress of their orders.

These laundry ordering apps typically work by allowing users to select their laundry preferences, such as the type of service they require and any special instructions, and then schedule a pickup and drop-off time. The app then connects the user with a local laundry or dry-cleaning service provider who will handle the order and deliver it back to the user once it's complete.

4.2 Proposed system

There are several key components that you may want to consider. Here are some ideas:

- **User Interface:** Your app's user interface should be intuitive, easy to navigate, and user-friendly. It should allow users to easily create an account, place orders, view their order history, and provide feedback. You may want to consider including features such as the ability to customize orders, schedule pickups, and deliveries, and view real-time order status updates.
- **Payment System:** Your app should have a secure and reliable payment processing system that allows users to pay for their laundry services using a variety of payment methods such as mobile payments.
- **Delivery System:** You'll need to have a system in place to manage deliveries, including scheduling pickups and drop-offs, managing driver routes, and tracking order status. You may also want to consider offering different delivery options such as same-day or next-day delivery.
- **Customer Service:** Providing excellent customer service is key to the success of your app. You may want to consider offering 24/7 customer support, providing multiple channels of communication such as phone, email, and chat, and offering refunds or compensation for any issues that arise.
- **Laundry services:** The app would offer a variety of laundry services, including pickup and delivery, dry cleaning, and wash and fold. Customers could select the specific services they need and customize their orders according to their preferences.

- Order processing: Once an order is placed, the app would notify the nearest laundry facility or partner to pick up the laundry. The facility would process the order and return the cleaned laundry to the customer within a specified timeframe.
- Notification system: The app would have a notification system that sends alerts to customers when their order is being processed, picked up, and delivered.

Overall, the proposed laundry ordering app would be designed to offer a seamless and hassle-free laundry ordering experience for customers, providing quality laundry services that are convenient and efficient.

4.3 System requirement:

System Requirement is guidelines and characteristics which must be included in the new system for satisfying all the users' requirements so that it would be acceptable to the user. We're going to discuss the work that needs to be done and share it in two separate

The requirements are divided into two categories: functional requirements and non-functional requirements

4.3.1 Functional Requirements

The functional requirement probably can divide into three categories namely, customer section, Staff section and administrator section.

- The customer's section is where to make order and others transactions as needed.
- The Staff section is dealing with customer to get the item he ordered.
- The administrator section is basically to responsibility of the system administrator to manage and maintain the database.

4.3.2 Non-functional requirement

Non-functional requirement is as important as functional requirement.

A non-functional requirement describes the features will be included in the system to provides a feasibility and ease of use to the users.

Reliability

A system is said to have reliability if it does not produce dangerous or costly failure when it is used in a reasonable manner that a typical user expected is normal. The

system to be developed must be reliable because reliability is one of the essential software qualities.

User Friendly and Usability

The system can be considered as attractive or as an easy-to-use application because the users only have to click on the task or image by using the mouse. The usage of suitable and meaningful icons or buttons will help the user to use the system with more confidence.

Response Time

The response time to retrieve the information can be considered within a reasonable interval time. It means that all desirable information should be available to users at any point in time.

4.4 Feasibility Study

The feasibility study is technical information and costs data to determine the economic potential and practicality of a project. The feasibility study uses techniques that help evaluate a project and/or compare it with other projects. A feasibility study is a part of the system development life cycle, which aims to determine whether it is sensible to develop some system. In this section, we will look at the feasibility of a study evaluating and analyzing two aspects of feasibility, operational feasibility, and scheduling feasibility.

4.4.1. Schedule feasibility

Duration: This survey was conducted between February and August 2022, and it took Months to prepare for this survey.

February	March	April
Chapter 1	Chapter 2	Chapter3
May	June	July
Chapter 4	Chapter 5	Chapter 6
August	August	
Completed our research	A panel of research Proposes	

4.4.2 Operational feasibility

The laundry industry being belonging to the service industry, the software product is usually the service that is provide to the clients' item. For example, the use of soaps for washing/cleaning the water consumption, space for drying the clothes and also the electricity consumption by the washing machine and ironing machine. Therefore, we feel that this sort of industry will be perceive on the go for target spectators due to practicability and convenience.

Dry cleaning is cleaning procedure by using a natural solvent instead of water since the clothing would be harmed by water and cleanser or detergent. Likewise, the cleaning method is observed to consume a lot of time, when contrasted with the usual washing, drying and folding processes.

4.4.3 Technical feasibility

There are several tools helped us to do this project including Laptop, internet, flutter app,

Word, Google, Google Drive.

4.4.4 Economic feasibility

The financial budget of our project includes:

Budget Item	Details	Amount
Transportation	This is the way that we struggle to overcome this project, to carry out we need to meet in two days (For transportation)	\$ 80
Other facilities	This is the fast foods & drinks during days in the meeting activities.	\$ 50
Internet	To do this project we needed to use high-quality internet services & also this is the most important of our project	\$ 60
		Total = \$190

4.5 System Design

Systems design is the process of defining elements of a system like modules, architecture,

Components and their interfaces and data for a system based on the specified requirements.

It is the process of defining, developing, and designing systems that satisfy the specific needs and requirements of a business or organization.

4.5.1 System Maintenance

The program may be maintained on the ground that the system requires an upgrade. When there is a new field to be added or a new form to be added in order to serve users well. Though it is compiled as a standalone software the database can be tempered with but it's advisable that the admin put a password on the file to secure the database from intrusion.

4.5.2 System Evaluation

This System is a high-standard program that can weather the storm of technological advancement, it is most needed in all Laundry and it is an antidote for business speed and transaction with record keeping and maintenance, it will be very helpful to clients and customers in the marketing business. All it needs is a computer-literate operative to make it work, it is stand-alone and automated.

4.6 System Architecture

A large system can be decomposed into sub-system that provides some related set of services. Thus, architecture design is the initial design process of identifying these subsystems and establishing a framework for sub-system control and communications. As part of the architecture design process~ the below activities are usually essential

4.6.1 Architecture Diagram

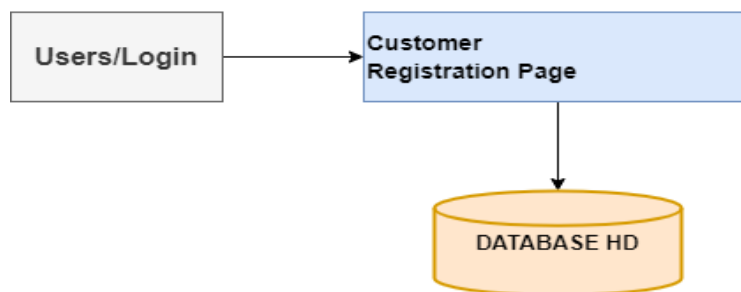


Figure 4. 1: Architecture diagram level

4.7 UML Diagram Types

UML provides both the structure diagrams and behavioral diagrams of the system. A set of diagrams with different graphical elements is the core part as well as the most expensive presentation in the UML. The UML includes nine kinds of diagrams, for the sake of grasping the most representative aspects of the design of the elevator system. In this paper only following the Parts of its UML diagrams are used:

- Use case Diagram
- Activity Diagram

4.7.1 Use case Diagram

A use-case model describes a system's functional requirements in terms of use cases. It is a model of the system's intended functionality (use cases) and its environment (actors). Use cases enable you to relate what you need from a system to how the system delivers on those needs.

The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases. Use-case diagrams generally show groups of use cases -- either all use cases for the complete system or a breakout of a particular group of use cases with related functionality.



Figure 4. 2: Use Case Diagram

4.7.2 Activity Diagram

The activity diagram is another important behavioral diagram in the UML diagram to describe dynamic aspects of the system. An activity diagram is essentially an advanced version of a flow chart that models the flow from one activity to another activity.

An activity diagram also is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. It describes the flow of control of the target system.

