



Products tracking application: **Haseef**

Graduation project 2 report

Prepared by:

Alanoud Alissa

Farah Alarfaj

Ghada Alharbi

Hayfa Alshareef

Raghad Althuwaini

Reem Alsenaidy

Supervisor:

Dr. Ahmed Ibrahim

A Graduation Project Report Submitted to College of Computer and Information Sciences
at PNU in Partial Fulfillment of the Requirement for the Degree of Bachelor of Computer
Sciences PNU – CCIS

2021

Acknowledgments

First of all, this report has been done with Allah's grace and mercy, praise be to him for success and relief. We would like to express our appreciation and thanks to our supervisor Dr. Ahmed Ibrahim for his encouragement, valuable advisement, and support during our preparation of this report. We would also like to acknowledge Princess Nourah bint Abdulrahman University and the department of computer science for supporting the educational and professional journey of students and providing them the opportunities to develop their specialized skills and knowledge by establishing partnerships with prominent bodies in computer science for training and education in line with 2030 vision. Finally, we would like to express our gratitude to our families for their love and support.

Content

Abstract	8
1. Introduction	9
1.1 Problem Statement & Significance	10
1.2 Solution	10
1.3 Project Domain & Limitations	11
2. Background information & related work	12
2.1 Background information	13
2.2 Related work survey	15
2.3 Proposed & Similar Systems Comparison	20
3. System analysis	22
3.1 Requirements specification	23
3.2 Requirements analysis	26
3.2.1 Use case diagram	26
3.2.2 Class diagram	31
3.2.3 Sequence diagram	32
3.2.4 Data flow diagram	40
4. System design	41
4.1 System architecture	42
4.2 User interface design	43
5. Implementation	52
5.1 Introduction	53

5.2 Implementation Requirements	53
5.3 Implementation details	54
5.4 I/O Screens	60
6. Testing	68
6.1 Test plan	69
6.2 Test Cases	71
6.3 Test result	79
7. Conclusion	80
7.1 Evaluation	81
7.2 Future work	81
8. References	82
9. Appendices	83

Table of Figures



Figure 2.2.1 Walmart logo	15
Figure 2.2.2 Recognize out-of-stock event using image processing	15
Figure 2.2.3 Trigger an alert in the event of out-of-stock	15
Figure 2.2.4 Kannita logo	17
Figure 2.2.5 Kannita business design	17
Figure 2.2.6 User interface design for mobile application	17
Figure 2.2.7 User interface design for mobile, desktop, and tablet applications	18
Figure 2.2.8 AWM smart shelf logo	19
Figure 2.2.9 Content management interface	19
Figure 3.2.1 Use case diagram for Haseef application	26
Figure 3.2.2 Class diagram for Haseef application	31
Figure 3.2.3.1 Log-in sequence diagram for Haseef application	32
Figure 3.2.3.2 Statistics sequence diagram for Haseef application	33
Figure 3.2.3.3 Setting sequence diagram for Haseef application	34
Figure 3.2.3.4 notification sequence diagram for Haseef application	35
Figure 3.2.3.5 Products staff sequence diagram for Haseef application	36
Figure 3.2.3.6 Restock staff sequence diagram for Haseef application	37
Figure 3.2.3.7 Products admin sequence diagram for Haseef application	38
Figure 3.2.3.8 Staff admin sequence diagram for Haseef application	39
Figure 3.2.4 Data flow for Haseef System	40

Table of Figures

Figure 4.1 System architecture	42
Figure 4.2.1 The first interface	43
Figure 4.2.2 log-in interface	43
Figure 4.2.3 Home page interface for staff	44
Figure 4.2.4 Notifications interface	44
Figure 4.2.5 Profile interface	45
Figure 4.2.6 Dairy products line restock interface	46
Figure 4.2.7 Juices line restock interface	46
Figure 4.2.8 Chocolate line restock interface	47
Figure 4.2.9 Statistics	47
Figure 4.2.10 best-selling product of the day	48
Figure 4.2.11 least-selling product of the day	48
Figure 4.2.12 Products interface for staff	48
Figure 4.2.13 Home page interface for admin	49
Figure 4.2.14 Staff list interface	49
Figure 4.2.15 Adding new staff member interface	50
Figure 4.2.16 Products interface for admin	50
Figure 4.2.17 Adding new product interface	51
Figure 5.3.1 Realtime database	54
Figure 5.3.2 Log-in code	55
Figure 5.3.3 Delete Staff code	56

Figure 5.3.4 Update product code	56
Figure 5.3.5 Add new product code	57
Figure 5.3.6 Connect to WI-FI router code	58
Figure 5.3.7 Connect database code	59
Figure 5.3.8 Average weight code	59
Figure 5.4.1 Welcome I/O screen	60
Figure 5.4.2 Log-in I/O screen	60
Figure 5.4.3 Home page ‘user’ I/O screen	61
Figure 5.4.4 Notification I/O screen	61
Figure 5.4.5 Settings I/O screen	62
Figure 5.4.6 Restock I/O screen	62
Figure 5.4.7 Statistics I/O screen	63
Figure 5.4.8 pop-up I/O screen	63
Figure 5.4.9 Product I/O screen	64
Figure 5.4.10 Home page ` Admin` I/O screen	65
Figure 5.4.11 Staff I/O screen	65
Figure 5.4.12 Add staff I/O screen	66
Figure 5.4.13 Product `Admin` I/O screen	67
Figure 5.4.14 Add product I/O screen	67
Figure 6.1.1 Haseef testing process	69

Abstract

Recently, the technical solutions in the commerce field have become a topic of interest, as this field plays a major role in the economy. Also, these solutions can be an indication of the technological improvement in the place they are in. In our system, we attempt to offer a solution for managing the products in the commercial shops to improve the selling experience.

As commercial shops become larger and have more products to manage, the employees may find monitoring each product's availability on the shelves manually difficult and time-consuming. Our proposed solution for this problem is the Internet of Things (IoT) system called "Haseef" consist of microcontroller and Android application, where appropriate sensors and weight connected to the microcontroller and installed on the shelves to track the number of products on it, check the cleanliness and indicate the location. Whenever a shelf is empty or dirty, a notification will be sent through the application indicating the location of that shelf. Also, we can extract inferences that help to improve the sales by analyzing the sales data using the restock notifications log in the application, such as knowing the best-selling products, the right time to sell a specific product, and knowing if the product's location can affect its purchase.

We designed the use case, class, sequence, data flow and system architecture diagrams depending on the requirements specification to clarify the system processes and have applied the solution.

Haseef has fulfilled its objective, and all of its functions are behaving correctly. Also, it has a friendly and simple UI that effectively serves user interaction.

Keywords:

Shelves.

IoT.

Monitoring System.

Microcontroller-based project

Counting items.

CHAPTER 1

INTRODUCTION

Introduction

This chapter will outline the problem statement to explain the choice of this project subject and it will also show the proposed solutions. In addition, it shall discuss the domain and some relevant limitations.

1.1 Problem Statement & Significance:

Today the Internet of Things (IoT) is an emerging technology across the world. In our present time, monitoring the shelves in the stores has become a hard job to do. Here we present an issue that many stores owners face. It is how to find out which product is left on the store shelves, and which product is not available. Besides, the responsible employee takes a lot of effort and time to do this job.

However, the only time for the employee to find out if the shelves are empty is at the closing time. Thus, we need a technology that benefits this issue, reduces administrator effort and helps make the quantity of products more accurate. Furthermore, one of the problems that store owners may face, is not knowing what the best-selling and least-selling commodity is. Through this technology the management team will have the ability to predict the rate of sales of all individual products and ensure the availability of the stock.

1.2 Solution (System)

Through Internet of Things (IoT) technological innovation, the Haseef application aims to strategically manage items in supermarket by applied sensors on the shelves to gather data, such as the number of items on a shelf.

This technology will improve employees jobs, to make their job more interesting, to help them alleviate some of mundane task. In this regard haseef can collect information about how many customers pick up merchandise versus how many sales occur. This helps in identifying what goods appeal to people and align procurement with consumer purchasing habits and this information can be used optimize the assortment of merchandise. Where it can help the management to increase their profits and by that they will satisfy the shareholders through a complete process. To achieve this goal, learn about sensor, Arduino, gather information and analyze them, the final product will have a user-friendly interface.

1.3 Project Domain & Limitations

The domain of the project is applying the concept of Internet of things (IoT) to offer a solution for products managing in commercial stores. Internet of Things (IoT) describes the network of physical objects "things" that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet [1].

Some of the limitations of the project are that the system is written only in two programming languages which are Java for the mobile application and C++ for the microcontroller. Also, it can be used only for android operating system users and needs a good Internet connection.

The targeted users of this system are only the workers in markets, groceries, shops, etc., who can read English as it is the used language in the user interface.

CHAPTER 2

BACKGROUND INFORMATION & RELATED WORK

2. Background information & related work

2.1 Background information

As students, we have a good background about programming languages to propose our project such as Java, C++, and SQL database.

In addition, information about database like SQL for storing, manipulating, and retrieving data stored in a relational database which is needed for our application. As we know, the database will connect with interfaces, so we have done a prototype (UI) user interface before.

Also, we chose Java and C++ language because it offers the following advantage:

- Java
 - 1. Java was designed to be easy to use.
 - 2. object-oriented.
 - 3. Platform-independent.
 - 4. Memory allocation [2].
- C++
 - 1. Rich library support.
 - 2. Mid-level language
 - 3. Pointer and direct Memory-Access.
 - 4. object-oriented.

We learned about Arduino, and we have done simple programming (turn on/off the light), and we continue learning about it more we will use the Arduino to apply our solution. Arduino is an open-source electronics platform based on easy-to-use hardware and software; it is made for anyone interested in creating interactive objects or environment. You can tell your Arduino what to do by sending a set of

instructions to the microcontroller on the board. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, this flexibility combined with the fact that the Arduino software is free. the very popular Arduino board is called the Arduino UNO.

2.2 Related work survey

In this section, we will present 3 related works to our project which are: Walmart IRL, Kannita (shelf monitor platform) and AWM Smart shelf.



Fig 2.2.1
Walmart logo

1. Walmart IRL

Walmart IRL stands for Walmart intelligent retail lab and is set up to gather information about what is happening inside the store through array of sensors, cameras, and processors.

Using combination of cameras and real-time analytics, IRL can detect the product availability on the shelf and recognize it. Also, it can compare the quantities on the shelf to the upcoming sales demand to ensure that the sales process runs efficiently. In addition, IRL will automatically trigger out-of-stock notifications to internal apps that alert associates when to re-stock, so items are available on shelves all the time [3].



Fig 2.2.2
Recognize out-of-stock event using image processing



Fig 2.2.3
Trigger an alert in the event of out-of-stock

Strength points:

- It uses real-time analytics.
- It has high recognition performance.
- Has a robust system that relies on different technologies such as Artificial Intelligence, Internet of Things, robotics, etc.

Weakness points:

- It is dedicated only to the Walmart branch in New York city.
- It only supports the English language.



Fig 2.2.4

Kannita's logo

2. Kannita

Kannita provides real-time, actionable insights on shelf contents and consumer behavior. The platform tracks shelf items using image processing and presents real-time data to the store's sales team. It boosts staff performance and training, provides planogram compliance warnings, predicts refill needs, and offers advanced big data tools.

Kannita's powerful tools help retailers eliminate out-of-stock events, improve their business results with in-store operations predictive analysis using machine learning, and maximize store ROI. Backed by Coca-Cola, Kannita is a part of the Coca-Cola Founders Platform [4].

Logo and user interface design:

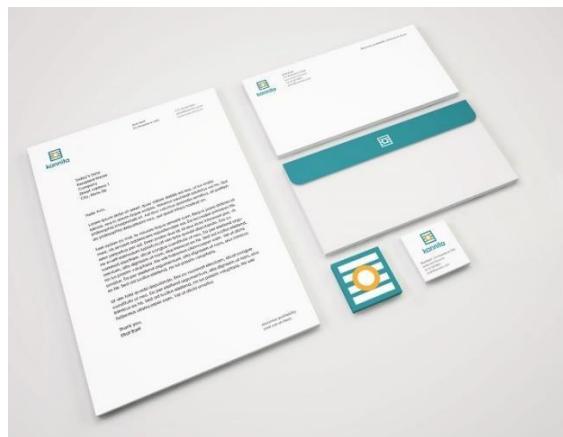


Fig 2.2.5

Kannita's business design

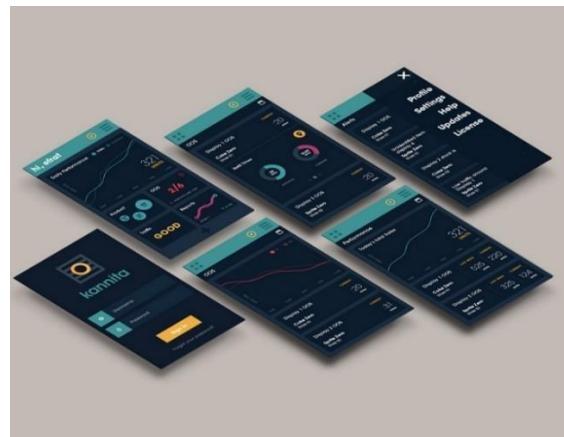


Fig 2.2.6

User interface design for mobile application



Fig 2.2.7

User interface design for mobile, desktop, and tablet applications

Strength points:

- It uses predictive analysis to improve business results.
- Friendly and clear user interface.

Weakness points:

- It only supports the English language.



Fig 2.2.8
AWM smart shelf logo

3. AWM Smart Shelf

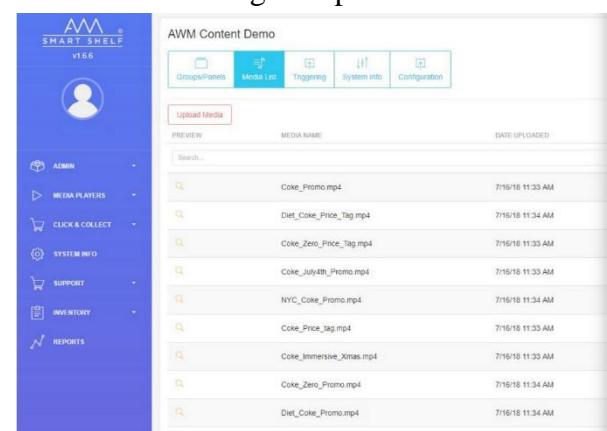
The platform utilizes the shelf's edge combined with high-definition optical sensors to automate intelligence into on-the-shelf inventory [5].

The platform has several major solutions, some of which are:

- Automated inventory intelligence: to view and track any product in a retail environment using image processing through the camera. A notification will be triggered in the event of out-of-stock.

Demographic engine: utilizes computer vision with machine learning to capture the demographics of the shoppers and use these data to play advertisements clips on the shelf's edge based on demographic criteria.

- Content management system: to easily manage the store's inventory, prices, and content.



PREVIEW	MEDIA NAME	DATE UPLOADED
	Coke_Promo.mp4	7/16/18 11:33 AM
	Diet_Coke_Price_Tag.mp4	7/16/18 11:34 AM
	Coke_Zero_Price_Tag.mp4	7/16/18 11:33 AM
	Coke_July4th_Promo.mp4	7/16/18 11:33 AM
	NYC_Coke_Promo.mp4	7/16/18 11:34 AM
	Coke_Price_Tag.mp4	7/16/18 11:33 AM
	Coke_Immersive_Xmas.mp4	7/16/18 11:33 AM
	Coke_Zero_Promo.mp4	7/16/18 11:34 AM
	Diet_Coke_Promo.mp4	7/16/18 11:34 AM

Fig 2.2.9
Content management interface

Strength points:

- Supports advertisements based on the shopper's demography.
- Real-time image processing.
- It triggers an alert in the event of less than 20% available for the product.