Login in Daigram

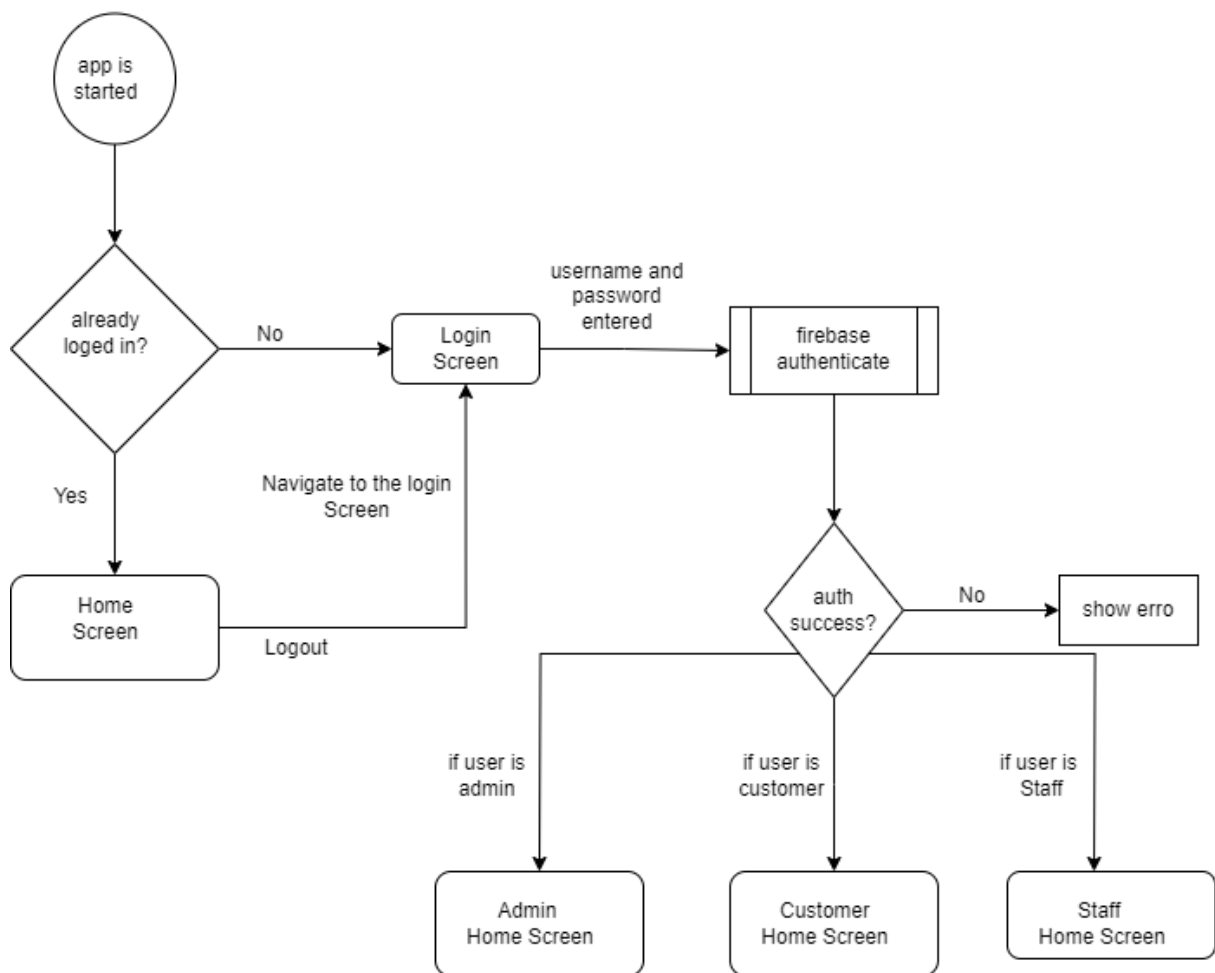


Figure 4. 3: Activity Diagram 1

4.8 Database Design

Database design is the process of creating a thorough data model for the actual database that the system uses for its back-end data storage. We are making Laundry Ordering App and since this is the initial version, we are utilizing Firebase Fire Store as our database, however

Cloud Firestore is a NoSQL, document-oriented database. Unlike a SQL database, there are no tables or rows. Instead, you store data in documents, which are organized into collections so, our database has the following collections

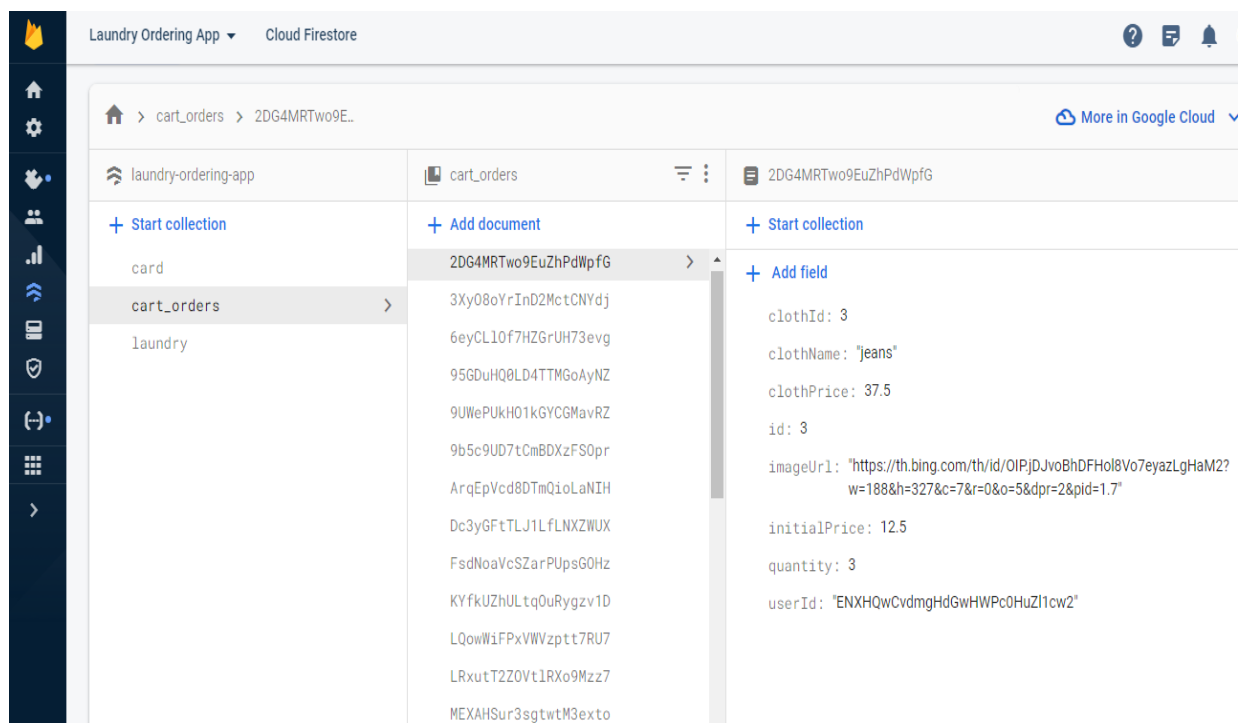


Figure 4. 4: Firebase Collections

4.9 System algorithms

Our laundry ordering app typically involves several algorithms to handle various tasks and optimize the user experience

1. User Authentication: this verifies the user's credentials, such as username and password.
2. Location-based: To provide laundry services, the app needs to determine the user's location.

3. **Order Management:** This handles the entire lifecycle of a laundry order, from placement to delivery
4. **Delivery Routing:** When the user requests laundry delivery, a routing algorithm is employed to optimize the delivery process
5. **Payment Processing:** This handles secure transactions between the user and the laundry service provider.

CHAPTER V: IMPLEMENTATION & TESTING

5.0 Introduction

This chapter considers the implementation and testing of the research which is laundry mobile application. The laundry app aims to provide users with a convenient and efficient way to manage their laundry needs. With the growing popularity of on-demand services and the increasing need for time-saving solutions, a laundry app offers a seamless platform for users to schedule pickups, select services, track orders, and make payments with ease. The chapter presents some important issues such as the overview of the implementation environment, the interface of the application and how they worked. This project demonstrates how to build a deep learning for laundry mobile application.

5.1 overview

A The main objective of the system is to provide an easy to use application that app serves as a digital interface between the users and the laundry service provider, streamlining the entire laundry process from start to finish. By leveraging mobile technology and online connectivity, users can effortlessly take care of their laundry requirements without the hassle of physically visiting a laundromat or dry cleaner.

5.2 Snapshots of the System

5.2.1 Login screen

The following snapshots determine the login screen of the system with their important characteristic or capability that a new system must achieve.

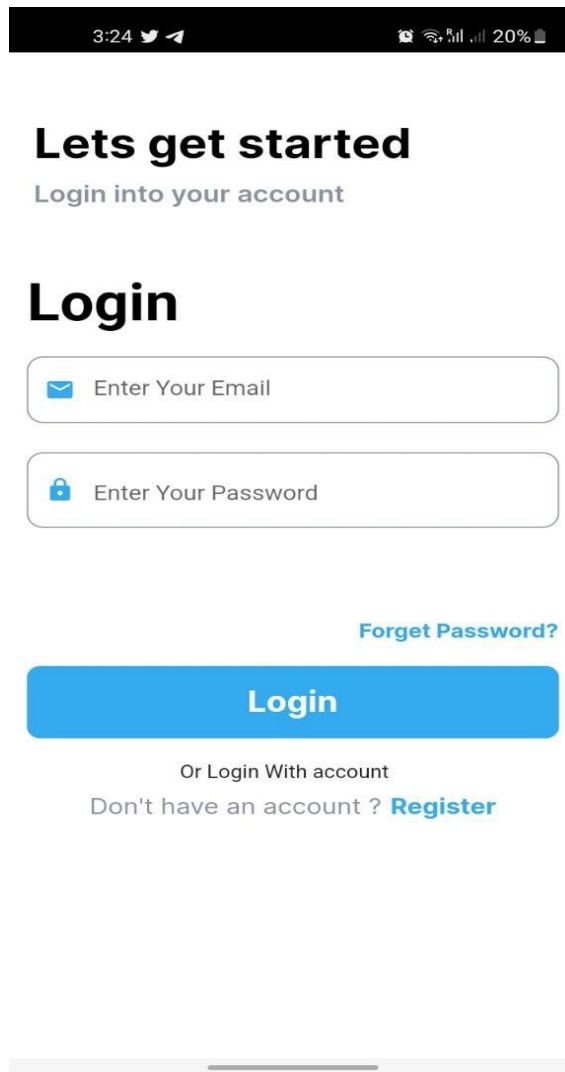
A mobile application login screen. At the top is a black status bar with white text showing '3:24' and various icons. Below the status bar, the text 'Lets get started' is in a large, bold, black font, followed by 'Login into your account' in a smaller, grey font. The main heading 'Login' is in a large, bold, black font. Below it are two rounded rectangular input fields. The first field has a blue envelope icon and the text 'Enter Your Email'. The second field has a blue padlock icon and the text 'Enter Your Password'. Below the password field is a blue link that says 'Forget Password?'. A large blue button with the word 'Login' in white text is centered below the links. Below the button, the text 'Or Login With account' is in a small, grey font, followed by 'Don't have an account ?' in a small, grey font, and a blue link that says 'Register'.