Weakness points:

- It only supports the English language.

1. 2.3 Proposed & Similar Systems Comparison:

Survey similar programs, systems, or applications to yours, and how yours is differentiated by constructing a comparison table as follows

features	Haseef	Walmart	AWM	Kannita
Problems Solved	Sensors monitoring the shelves in the stores if they dirty or empty and can monitor the best-selling and least-selling products. It will reduce the employee's effort and time.	Cameras can see if the employee needs to bring the products to the front shelves, and able to see if the products are sitting too long on the shelves.	Track any product and results can be provided as quickly as near-real-time and as frequently as hourly or even in response to shopper activity. It is simply the most robust inventory tool available.	Helping retailers eliminate Out-of-stock events, improve their business results with in-store operations, predictive analysis, and maximize store ROI.
Compatibility	Android.	Website, IOS and Android.	Website, IOS and Android.	Website, IOS and Android.
Website/Application interface	Simple to interact with, easy.	Friendly, simple to interact with.	Useful, clear.	Simple, clear.
Interface language	English	English	English	English
Registration	Yes	Yes	Yes	Yes

Output	Reducing costumers, and employees time and effort, help the suppliers to control the amount of the products.	Cameras are not monitoring the costumers. cameras are checking the level of inventory and replenishing reasons.	Knowing which products have the highest and the lowest sales and decreasing the labor cost of employee walking around the store.	New way to minimize Out Of Stock, and maximize On Shelf Availability (OSA) and performance.
Reminder alert	Yes	Yes	Yes	Yes
Hardware included	Weight, sensors.	Camera, sensors.	Camera, sensors.	Sensors.

CHAPTER 3

SYSTEM ANALYSIS

3. System analysis

3.1 Requirements specification

Functional requirements:

Staff:

- 1- The staff member shall be able to go through the product to see what products are available in the store.
- 2- The staff member shall be able to see all the products in the restock page, if one of the shelves is empty, the staff member will restock the shelf and then click on the check mark button.

Admin:

- 1- The manager shall be able to add or update the information of the staff.
- 2- The manager shall be able to display all the registered staff.
- 3- The manager shall be able to add new products and display all products.

Shared functional requirements between the staff and manager:

- 1- The staff member and the manager shall be able to log-in into the system by entering their e-mail and password.
- 2- The staff member and the manager shall be able to log-out from the entire system.
- 3- The staff member and the manager shall be able to view the statistic of each products.

System:

- 1- The system must be able to check e-mail and password validity.
- 2- The system shall be able to send a notification to the staff member if one of the shelves is empty or half empty.

Non-functional Requirement:

- 1- Availability: the system shall be available 24 hours per day, 365 days per year. The system must not lose any data.
- 2- In the case of unplanned system downtime, all features shall be available again after one working day.
- 3- Ease of learning: the system shall be easy to learn for both novices and users with experience from similar systems.
- 4- Task efficiency: the system shall be efficient for the frequent user.
- 5- Performance: The performance of the Application can be determined by its responsive time, time to complete the given task.
- 6- The Application is made to start up it shall not take more than three seconds to load the initial screen, also it should make sure the app will not hindrances to the User Input.
- 7- Responsiveness: When the employee selects any product, it shall be easy for her/him to see the information about it with a few touches, and this should happen in fewer seconds or instantaneous with a user touch.

Software requirements:

1. C++: for uploading the program's functions on the microcontroller and handling its inputs.
2. Android studio: is an integrated development environment (IDE) for Google's Android operating system, we will use it to design and develop the mobile application [6].
3. firebase: Firebase is a platform developed by Google for creating mobile and web applications [7].

Hardware requirements:

1. A load cell is a force transducer. It converts a force such as tension, compression, pressure, or torque into an electrical signal that can be measured and standardized. As the force applied to the load cell increases, the electrical signal changes proportionally [8].
2. HX711 is an electronic scale module, whose working principle is to convert the measured changes in resistance value changes, through the conversion circuit into electrical output [9].
3. ESP8266 is a low-cost Wi-Fi chip, with full TCP / IP stack and microcontroller capacity [10].
4. PC with a high-quality processor to develop the system.

All the above requirements are proposed, and they will be finalized during the development stage.

3.2 Requirements analysis.

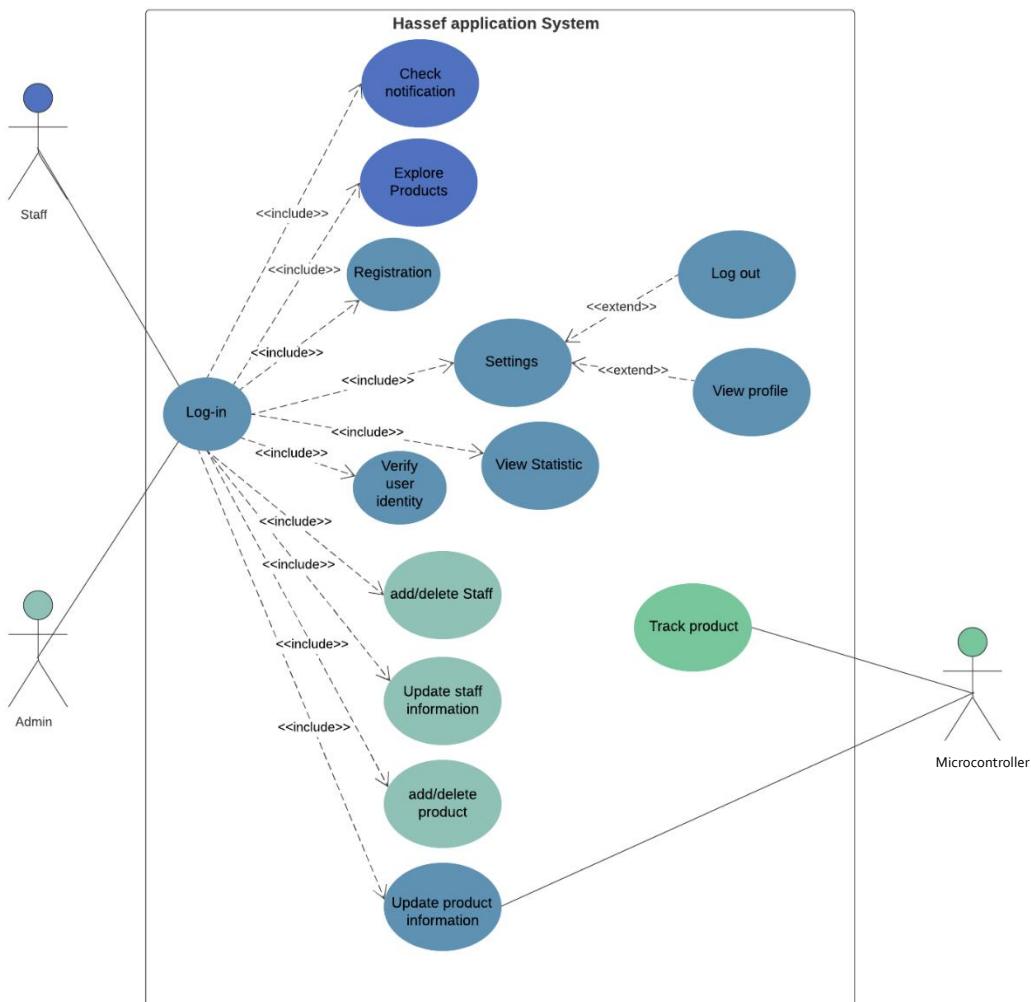
3.2.1 Use case diagram

The use case diagram in Figure 3.2.1.1 shows the interaction between the system and the actors, which are Staff, Admin, and microcontroller.

The staff can apply several actions such as: explore products and check notifications.

The admin can apply addition, deletion, updating on the staff and products information in the database. Manager and staff have common actions, which are log-in, view statistics, view profile and log-out.

The microcontroller can track the product and update its information in the database.



3.2.1.1
Use case diagram for Haseef application

Use case descriptions:

Table 3.2.1.1
Log-in.

Use case name	Log-in
Actor	Admin and staff
Description	admin and staff need to log in the system successfully.
Precondition	Both should have a valid e-mail number and password to login into the system.
Postcondition	The system will show different home pages for each.

Table 3.2.1.2
Settings.

Use case name	Settings.
Actor	Admin and staff.
Description	Settings is where the actor can view its profile and log-out from the app.
Precondition	Log-in to the app and click on the settings icon.
Postcondition	View profile or log-out.

Table 3.2.1.3
Statistics.

Use case name	View statistics.
Actor	Admin and staff.
Description	Actor can view best and least seller products in day, month, or year.
Precondition	Log-in to the app and click on the statistics icon.
Postcondition	View products sales statistics.

Table 3.2.1.4
View profile.

Use case name	View profile
Actor	Admin and staff.
Description	The user can view the staff name and ID.
Precondition	Log-in to the system, entering setting page and choose view profile.
Postcondition	The system will show the profile page.

Table 3.2.1.5
Add/ remove.

Use case name	Add/remove product.
Actor	Admin.
Description	Actor can add a product to the system database by entering its information and can remove it.
Precondition	Log-in to the app, enter the products page and click on the "plus" or "garbage" icon.
Postcondition	Product has been added in or deleted from the database.

Table 3.2.1.6
Update product information.

Use case name	Update product information.
Actor	Admin and microcontroller.
Description	Actors can update the product information in the database.
Precondition	For the admin, log-in to the app, enter the products page and click on the "update" icon. For the microcontroller, load cell and Wi-Fi shield are loaded on it.
Postcondition	Product information is updated in the database.

Table 3.2.1.7
Add/ remove staff.

Use case name	Add/remove staff.
Actor	Admin.
Description	Actor can add a staff member to the system database by entering his information to enable him to access the system and can remove him.
Precondition	Log-in to the app, enter the staff page and click on the "plus" or "garbage" icon.
Postcondition	Staff member has been added in or deleted from the database.

Table 3.2.1.8
Update staff information.

Use case name	Update staff information.
Actor	Admin.
Description	Actor can update the staff information (ID, name, location) in the database.
Precondition	Log-in to the app, enter the staff page and click on the "update" icon.
Postcondition	Staff information is updated in the database.

Table 3.2.1.9
Track product.

Use case name	Track product.
Actor	Microcontroller.
Description	Tracking the products.
Precondition	Load cell and WI-FI shield with stable connection are located on the microcontroller
Postcondition	Update the database.

Table 3.2.1.10
Explore products.

Use case name	Explore products.
Actor	Staff.
Description	The staff can see a list of all products by their line.
Precondition	Log-in to the system successfully.
Postcondition	The system will show products page.

Table 3.2.1.11
Check notifications.

Use case name	Check notifications.
Actor	Staff.
Description	Staff can see all alerts send to him/her.
Precondition	Log-in to the system successfully.
Postcondition	The system will show all the alerts.

Table 3.2.1.12
Log-out.

Use case name	Log-out.
Actor	Admin and staff.
Description	Log out from the system.
Precondition	Log-in to the system and entering the setting page to choose to log-out.
Postcondition	Log-out from the system successfully.

3.2.2 Class diagram

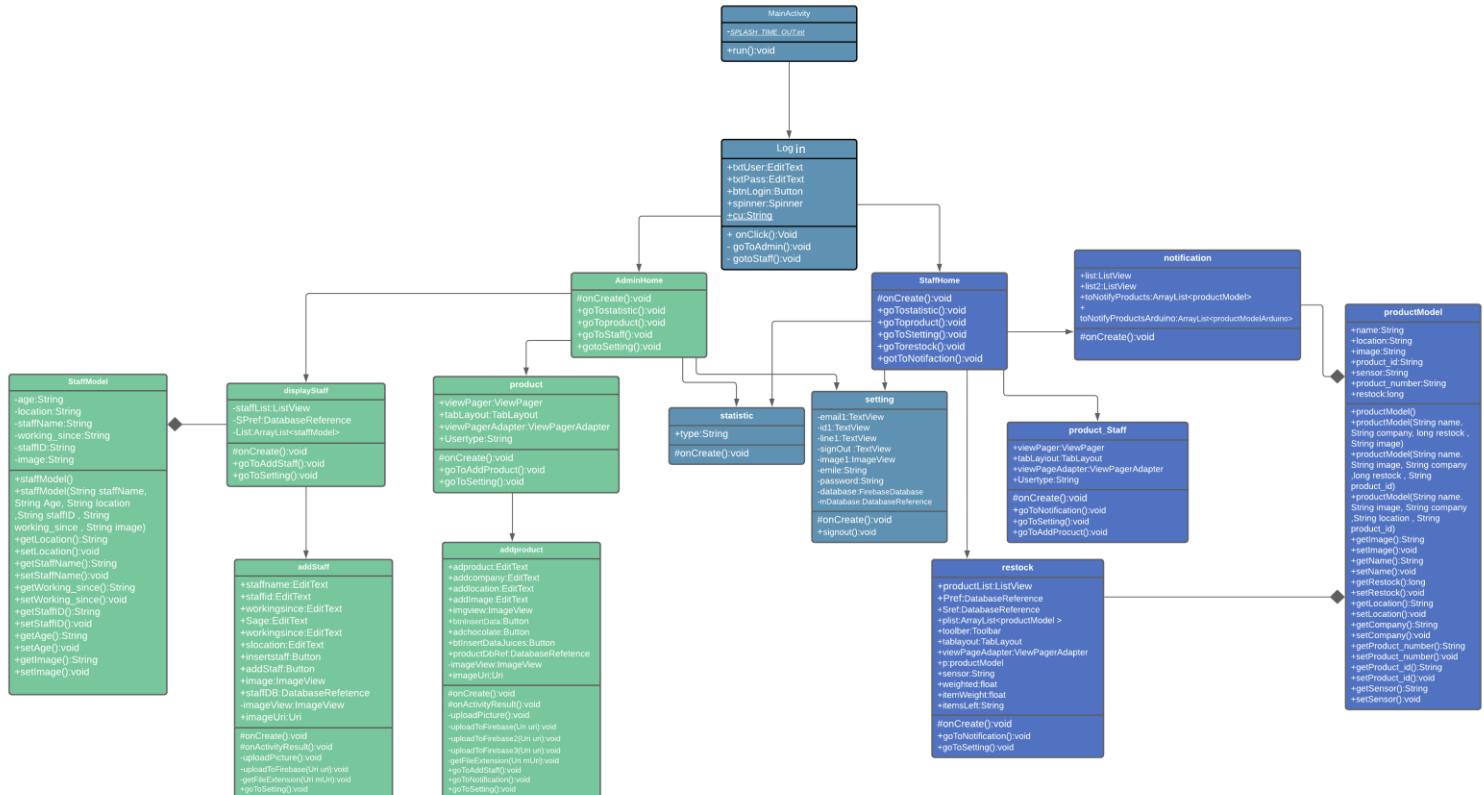


Figure 3.2.2.1
Class diagram for Hasseef application.