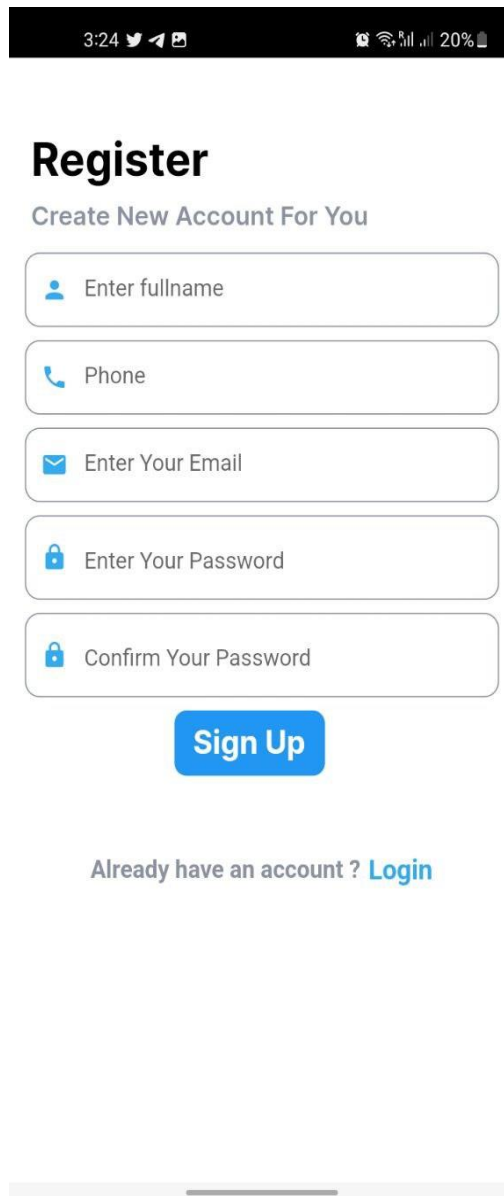
Figure 5. 1: Login screen

Brief description

A login page is a web page or interface that allows users to authenticate their identity and gain access to a secured system, application, or website. It serves as the initial entry point where users enter their credentials, such as a username and password, to verify their identity and establish a secure session.

It is important for login pages to users secure practices, such as using encryption for transmitting login credentials, implementing password strength requirements, and incorporating measures like CAPTCHA to prevent automated login attempts by bots. And all that service done by firebase authentication.

5.2.2 Register user screen



The image shows a mobile application screen for user registration. At the top is a black status bar with the time 3:24, social media icons, and a 20% battery level. The main heading is 'Register' in a large, bold, black font. Below it is the subtitle 'Create New Account For You' in a smaller, grey font. The registration form consists of five rounded rectangular input fields, each with a blue icon on the left and placeholder text: a person icon for 'Enter fullname', a phone icon for 'Phone', an envelope icon for 'Enter Your Email', a lock icon for 'Enter Your Password', and another lock icon for 'Confirm Your Password'. Below the form is a prominent blue button with the text 'Sign Up' in white. At the bottom of the screen, there is a link that says 'Already have an account ? Login' in a grey font, with 'Login' being a blue hyperlink. A thin horizontal line is visible at the very bottom of the screen.

Figure 5. 2:Signup Screen

Brief description

A register screen in Flutter refers to the user interface screen or page where users can create a new account or register for a service or application.

Here's a brief description of common elements found on a register screen in Flutter:

1. Text Fields/Input Fields: These are UI components where users can enter their information, such as username, email, password, and any other required details. Flutter provides different types of text fields, such as `TextField` or `TextFormField`, which you can customize based on your specific requirements.
2. Buttons: Register screens often include buttons for users to submit their registration information.
4. Error Messages: To handle errors during the registration process,
5. Navigation and Routing: Register screens often include navigation elements, allowing you to define routes and navigate between different screens.
6. Validation: It is common practice to validate user input on the register screen to ensure that the provided information meets certain criteria, such as password complexity or valid email format.

Home page screen

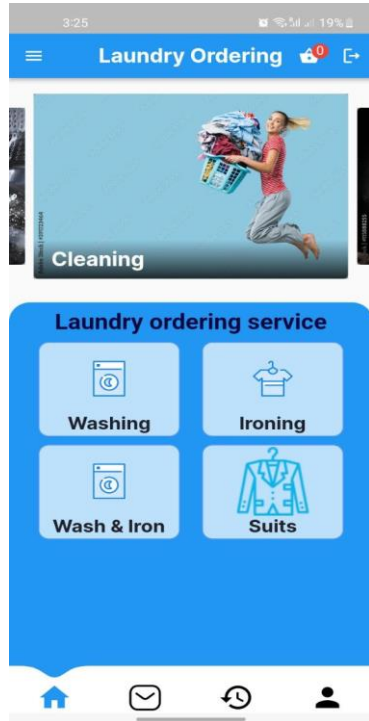


Figure 5. 3: Home page Screen

Brief description

A home screen in refers to the primary user interface (UI) screen or page that users see when you open an application. It serves as the main entry point and often contains various components and features that provide easy access to different sections and functionalities of the app.

1. App Bar/Top Bar: This is the horizontal bar at the top of the screen that displays the app's title and may include navigation icons or buttons as you see the laundry ordering and icons behind.

3. Content/Widgets: The main content area of the home screen contains four buttons which are:

Washing button: if there are clothes that are being wanted to be washed only.

Iron button: here this service will make clothes are being wanted to be ironed.

Wash and Iron button: clothes are being wanted to washed and ironed.

Suits button: this section is special for suites.

4. Bottom Navigation Bar: this offers quick access to different sections of the app. It typically consists of icons or labels representing various screens or functionalities. Our Bottom Navigation Bar consist of four sections.

Home screen: here is the first screen after login user it displays different service

Chats: this section is only communication screen.

History: this section will appear if the customer's order status if they completed, ongoing or delivered.

Profile: this screen shows users information and image

5.2.3 Cart Screen

Since we have already said that there are Content/Widgets which consist of four buttons such as washing button, ironing button, washing and ironing button and suits button. If one of them is hit, it will give us a special service and a price that is different from the others

Table 1

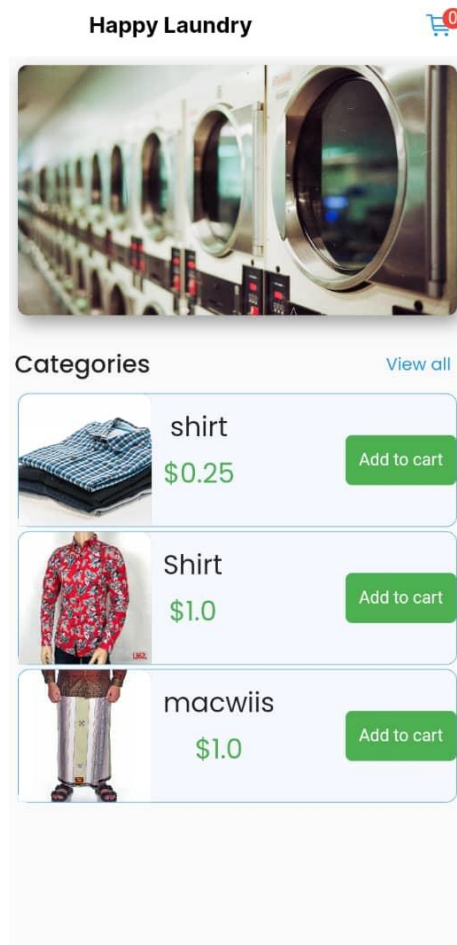


Figure 5. 4: Order Screen

Brief description

This cart screen service as when you select the clothes you want to for example to iron or washed by clicking add to cart button after select one item or more then click the corner cart iron button to navigator the next screen