3.2.3 Sequence diagram

1– Log-in:

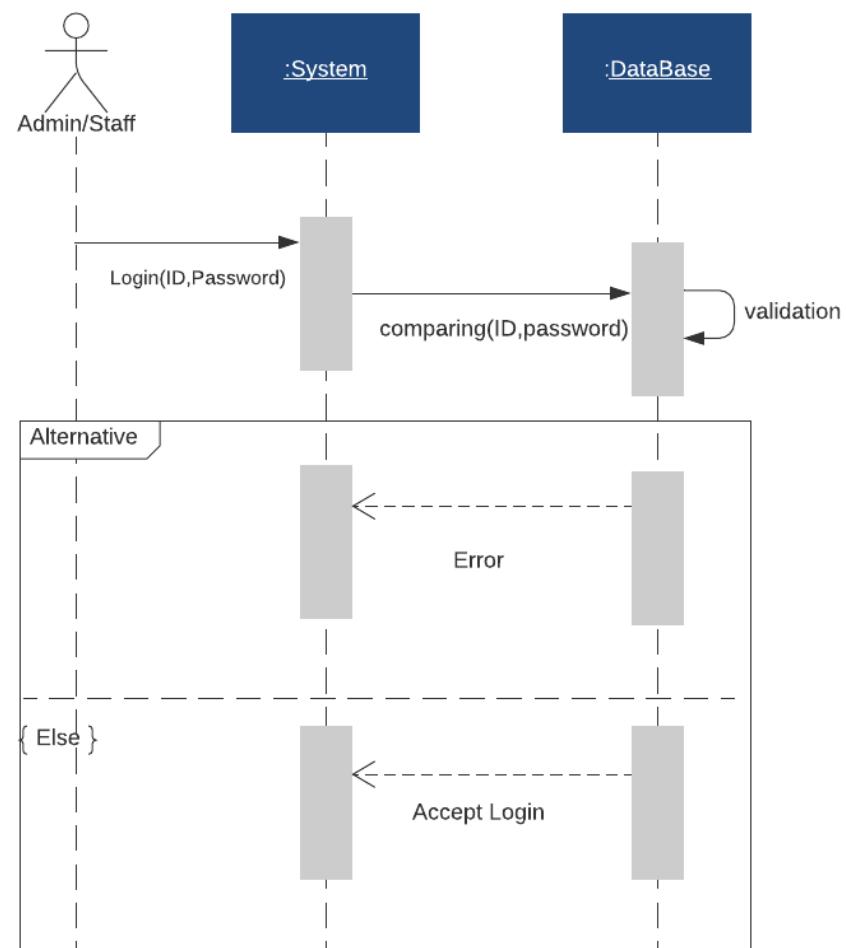


Figure 3.2.3.1
Log-in sequence diagram for Haseef application.

2- Statistics:

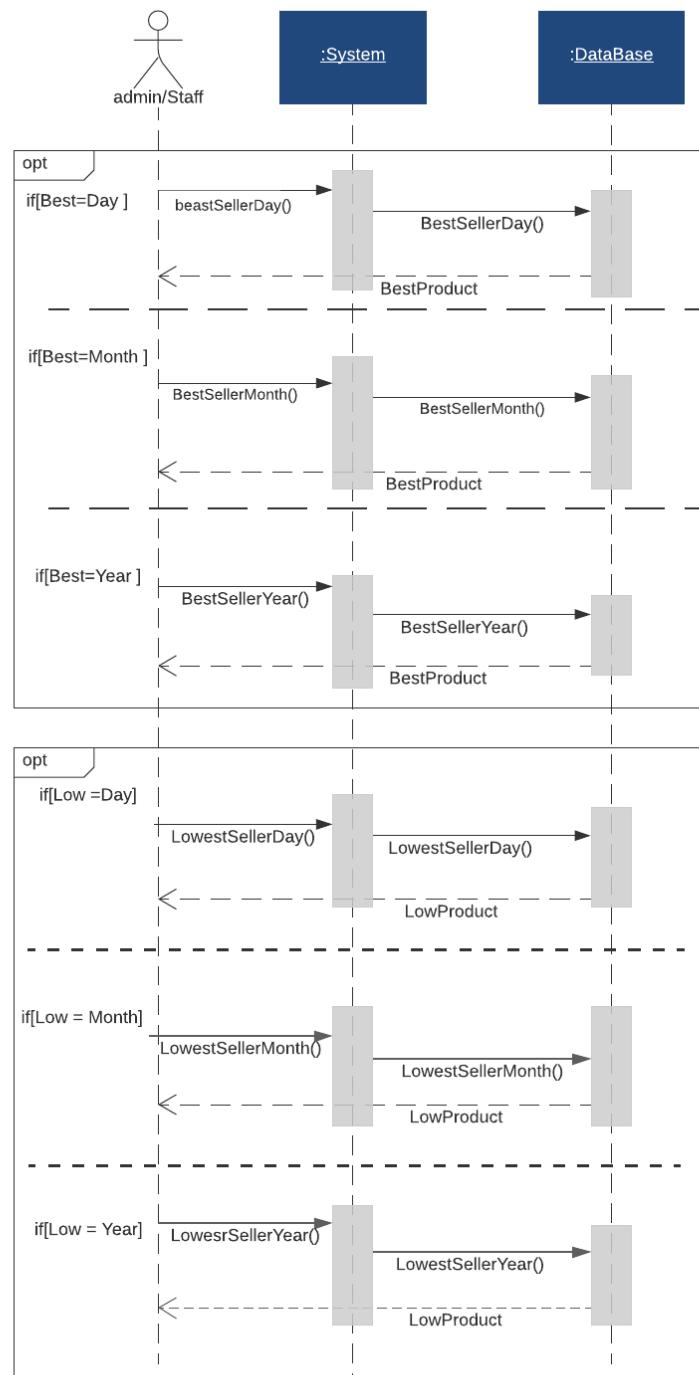


Figure 3.2.3.2
Statistics sequence diagram for Haseef application.

3- Setting:

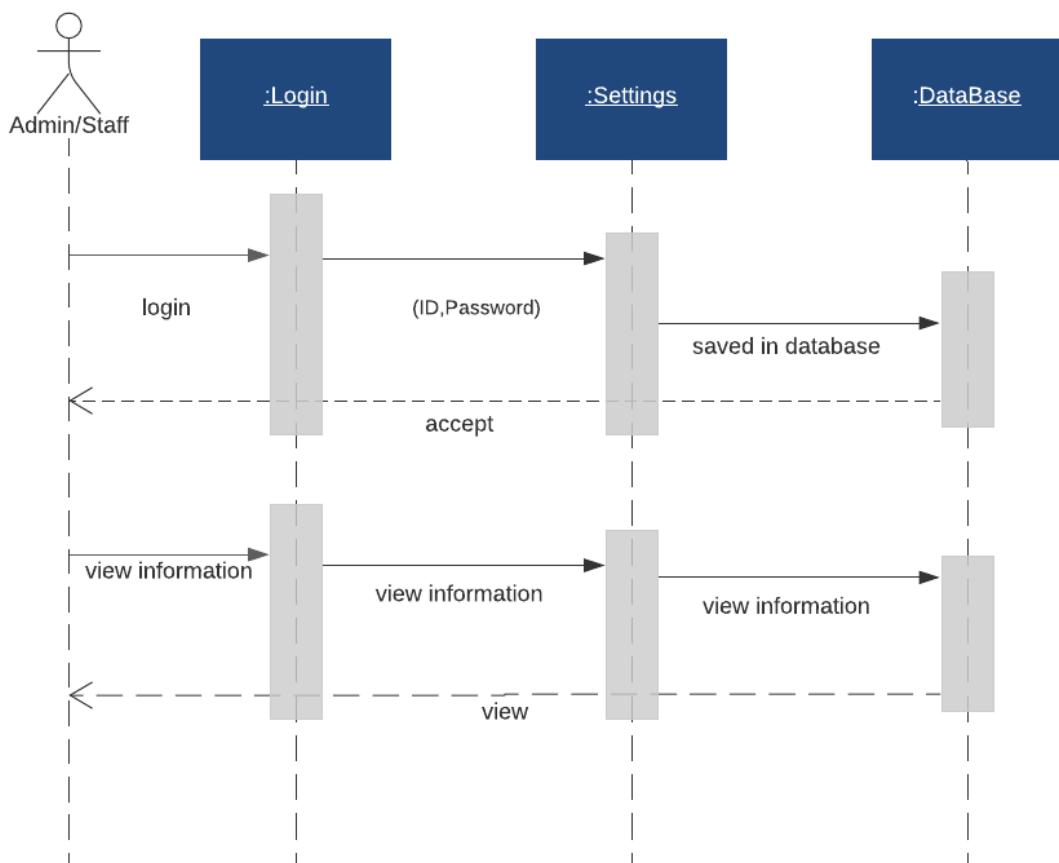


Figure 3.2.3.3
Setting sequence diagram for Haseef application.