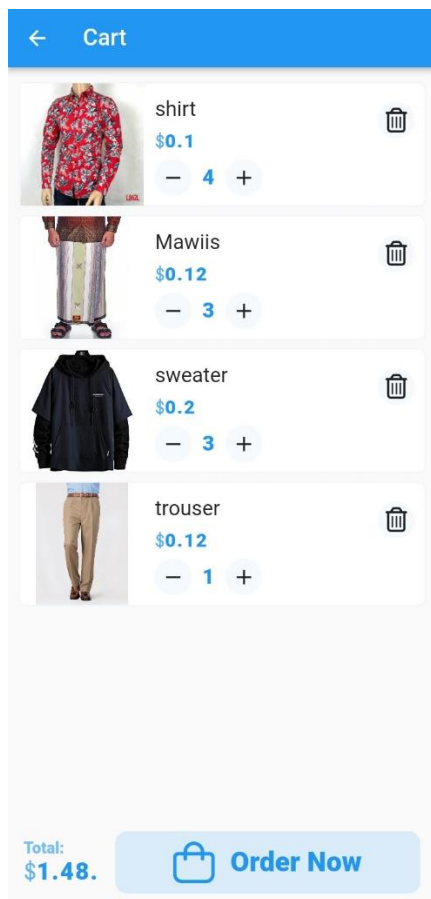


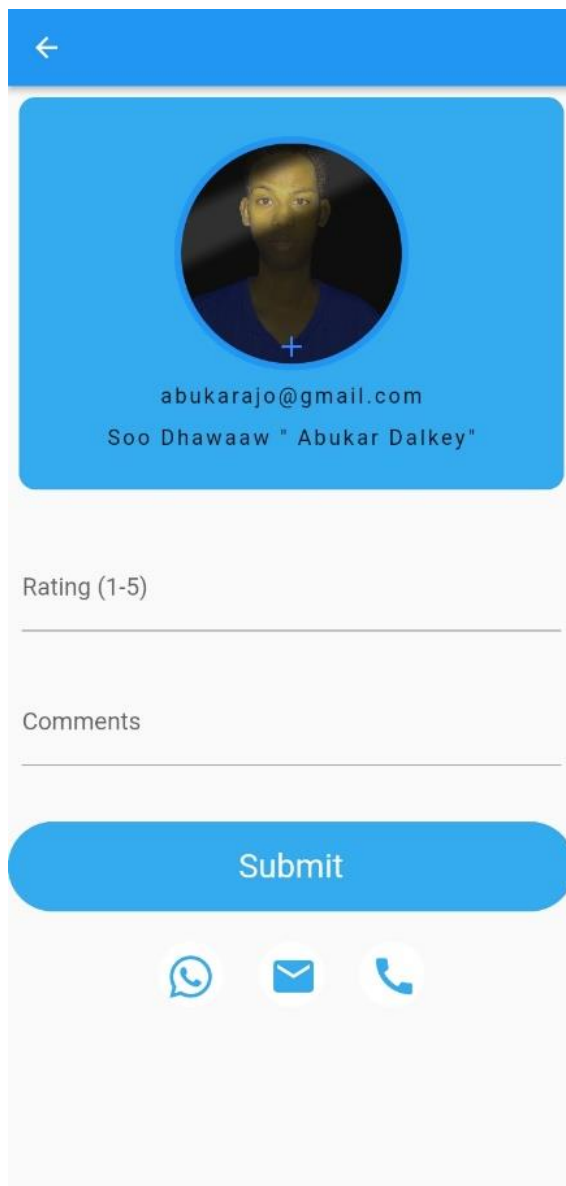
Figure 5. 5: Cart Screen

Brief description

This is cart screen which you can increment or decrement the cloth numbers and calculate the total money before Order now

5.2.4 Customer feedback

User Feedback and Problem Resolution: Including comprehensive feedback systems, such as reviews and ratings, will assist users in making educated judgments regarding laundry service providers. Furthermore, responding quickly to any concerns or complaints expressed by consumers via effective customer service channels can help to preserve user happiness.



The image shows a mobile application feedback screen. At the top is a blue header with a white back arrow. Below it is a blue rounded rectangle containing a circular profile picture of a man, a plus sign, the email address 'abukarajo@gmail.com', and the name 'Soo Dhawaaw " Abukar Dalkey"'. Underneath this is a 'Rating (1-5)' section with a horizontal line. Below that is a 'Comments' section with another horizontal line. A large blue rounded button labeled 'Submit' is positioned below the comments. At the bottom, there are three circular icons: a speech bubble with a phone handset, an envelope, and a phone handset.

Figure 5. 6: feedback screen

Brief description

This screen is where we obtain customers' feedback. And to know App's rate and if there are some issues that wanted to be fixed.

CHAPTER VI: DISCUSSION OF RESULTS

6.0 Introduction:

This chapter discusses the results and outcomes of utilizing such apps, highlighting the benefits, challenges, and potential areas for improvement.

6.1 Benefits of Laundry Order Service Apps:

6.2.1 Convenience and Time-Saving:

One of the main advantages of laundry services is the convenience they offer. Users can schedule pickup and delivery at their preferred time, eliminating the need for physical visits to laundromats. This saves time and effort, allowing users to focus on other activities or tasks.

6.2 Customization and Personalization:

Personalization and customization Many laundry programmers let users to customize their laundry preferences, such as washing and ironing particular clothing, ironing just specific clothes, washing only specified clothes, or unique clothes suits, which boosts customer pleasure.

6.2.2 Transparent Tracking and Updates:

Transparent Tracking and Updates: Laundry order service apps provide users with real-time updates on the progress of their laundry cycle. From pickup to cleaning to delivery, ensuring they are informed that if the state of their clothes are still ongoing or completed or delivered to customer's place.

6.2.3 Payment Options and clarity:

These apps often incorporate a variety of payment methods, including credit/debit cards and mobile wallets. Users may also examine clear pricing, which enables them to know up front how much services will cost them and prevents any unpleasant surprises. Currently, we utilize a local service in this app called EVC plus Service.

6.3 Challenges and Areas for Improvement:

6.2.4 Delivery Time Accuracy

Getting precise delivery timings while using laundry order service applications is a regular difficulty. Delivery deadlines could not always be reached because of things like excessive demand or unanticipated delays. Enhancing user happiness necessitates increasing delivery time accuracy and controlling consumer expectations in this respect.

6.2.5 Potential Clothing Damage

While partnering with professional laundry services reduces the risk, there is still a Possibility of clothes getting damaged or lost during the cleaning process
Implementing robust quality control measures and providing appropriate compensation or Reimbursement for any damages are essential to address this concern.

6.2.6 Privacy and Security

As users share personal information, including addresses and payment details, privacy and Security become important considerations. Implementing strong data protection measures, Encryption protocols, and ensuring compliance with relevant regulations can instill user Confidence in the app's security.

6.2.7 Handling Special Garments or Requirements

Some users may have specific garments or delicate fabrics that require special care. Ensuring that the laundry service providers associated with the app can handle such requirements properly and communicate any limitations or recommendations to users is crucial.

6.2.8 User Feedback and Issue Resolution:

Incorporating robust feedback mechanisms, such as reviews and ratings, can help users make informed decisions about the laundry service providers. Additionally, promptly addressing any issues or complaints raised by users through effective customer support channels can contribute to maintaining user satisfaction.

CHAPTER VII: CONCLUSION AND FUTURE WORK

7.0 Introduction

This chapter summarizes the results of the exploration that the researchers carried out over the course of a system that was in place for up to 5-6 months.

The chapter covers important topics such the research's conclusion and future work steps for individuals who want to undertake similar studies.

7.1 Conclusion

Laundry ordering app aims to simplify and enhance the laundry experience for users while providing a platform for laundry service providers to efficiently manage their operations. By combining convenience, user-friendly features, and effective customer service, a well-executed app can create a win-win situation for both users and businesses in the laundry industry.

By the end of this study, Laundry Ordering App using Mobile App is expected that it can encourage the citizens to use the system platform to use laundry service in their daily life. The interfaces less complex design and easy to use which the user only needs to fulfil the booking form and confirmation. Then, the automatic result for their laundry service will display. While on the staff site, this system will help them to view a list of booking data from the customer and update the data.

The results of utilizing laundry order service apps showcase several benefits, including convenience, time-saving, professional cleaning, customization, and transparent tracking. However, challenges such as delivery time accuracy, potential clothing damage, privacy concerns, and handling special garments require attention for continuous improvement.

By addressing these challenges and focusing on enhancing user experience, laundry order service apps can further solidify their position as a reliable and convenient solution for outsourced laundry services.

7.2 Future work:

- In the future, we plan to add Laundry apps provide flexibility in terms of scheduling. Users can choose pickup and delivery times that suit their preferences and availability. This flexibility accommodates varying schedules and helps users fit laundry tasks into their busy lives.
- We suggest that to add different payment methods as we use only EvcPlus.
- We plan to add women's laundry section. And house items section

REFERENCES

- Adekola, O. D., Maitanmi, S. O., Akande, O., Ajayi, W., Omotunde, A., Ayo-Fanibe, F. S., & Adeoye, I. T. (2021). *Online Laundry Management System*. 41(1).
- Bamasoud, D. M., M., A., A., E., A., R., S., M., A., Z., M., A., F., S., & D., A. (2018). An Explorative Study for Laundry Mobile Application. *International Journal of Advanced Computer Science and Applications*, 9(4). <https://doi.org/10.14569/IJACSA.2018.090417>
- D. V. Kozvov, D. Luciv and G. A. Chernishev. “Duplicate management in software documentation maintenance.” *Proceedings of V International conference Actu*
- Gas Sector in Yemen. (2020). *International Journal of Advanced Trends in Computer Science and Engineering*, 9(5), 6959–6971. <https://doi.org/10.30534/ijatcse/2020/13952020>.
- Gupta, A., Panda, D. K., & Pande, M. (2018). Development of Mobile Application for Laundry Services Using Android Studio. 13(12).
- Intention to Adopt Electronic Records Management System in the Oil and
- Norolazmi, N., Kadir, I. K. A., & Ridwan, M. (2022). A Review of the Common Factors Influencing E-Records Implementation in Business Organizations. 1.
- Omar, A. H., Rosli, N., & Esa, N. A. (2022). Dobby2U: Online Laundry Services Management System. 3(1).
- Shoewu, O., Makanjuola, N. T., Phillips, D. A., & Emmanuel, A. (2016). Design and Implementation of a Laundry Management System. . . *Number*, 17(2).
- Soni, V. D. (2020). Management Information Systems: Mastering the Discreet Planning. *International Journal for Research in Applied Science and Engineering Technology*, 8(6), 1587–1591. <https://doi.org/10.22214/ijraset.2020.6258>
- Einav, L., Levin, J., Popov, I., & Sundaresan, N. (2014). Growth, Adoption, and Use of Mobile E-Commerce. *American Economic Review*, 104(5), 489–494. <https://doi.org/10.1257/aer.104.5.489>

Otakwar, P., Darde, D., Gondke, N., Rokade, M., & Kulkarni, M. (n.d.). *Laundry Service System [LSS] (Web Application)*. 2(2).

Jamgade, A., Gawande, K., Shingade, S., & Anjankar, P. (2016). *E – Laundry Portal- A Review*.

Bamasoud, D. M., M., A., A., E., A., R., S., M., A., Z., M., A., F., S., & D., A. (2018). An Explorative Study for Laundry Mobile Application. *International Journal of Advanced Computer Science and Applications*, 9(4). <https://doi.org/10.14569/IJACSA.2018.090417>

Gupta, A., Panda, D. K., & Pande, M. (2018). *Development of Mobile Application for Laundry Services Using Android Studio*. 13(12).

Ibrahim Mohammed Saati. (2020). *Laundry Services Application*. <https://doi.org/10.13140/RG.2.2.15399.16804>

International Journal of Scientific Research in Computer Science, Engineering and Information Technology. (n.d.). 2(2).

Jack, T. (2013). Laundry routine and resource consumption in Australia: Laundry routines and consumption. *International Journal of Consumer Studies*, 37(6), 666–674. <https://doi.org/10.1111/ijcs.12048>

Justclean in play store Available at "https://play.google.com/store/apps/details?id=com.justclean.justclean" Last accessed: December 2020 Superwash Laundries in play store Available at "https://play.google.com/store/apps/details?id=com.superwash" Last accessed: December 2020

Kaburuan, E. R. (n.d.). DESIGNING OF IOT-BASED LINEN MANAGEMENT SYSTEM USING TOGAF ENTERPRISE ARCHITECTURE.

Máša, V., Bobák, P., Kuba, P., & Stehlík, P. (2013). Analysis of energy efficient and environmentally friendly technologies in professional laundry service. *Clean Technologies and Environmental Policy*, 15(3), 445–457. <https://doi.org/10.1007/s10098-013-0618-2>

- Mei, L. Y., Ku Azir, K. N. F., Ibrahim, S. Z., & Azemi, S. N. (2020a). LaundryMama: Humanising Laundry Tasks using Laundry Management System and Laundry-On-Demand Mobile Applications. *IOP Conference Series: Materials Science and Engineering*, 767(1), 012061. <https://doi.org/10.1088/1757-899X/767/1/012061>
- Mei, L. Y., Ku Azir, K. N. F., Ibrahim, S. Z., & Azemi, S. N. (2020b). LaundryMama: Humanising Laundry Tasks using Laundry Management System and Laundry-On-Demand Mobile Applications. *IOP Conference Series: Materials Science and Engineering*, 767(1), 012061. <https://doi.org/10.1088/1757-899X/767/1/012061>
- Omar, A. H., Rosli, N., & Esa, N. A. (2022). *Dobby2U: Online Laundry Services Management System*. 3(1).
- Upadhyaya, B. R., & Kavatekar, D. S. (n.d.). *LAUNDRY MANAGEMENT AND ENVIRONMENT – A*.
- Zhang, S., Alanezi, K., Gartrell, M., Han, R., Lv, Q., & Mishra, S. (2018). Understanding Group Event Scheduling via the OutWithFriendz Mobile Application. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4), 1–19. <https://doi.org/10.1145/3161200>
- Celikkan, U., & Kurtel, K. (2017). Application of Service-Oriented Context-Aware Architecture to Laundry Management System. *Studies in Informatics and Control*, 26(2). <https://doi.org/10.24846/v26i2y201707>
- Ibrahim, L. A. (n.d.). *LAUNDRY MANAGEMENT SYSTEM: DESIGN AND IMPLEMENTATION*.
- Tong, L., & Gao, W. (2016). Application-aware traffic scheduling for workload offloading in mobile clouds. *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 1–9. <https://doi.org/10.1109/INFOCOM.2016.7524520>
- TeachCert, L. B. (n.d.). *A History of Union Organisation of Laundry Work in Western Australia 1912—1975*.

APPENDIX

Appendix A // Connection Firebase

```
----- CONNECTION FIREBASE -----

import 'package:firebase_auth/firebase_auth.dart';

import 'package:provider/provider.dart';

import 'firebase_options.dart';

void main() async {

  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp(

    options: DefaultFirebaseOptions.currentPlatform,

  );

  runApp(

    child: LaundryApp(),

  );

}

class LaundryApp extends StatefulWidget {

  const LaundryApp({super.key});

  @override

  State<LaundryApp> createState() => _LaundryAppState();

}

class _LaundryAppState extends State<LaundryApp> {

  //const LaundryApp({super.key});

  late StreamSubscription<User?> user;
```

```

@override

void initState() {
  super.initState();

  user = FirebaseAuth.instance.authStateChanges().listen((user) {
    if (user == null) {
      print('User is currently signed out!');
    } else {
      print('User is signed in!');
    }
  });
}

```

```

@override

void dispose() {
  user.cancel();
  super.dispose();
}

```

```

@override

Widget build(BuildContext context) {
  return Sizer(builder: (context, orientation, deviceType) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      // ignore: unnecessary_null_comparison
      initialRoute: FirebaseAuth.instance.currentUser == null
        ? OnboardingPage.id

```

```
        : LoginPage.id,  
    routes: {  
        OnboardingPage.id: (context) => OnboardingPage(),  
        LoginPage.id: (context) => LoginPage(),  
    },  
    home: OnboardingPage(),  
);  
});  
}  
}
```


Appendix B // Signup User

- - - - - SIGNUP User-----

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart'
  show CollectionReference, FirebaseFirestore;
import 'package:firebase_database/firebase_database.dart';

import '../exports.dart';

class SignUpUsers extends StatefulWidget {
  static String id = 'signUpUsers';
  const SignUpUsers({super.key});

  @override
  State<SignUpUsers> createState() => _SignUpUsersState();
}

class _SignUpUsersState extends State<SignUpUsers> {
  bool showProgress = false;
  bool _isObscure1 = true;
  bool _isObscure2 = true;
  bool visible = false;
  bool isloading = false;
  late String txtname = "",
    txtphone = "",
    txtemail = "",
    txtpassword = "",
    txtconfirmpassword = "",
    orderstatus = "",
    image = "";
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _name = TextEditingController();
  final TextEditingController _phone = TextEditingController();
  final TextEditingController _email = TextEditingController();
  final TextEditingController _password = TextEditingController();
  final TextEditingController _confirmpassword = TextEditingController();
  final databaseReference =
    // ignore: deprecated_member_use
    FirebaseDatabase.instance.reference().child("Users");
  final _auth = FirebaseAuth.instance;
  FirebaseAuth auth = FirebaseAuth.instance;

  void _clear() {
    _name.clear();
    _phone.clear();
```

```

_email.clear();
_password.clear();
_confirmpassword.clear();
}

Future<User?> signUp(
  {required String email,
  required String password,
  required String name,
  required String phone,
  required String orderstatus,
  required String image,
  required BuildContext context}) async {
const CircularProgressIndicator();
if (_formKey.currentState!.validate()) {
  await _auth
    .createUserWithEmailAndPassword(email: email, password: password)
    .then((value) => {
      postDetailsToFirestore(
        name, email, password, phone, orderstatus, image)
    })
  // ignore: body_might_complete_normally_catch_error
  .catchError((e) {
    if (e.code == 'weak-password') {
      print(e.toString());
      var snackbar = SnackBar(
        content:
          Text('The password provided is too weak. ${e.toString()}'),
        ScaffoldMessenger.of(context).showSnackBar(snackbar);
    } else if (e.code == 'email-already-in-use') {
      var snackbar = SnackBar(
        content: Container(
          padding: const EdgeInsets.all(16),
          decoration: const BoxDecoration(
            color: Color(0xffC72c41),
            borderRadius: BorderRadius.all(Radius.circular(20)),
          ),
          child: const Text('email already exists '),
        ),
        behavior: SnackBarBehavior.floating,
        backgroundColor: Colors.transparent,
        elevation: 0,
        duration: const Duration(seconds: 2),
      );
      // SnackBar(
      //   content: Text(
      //     'The account already exists for that email.. ${e.toString()}');

```