4- Notification:

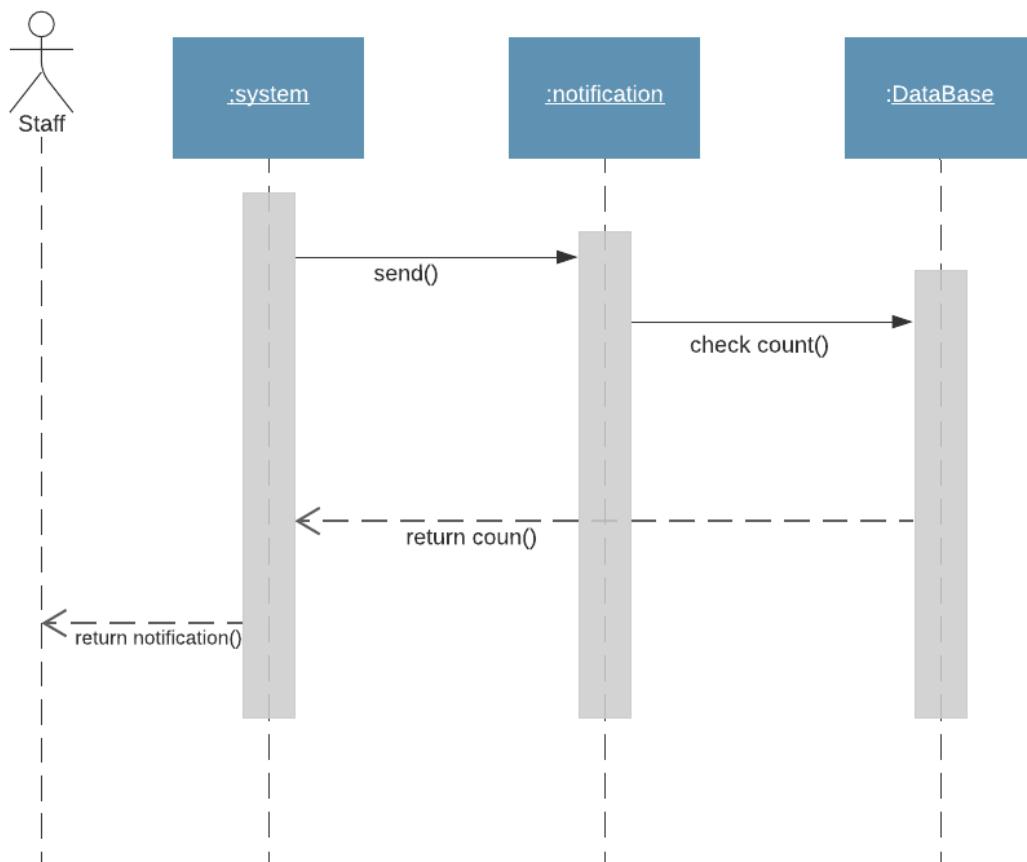


Figure 3.2.3.4
Notification sequence diagram for Haseef application.

5-Products staff:

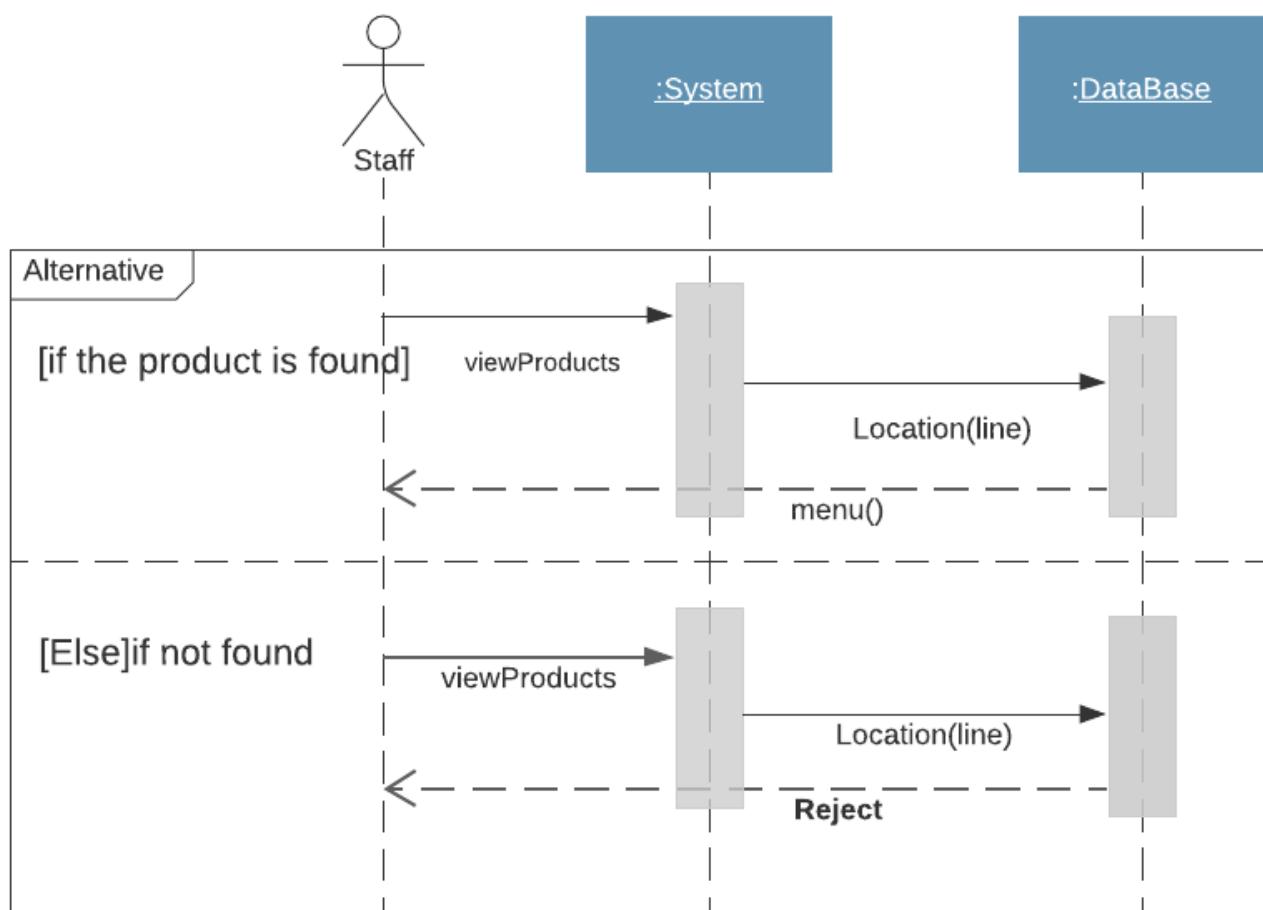


Figure 3.2.3.5
Products staff sequence diagram for Haseef application.