```

        ScaffoldMessenger.of(context).showSnackBar(snackbar);
    }
  });
}
return null;
}

///// here is the signup method
///
postDetailsToFirestore(String email, String password, String name,
    String phone, String orderstatus, String image) async {
  // ignore: unused_local_variable
  FirebaseFirestore firebaseFirestore = FirebaseFirestore.instance;
  var user = _auth.currentUser;
  CollectionReference ref =
    FirebaseFirestore.instance.collection('customers');
  ref.doc(user!.uid).set({
    'name': txtname,
    'email': txtemail,
    'password': txtpassword,
    'phone': txtphone,
    'orderstatus': orderstatus,
    'image': image,
  });
  // Navigator.pushReplacement(
  //   context, MaterialPageRoute(builder: (context) => const LoginPage()));
}

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.grey.withOpacity(0.0),
        elevation: 0,
      ),
      backgroundColor: Colors.white,
      body: ListView(
        padding: const EdgeInsets.symmetric(horizontal: 12),
        children: [
          Form(
            key: _formKey,
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                const CustomTitles(

```

```

        subtitle: "Create New Account For You",
        title: "Register"),
    TextFormField(
      controller: _name,
      keyboardType: TextInputType.emailAddress,
      decoration: InputDecoration(
        prefixIcon: const Icon(MdiIcons.account,
          color: Kactivecolor, size: 22),
        contentPadding: const EdgeInsets.all(18),
        fillColor: Colors.black,
        enabledBorder: OutlineInputBorder(
          borderSide:
            const BorderSide(color: Kactivecolor, width: 2),
          borderRadius: BorderRadius.circular(12),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
          borderSide:
            const BorderSide(color: Kactivecolor, width: 1),
        ),
        hintText: "Enter fullname",
      ),
      onSave: (value) {
        txtname = value!;
      },
      validator: validatename,
    ),
    const SizedBox(
      height: 10,
    ),
    TextFormField(
      controller: _phone,
      keyboardType: TextInputType.number,
      decoration: InputDecoration(
        prefixIcon: const Icon(MdiIcons.phone,
          color: Kactivecolor, size: 22),
        contentPadding: const EdgeInsets.all(18),
        fillColor: Colors.black,
        enabledBorder: OutlineInputBorder(
          borderSide:
            const BorderSide(color: Kactivecolor, width: 2),
          borderRadius: BorderRadius.circular(12),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
          borderSide: const BorderSide(
            color: Kactivecolor, width: 1)),
      ),
    ),
  ),
),

```

```

        hintText: "Phone",
      ),
      onSave: (value) {
        txtphone = value!;
      },
      validator: validatePhone,
    ),
    const SizedBox(
      height: 10,
    ),
    TextFormField(
      controller: _email,
      keyboardType: TextInputType.emailAddress,
      decoration: InputDecoration(
        prefixIcon: const Icon(MdiIcons.email,
          color: Kactivecolor, size: 22),
        contentPadding: const EdgeInsets.all(18),
        fillColor: Colors.black,
        enabledBorder: OutlineInputBorder(
          borderSide:
            const BorderSide(color: Kactivecolor, width: 2),
          borderRadius: BorderRadius.circular(12),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
          borderSide: const BorderSide(
            color: Kactivecolor, width: 1)),
        hintText: "Enter Your Email",
      ),
      onSave: (value) {
        txtemail = value!;
      },
      validator: validateEmail,
    ),
    const SizedBox(
      height: 10,
    ),
    TextFormField(
      obscureText: _isObscure1,
      controller: _password,
      decoration: InputDecoration(
        prefixIcon: const Icon(MdiIcons.lock,
          color: Kactivecolor, size: 22),
        contentPadding: const EdgeInsets.all(18),
        fillColor: Colors.black,
        enabledBorder: OutlineInputBorder(
          borderSide:

```

```

        const BorderSide(color: Kactivecolor, width: 2),
        borderRadius: BorderRadius.circular(12),
    ),
    suffix: InkWell(
      child: Icon(
        _isObscure1
          ? Icons.visibility
          : Icons.visibility_off,
        color: Kactivecolor,
      ),
      onTap: () {
        setState(() {
          _isObscure1 = !_isObscure1;
        });
      },
    ),
    focusedBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(
        color: Kactivecolor, width: 1)),
    hintText: "Enter Your Password",
  ),
  onSave: (value) {
    txtpassword = value!;
  },
  validator: validatePassword,
),
const SizedBox(
  height: 10,
),
TextFormField(
  obscureText: _isObscure2,
  controller: _confirmpassword,
  decoration: InputDecoration(
    suffix: InkWell(
      child: Icon(
        _isObscure2
          ? Icons.visibility
          : Icons.visibility_off,
        color: Kactivecolor),
      onTap: () {
        setState(() {
          _isObscure2 = !_isObscure2;
        });
      },
    ),
    prefixIcon: const Icon(MdiIcons.lock,

```

```

        color: Kactivecolor, size: 22),
        contentPadding: const EdgeInsets.all(18),
        fillColor: Colors.black,
        enabledBorder: OutlineInputBorder(
          borderSide:
            const BorderSide(color: Kactivecolor, width: 2),
          borderRadius: BorderRadius.circular(12),
        ),
        focusedBorder: OutlineInputBorder(
          borderRadius: BorderRadius.circular(12),
          borderSide: const BorderSide(
            color: Kactivecolor, width: 1)),
        hintText: "Confirm Your Password",
      ),
      onSave: (value) {
        txtconfirmpassword = value!;
      },
      validator: (value) {
        if (value!.isEmpty) {
          return 'Empty';
        }
        if (txtconfirmpassword != txtpassword) {
          return 'Not Match';
        }
        return null;
      },
    ),
    const SizedBox(
      height: 10,
    ),
    Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        showProgress
          ? const CircularProgressIndicator()
          : custombtn(
              colorbtn: Kactivecolor,
              colortxt: Colors.white,
              txtbtn: "SignUp",
              onpress: () async {
                if (_formKey.currentState!.validate()) {
                  _formKey.currentState!.save();
                  showProgress = true;

                  await signUp(
                    email: _email.text.trim(),

```

```

        password: _password.text.trim(),
        name: txtname,
        phone: txtphone,
        orderstatus: orderstatus,
        image: image,
        context: context,
    );
}

if (auth.currentUser != null) {
    Navigator.pushAndRemoveUntil(
        context,
        MaterialPageRoute(
            builder: (ctx) =>
                const EmailVerificationScreen(),
        ),
        (route) => false);
}
setState(() {
    showProgress = false;
});
FocusScope.of(context)
    .requestFocus(FocusNode());
},
),
const SizedBox(
    width: 20,
),
],
),
const SizedBox(
    height: 30,
),
Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
        const Text(
            "Already have an account ?",
            style: TextStyle(
                color: Kinactivetextcolor,
                fontSize: 17,
                fontWeight: FontWeight.bold),
        ),
        const SizedBox(
            width: 6,
        ),
        GestureDetector(

```



```

        onTap: () => Navigator.pushAndRemoveUntil(
          context,
          MaterialPageRoute(
            builder: (ctx) => const LoginPage(),
          ),
          (route) => false),
        child: const Text("Login",
          style: TextStyle(
            color: Kactivecolor,
            fontSize: 19,
            fontWeight: FontWeight.bold)),
      ),
    ],
  ),
],
),
),
],
)),
);
}
}

```

```

String? validateName(String? user) {
  if (user!.isEmpty) {
    return 'Enter your name';
  } else {
    return null;
  }
}

```

```

String? validatePhone(String? phone) {
  if (phone!.isEmpty) {
    return 'Enter your phone';
  } else {
    return null;
  }
}

```

```

String? validateEmail(String? email) {
  if (email!.isEmpty) {
    return 'Enter email address';
  } else {
    return null;
  }
}

```

```

String? validatePassword(String? password) {
  if (password!.isEmpty) {
    return 'Enter the password';
  } else {
    return null;
  }
}

String? validateConfirmPassword(String? cpassword) {
  if (cpassword!.isEmpty) {
    return 'confirm the password';
  } else {
    return null;
  }
}

class BuildButton extends StatelessWidget {
  final IconData iconData;
  final Function() onPressed;
  final String title;
  const BuildButton({
    super.key,
    required this.iconData,
    required this.onPressed,
    required this.title,
  });

  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
      style: ElevatedButton.styleFrom(
        minimumSize: const Size.fromHeight(56),
        primary: Colors.white,
        onPrimary: Colors.black,
        textStyle: const TextStyle(fontSize: 20)),
      onPressed: onPressed,
      child: Row(
        children: [
          Icon(
            iconData,
            size: 28,
          ),
          Text(title),
        ],
      ));
  }
}

```

Appendix C // Login Screen

----- LOGIN SCREEN -----

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:laundry_order_app/admin/admin_dashboard.dart';
import 'package:laundry_order_app/exports.dart';
import 'package:laundry_order_app/utility/middle_auth.dart';

class LoginPage extends StatefulWidget {
  static String id = 'loginpage';

  const LoginPage({ Key? key }) : super(key: key);

  @override
  State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
  // Get the device token using Firebase Messaging
  Future<String?> getDeviceToken() async {
    FirebaseMessaging messaging = FirebaseMessaging.instance;

    // Request permission (iOS only)
    NotificationSettings settings = await messaging.requestPermission(
      alert: true,
      badge: true,
      sound: true,
    );

    print('User granted permission: ${settings.authorizationStatus}');

    // Get the device token
    String? token = await messaging.getToken();
    print('Device token: $token');
    return token;
  }

  final TextEditingController _email = TextEditingController();
  final TextEditingController _password = TextEditingController();

  bool _isObscure = true;
  bool visible = false;
  bool isloading = false;
```

```

final GlobalKey<FormState> _formkey = GlobalKey<FormState>();

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.white,
        elevation: 0,
      ),
      backgroundColor: Colors.white,
      body: ListView(
        children: [
          Padding(
            padding: const EdgeInsets.only(left: 13, right: 13),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                const CustomTitles(
                  subtitle: "Login into your account",
                  title: "Lets get started",
                ),
                const SizedBox(
                  height: 9,
                ),
                Form(
                  key: _formkey,
                  child: Column(
                    mainAxisAlignment: MainAxisAlignment.center,
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      const SizedBox(
                        height: 10,
                      ),
                      const Text(
                        "Login",
                        style: TextStyle(
                          fontWeight: FontWeight.bold,
                          color: Colors.black,
                          fontSize: 40,
                        ),
                      ),
                      const SizedBox(
                        height: 20,
                      ),
                      TextFormField(
                        controller: _email,

```

```

decoration: InputDecoration(
  prefixIcon: const Icon(
    MdiIcons.email,
    color: Kactivecolor,
    size: 22,
  ),
  contentPadding: const EdgeInsets.all(18),
  fillColor: Colors.black,
  enabledBorder: OutlineInputBorder(
    borderSide: const BorderSide(
      color: Kactivecolor, width: 2),
    borderRadius: BorderRadius.circular(12),
  ),
  focusedBorder: OutlineInputBorder(
    borderRadius: BorderRadius.circular(12),
    borderSide: const BorderSide(
      color: Kactivecolor, width: 1),
  ),
  hintText: "Enter Your Email",
),
validator: (value) {
  if (value!.isEmpty) {
    return "Email cannot be empty";
  }
  bool emailValid = RegExp(
    r'^.+@[a-zA-Z]+\.{1}[a-zA-Z]+(\.{0,1}[a-zA-Z]+)$',
  ).hasMatch(value);

  if (!emailValid) {
    return "Please enter a valid email";
  }
  return null;
},
keyboardType: TextInputType.emailAddress,
),
const SizedBox(
  height: 25,
),
TextFormField(
  controller: _password,
  obscureText: _isObscure,
  decoration: InputDecoration(
    prefixIcon: const Icon(
      MdiIcons.lock,
      color: Kactivecolor,
      size: 22,
    ),
  ),

```

```

suffix: InkWell(
  child: Icon(
    _isObscure
      ? Icons.visibility
      : Icons.visibility_off,
    color: Kactivecolor),
  onTap: () {
    setState(() {
      _isObscure = !_isObscure;
    });
  },
),
contentPadding: const EdgeInsets.all(18),
fillColor: Colors.black,
enabledBorder: OutlineInputBorder(
  borderSide: const BorderSide(
    color: Kactivecolor, width: 2),
  borderRadius: BorderRadius.circular(12),
),
focusedBorder: OutlineInputBorder(
  borderRadius: BorderRadius.circular(12),
  borderSide: const BorderSide(
    color: Kactivecolor, width: 1),
),
hintText: "Enter Your Password",
),
validator: (value) {
  RegExp regex = RegExp(r'^.{6,}$');
  if (value!.isEmpty) {
    return "Password cannot be empty";
  }
  if (!regex.hasMatch(value)) {
    return "Please enter a valid password (minimum 6 characters)";
  }
  return null;
},
keyboardType: TextInputType.emailAddress,
),
Visibility(
  maintainSize: true,
  maintainAnimation: true,
  maintainState: true,
  visible: visible,
  child: Center(
    child: Container(
      child: const CircularProgressIndicator(
        color: Colors.black,

```

```

        ),
      ),
    ),
  ],
),
GestureDetector(
  onTap: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (_) => const ForgotScreen(),
      ),
    );
  },
  child: Row(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      Text(
        "Forget Password?",
        style: GoogleFonts.inter(
          color: Kactivecolor,
          fontWeight: FontWeight.bold,
          fontSize: 15,
        ),
      ),
    ],
  ),
  const SizedBox(
    height: 20,
  ),
  ElevatedButton(
    style: ElevatedButton.styleFrom(
      minimumSize: const Size(390, 62),
      backgroundColor: Kactivecolor,
      elevation: 0,
      shape: const RoundedRectangleBorder(
        borderRadius: BorderRadius.all(
          Radius.circular(10),
        ),
      ),
    ),
    onPressed: () {
      signIn(context, _email.text, _password.text);
      FocusScope.of(context).requestFocus(FocusNode());
    },
  ),
),

```

```

    },
    child: isloading
      ? const CircularProgressIndicator(
        backgroundColor: Kactivecolor,
        color: Colors.white,
      )
      : Text(
        "Login",
        style: GoogleFonts.inter(
          color: Colors.white,
          fontSize: 22,
          fontWeight: FontWeight.bold,
        ),
      ),
  ),
  const SizedBox(
    height: 20,
  ),
  SizedBox(
    height: 40,
    child: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text(
          "Don't have an account?",
          style: GoogleFonts.inter(
            color: Kinactivetextcolor,
            fontSize: 17,
            fontWeight: FontWeight.w400,
          ),
        ),
        const SizedBox(
          width: 6,
        ),
        GestureDetector(
          onTap: () => Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => const SignUpUsers(),
            ),
          ),
          child: Text(
            "Register",
            style: GoogleFonts.inter(
              color: Kactivecolor,
              fontSize: 17,
              fontWeight: FontWeight.bold,
            ),
          ),
        ),
      ],
    ),
  ),

```



```

        ),
      ),
    ),
    ],
  ),
),
],
),
),
],
),
),
);
}

void route() {
  var user = FirebaseAuth.instance.currentUser;
  // ignore: unused_local_variable
  var newuser = FirebaseFirestore.instance
    .collection('users')
    .doc(user!.uid)
    .get()
    .then((DocumentSnapshot documentSnapshot) {
  try {
    if (documentSnapshot.exists) {
      if (documentSnapshot.get('role') == "admin") {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (context) => const AdminDashboard(),
          ),
        );
      } else {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (context) => const StaffDashboard(),
          ),
        );
      }
    } else {
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(
          builder: (context) => const HomePage(),
        ),
      );
    }
  }
}

```

```

    }
  } catch (Error) {
    print(Error.toString());
  }
});
}

void signIn(BuildContext context, String email, String password) async {
  // ignore: unused_local_variable
  User user;
  // ignore: unused_local_variable
  String errorMessage;
  if (_formkey.currentState!.validate()) {
    try {
      isLoading = true;
      setState(() {});
      UserCredential userCredential =
        await FirebaseAuth.instance.signInWithEmailAndPassword(
          email: email,
          password: password,
        );
      // Get the device token
      String? deviceToken = await getDeviceToken();

      // Update the device token in Firestore
      FirebaseFirestore.instance
        .collection('users')
        .doc(userCredential.user!.uid)
        .update({
          'deviceToken': deviceToken,
        });
      route();
    } on FirebaseAuthException catch (e) {
      if (e.code == 'user-not-found') {
        // print('No user found for that email.');
```

ScaffoldMessenger.of(context).showSnackBar(
 SnackBar(
 content: Container(
 padding: const EdgeInsets.all(16),
 decoration: const BoxDecoration(
 color: Kactivecolor,
 borderRadius: BorderRadius.all(Radius.circular(20)),
),
 child: const Text(
 'email is not Register',
 style: TextStyle(
 fontWeight: FontWeight.bold, color: Colors.black),


```

        (route) => false);
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(
    content: Container(
      padding: const EdgeInsets.all(16),
      decoration: const BoxDecoration(
        color: Color(0xffC72c41),
        borderRadius: BorderRadius.all(Radius.circular(20)),
      ),
      child: const Text('The reset password link in sent'),
    ),
    behavior: SnackBarBehavior.floating,
    backgroundColor: Colors.transparent,
    elevation: 0,
    duration: const Duration(seconds: 10),
  ),
);
}).catchError((onError) {
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(
    content: Container(
      padding: const EdgeInsets.all(16),
      decoration: const BoxDecoration(
        color: Color(0xffC72c41),
        borderRadius: BorderRadius.all(Radius.circular(20)),
      ),
      child: Column(
        children: [
          const Text('Erro in Email Reset'),
          Text(onError.message),
        ],
      ),
    ),
    behavior: SnackBarBehavior.floating,
    backgroundColor: Colors.transparent,
    elevation: 0,
    duration: const Duration(seconds: 2),
  ),
);
});
}
}