6- Restock staff:

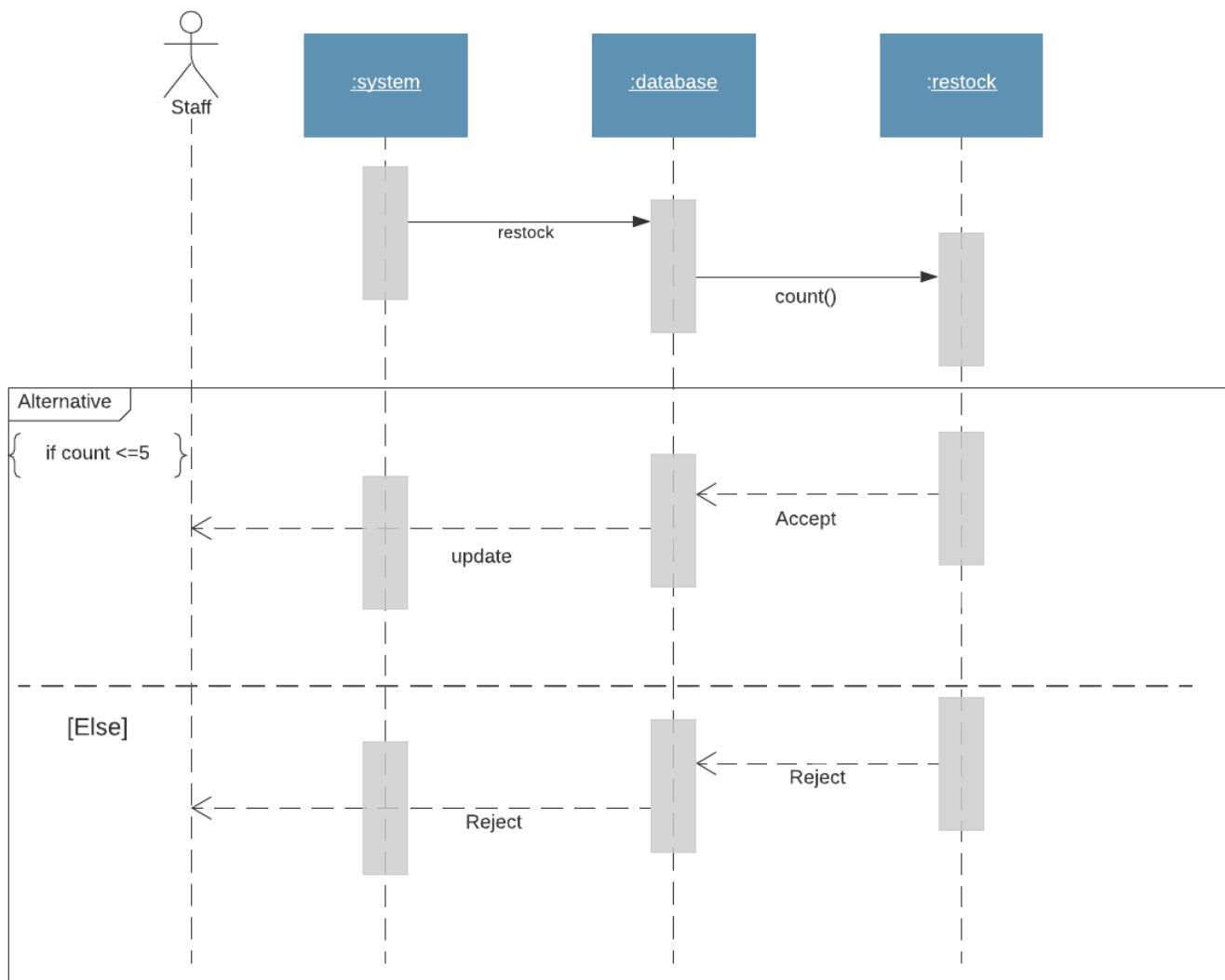


Figure 3.2.3.6
Restock staff sequence diagram for Haseef application.

7- Products admin:

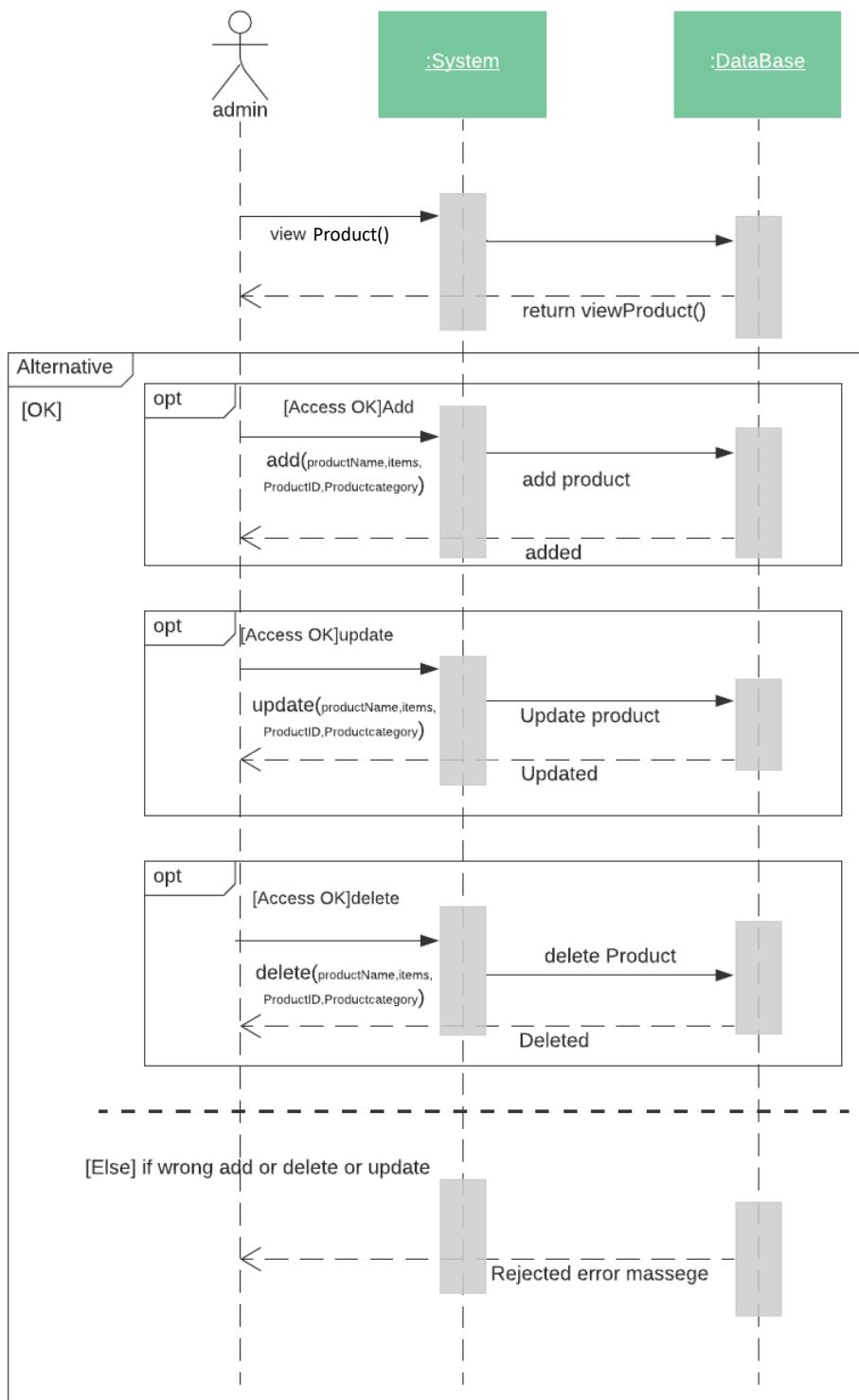


Figure 3.2.3.7
 Products admin sequence diagram for Haseef application.