```

Appendix D // Forgot Password

----- Forgot Password -----

```
import 'package:firebase_auth/firebase_auth.dart';

import '../exports.dart';

export 'package:material_design_icons_flutter/material_design_icons_flutter.dart';

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

  @override
  _ForgotScreenState createState() => _ForgotScreenState();
}

class _ForgotScreenState extends State<ForgotScreen> {
  // ignore: unused_field
  final bool _isObscure3 = true;
  bool visible = false;
  final _formkey = GlobalKey<FormState>();
  final TextEditingController emailController = TextEditingController();

  final _auth = FirebaseAuth.instance;
  void sendpasswordresetemail(String email) async {
    await _auth.sendPasswordResetEmail(email: email).then((value) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Container(
            padding: const EdgeInsets.all(16),
            decoration: const BoxDecoration(
              color: Color(0xffC72c41),
              borderRadius: BorderRadius.all(Radius.circular(20)),
            ),
            child: const Text('The password reset link was sent'),
          ),
          behavior: SnackBarBehavior.floating,
          backgroundColor: Colors.transparent,
          elevation: 0,
          duration: const Duration(seconds: 10),
        ),
      );
      // ignore: invalid_return_type_for_catch_error
    }).catchError((onError) => ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        content: Container(
```

```

padding: const EdgeInsets.all(16),
decoration: const BoxDecoration(
  color: Color(0xffC72c41),
  borderRadius: BorderRadius.all(Radius.circular(20)),
),
child: Column(
  children: [
    const Text('Error in Email Reset'),
    Text(onError.toString()),
  ],
),
),
behavior: SnackBarBehavior.floating,
backgroundColor: Colors.transparent,
elevation: 0,
duration: const Duration(seconds: 10),
),
));
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      elevation: 0,
    ),
    body: SingleChildScrollView(
      child: Center(
        child: Column(
          children: <Widget>[
            SizedBox(
              // color: Colors.orangeAccent[700],
              width: MediaQuery.of(context).size.width,
              height: MediaQuery.of(context).size.height * 0.70,
              child: Center(
                child: Container(
                  margin: const EdgeInsets.all(12),
                  child: Form(
                    key: _formkey,
                    child: Column(
                      mainAxisAlignment: MainAxisAlignment.center,
                      crossAxisAlignment: CrossAxisAlignment.start,
                      children: [
                        const SizedBox(
                          height: 30,
                        ),
                        const Text(

```

```

"Reset",
style: TextStyle(
  fontWeight: FontWeight.bold,
  color: Colors.black,
  fontSize: 40,
),
),
const SizedBox(
  height: 20,
),
TextFormField(
  controller: emailController,
  decoration: InputDecoration(
    prefixIcon: const Icon(
      MdiIcons.email,
      color: Kactivecolor,
      size: 22,
    ),
    contentPadding: const EdgeInsets.all(18),
    fillColor: Colors.black,
    enabledBorder: OutlineInputBorder(
      borderSide: const BorderSide(
        color: Kinactivetextcolor, width: 1),
      borderRadius: BorderRadius.circular(12),
    ),
    focusedBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: const BorderSide(
        color: Kinactivetextcolor, width: 1)),
    hintText: "Enter Your Email",
  ),
  validator: (value) {
    if (value!.isEmpty) {
      return "Email cannot be empty";
    }
    bool emailvalidator = RegExp(
      r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*-+./=?^_`{|}~]+@a-zA-Z0-9]+\.[a-
zA-Z]+")
      .hasMatch(value);
    if (emailvalidator) {
      return ("Please enter a valid email");
    }
    return null;
  },
  onSave: (value) {
    emailController.text = value!;
  },

```

```

        keyboardType: TextInputType.emailAddress,
      ),
      const SizedBox(
        height: 25,
      ),
      Center(
        child: custombtn(
          colorbtn: Kactivecolor,
          colortxt: Colors.white,
          txtbtn: "Reset Password",
          onPress: () {
            sendpasswordresetemail(emailController.text);
          },
        )),
      const SizedBox(
        height: 10,
      ),
      Visibility(
        maintainSize: true,
        maintainAnimation: true,
        maintainState: true,
        visible: visible,
        child: Center(
          child: Container(
            child: const CircularProgressIndicator(
              color: Colors.black,
            )),
        )),
    ],
  ),
),
),
),
),
),
),
),
),
),
),
),
);
}
}

```


Appendix E // Cart Items Screen

```
----- Cart Items Screen -----

import 'package:cached_network_image/cached_network_image.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:laundry_order_app/provider.dart';
import 'package:provider/provider.dart';

class CartItem extends StatefulWidget {
  // final product;

  final product;

  const CartItem({super.key, required this.product, required this.index});

  final int index;

  @override
  State<CartItem> createState() => _CartItemState();
}

class _CartItemState extends State<CartItem> {
  int quantity = 1;

  int totalPrice = 0;

  @override
  Widget build(BuildContext context) {
```

```

return Consumer<CartProvider>(builder: (context, value, _) {

  return Container(

    height: 120,

    decoration: BoxDecoration(

      borderRadius: BorderRadius.circular(8),

      // ignore: deprecated_member_use

      color: Theme.of(context).bottomAppBarColor,

      border: Border.all(

        color: Theme.of(context).colorScheme.secondary.withOpacity(0.08),

      ),

    ),

    child: Row(

      crossAxisAlignment: CrossAxisAlignment.stretch,

      children: <Widget>[

        Container(

          width: 120,

          height: double.infinity,

          child: Image(

            image: NetworkImage(widget.product['imageUrl']),

          ),

        ),

        Expanded(

          child: Padding(

            padding: const EdgeInsets.symmetric(

              horizontal: 10.0,

```

```

        vertical: 4.0,
    ),
    child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: <Widget>[
            Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: <Widget>[
                    Expanded(
                        child: Column(
                            crossAxisAlignment: CrossAxisAlignment.start,
                            children: <Widget>[
                                Padding(
                                    padding:
                                        const EdgeInsets.symmetric(vertical: 8.0),
                                    child: Text(
                                        widget.product['clothName'].toString(),
                                        style: GoogleFonts.roboto(fontSize: 18),
                                        overflow: TextOverflow.ellipsis,
                                    ),
                                ),
                                RichText(
                                    text: TextSpan(
                                        text: '\$',
                                        style: TextStyle(

```

```

        color: Theme.of(context)
          .colorScheme
            .secondary
              .withOpacity(0.8),
        fontSize: 17,
        fontWeight: FontWeight.w400,
      ),
      children: [
        TextSpan(
          text: '${widget.product['initialPrice']}',
          style: TextStyle(
            color: Theme.of(context)
              .colorScheme
                .secondary,
            letterSpacing: 1.2,
            fontSize: 18,
            fontWeight: FontWeight.w900,
          ),
        ),
      ],
    ),
  ],
),
),
CustomButton(

```

```

// ignore: deprecated_member_use

icon: Icon(FontAwesomeIcons.trashAlt),

onPressed: () => showDialog(
  context: context,
  barrierDismissible: false,
  builder: (_) => AlertDialog(
    actions: <Widget>[
      TextButton(
        onPressed: () {
          setState(() {
            value.removeFromCart(widget.product);
            Navigator.of(context).pop();
          });
        },
        child: Text('Yes'),
      ),
      TextButton(
        onPressed: () {
          setState(() {
            Navigator.of(context).pop();
          });
        },
        child: Text('No'),
      ),
    ],
  ),
),
],

```

```

        title: Text('Confirm'),

        content: Text(
          'Are you sure to remove this item from your cart?',

        ),

      ),

    ),

  ],

),

Row(
  children: <Widget>[
    CustomButton(
      icon: Icon(Icons.remove),
      onPressed: () {
        if (quantity > 0) {
          setState(() {
            value.decreaseQuantity(widget.index);
          });
        }
      },
    ),
    Padding(
      padding: const EdgeInsets.all(10.0),
      child: Text(
        "${widget.product['quantity']}",
        style: TextStyle(

```

```

        color: Theme.of(context).colorScheme.secondary,

        letterSpacing: 1.2,

        fontSize: 18,

        fontWeight: FontWeight.w900,

    ),

),

),

CustomButton(
    onPressed: () {
        setState(() {
            //quantity++;

            value.changeQuantity(widget.index);

        });
    },
    icon: Icon(Icons.add),

),

],

)

],

),

)),

],

),

);

});

}

```

```

}

class CustomButton extends StatelessWidget {

  const CustomButton({ Key? key, this.onPressed, required this.icon })

    : super(key: key);

  final Icon icon;

  final void Function()? onPressed;

  @override
  Widget build(BuildContext context) {
    return Material(
      type: MaterialType.transparency,
      child: InkWell(
        borderRadius: BorderRadius.circular(50),
        onTap: onPressed,
        child: Container(
          width: 35.0,
          height: 35.0,
          decoration: BoxDecoration(
            color: Theme.of(context).colorScheme.secondary.withOpacity(0.05),
            borderRadius: BorderRadius.circular(50.0),
          ),
          child: icon,
        ),
      ),
    ),
  ),

```



```
);  
}  
}
```