8- Staff admin:

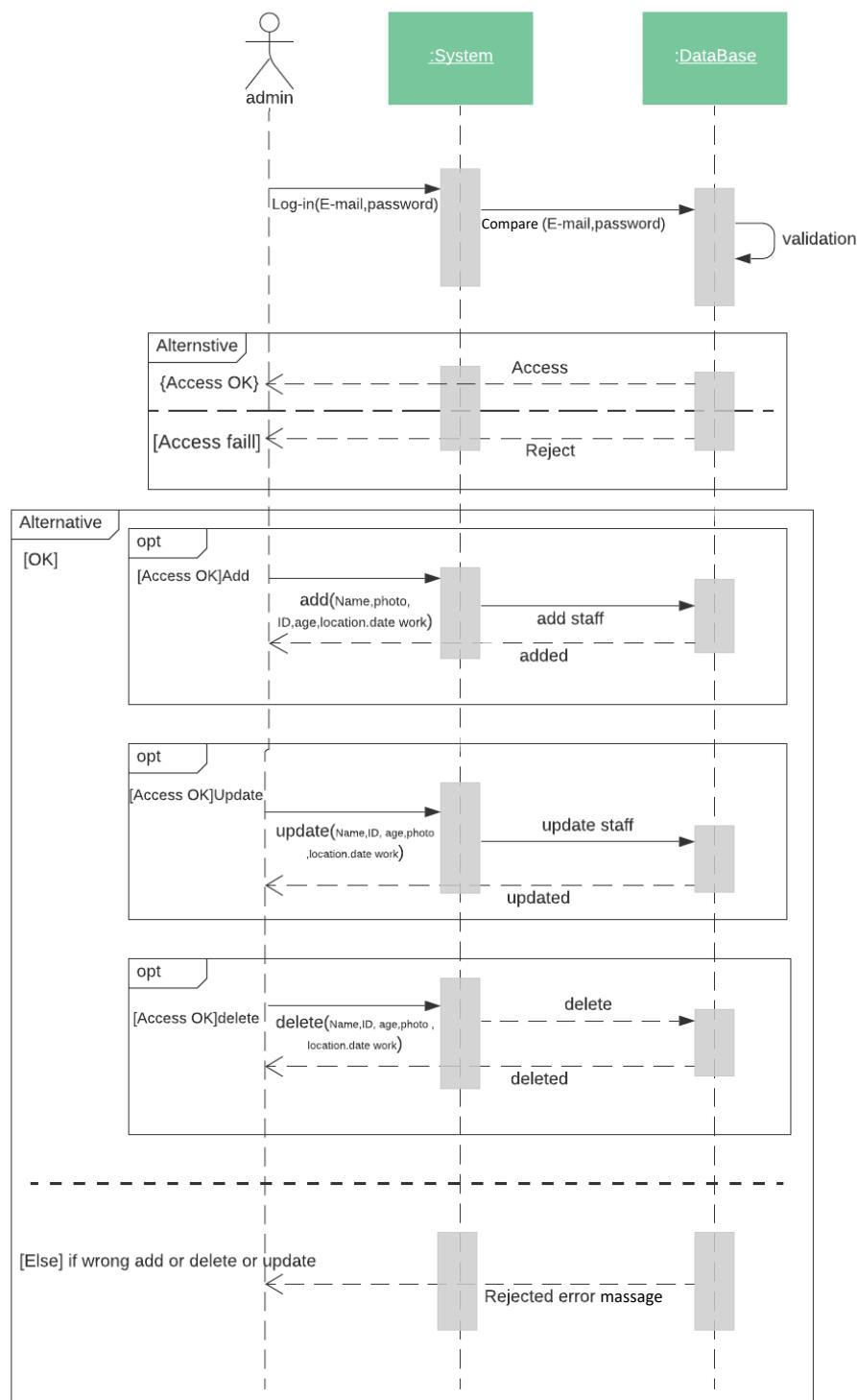


Figure 3.2.3.8
Staff admin sequence diagram for Haseef application.

3.2.4 Data flow diagram

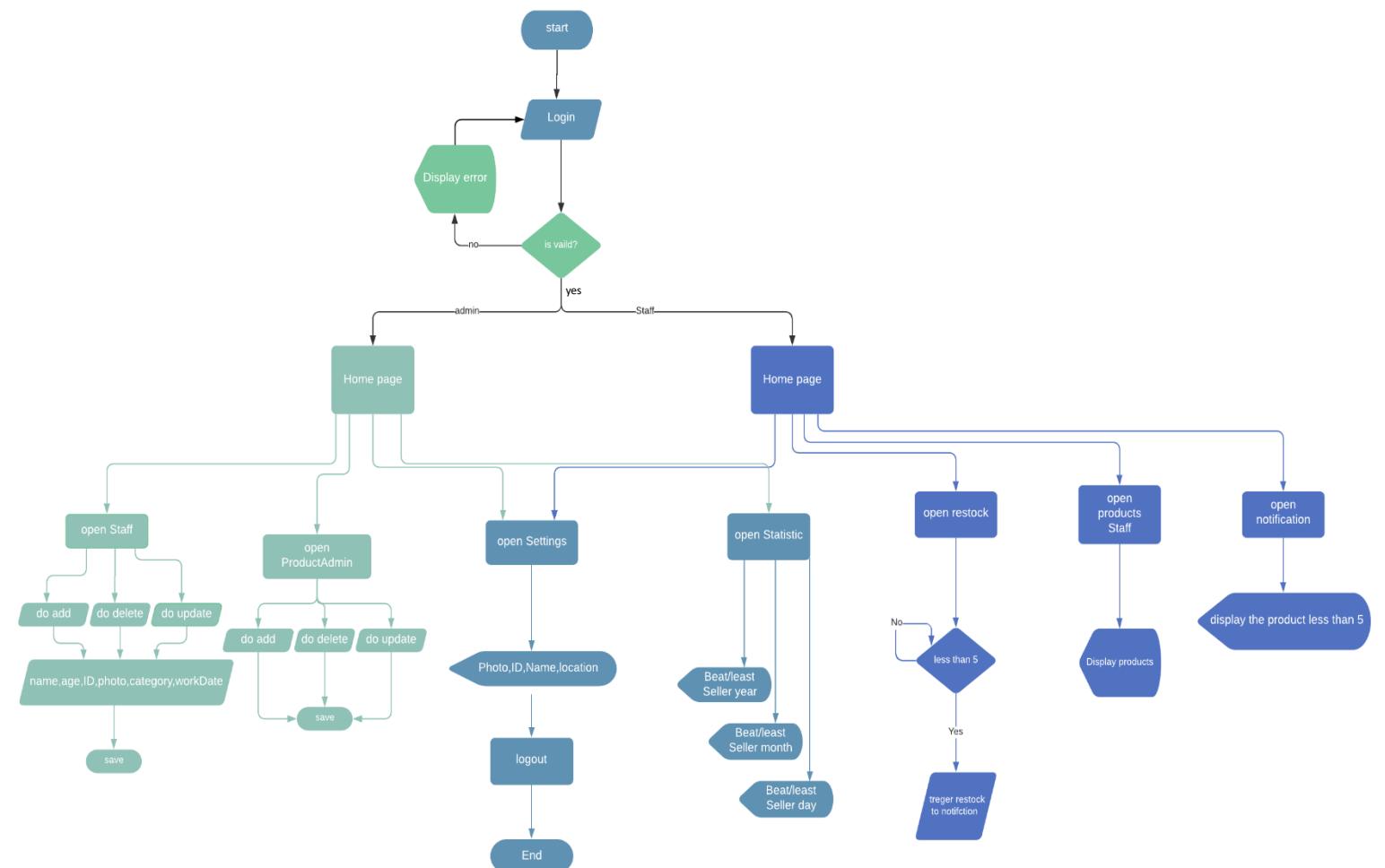


Figure 3.2.4
Data flow diagram for Haseef System.

CHAPTER 4

SYSTEM DESIGN

4. System Design

4.1 System Architecture

In our project, we choose the layered architecture pattern. The layered pattern is one of the most well-known software architecture patterns. The idea is to split up your code into “layers”, where each layer has a certain responsibility and provides a service to a higher layer. The following is about advantages and disadvantages of the layered architecture pattern.

Advantages:

1. Most developers are familiar with this pattern.
2. It provides an easy way of writing a well-organized and testable application.

Disadvantages:

1. It must be deployed as a singular unit thus, a change to a particular layer means the whole system must be redeployed.
2. The larger it is, the more resources it requires for requests to go through multiple layers and this will cause performance issue [11].

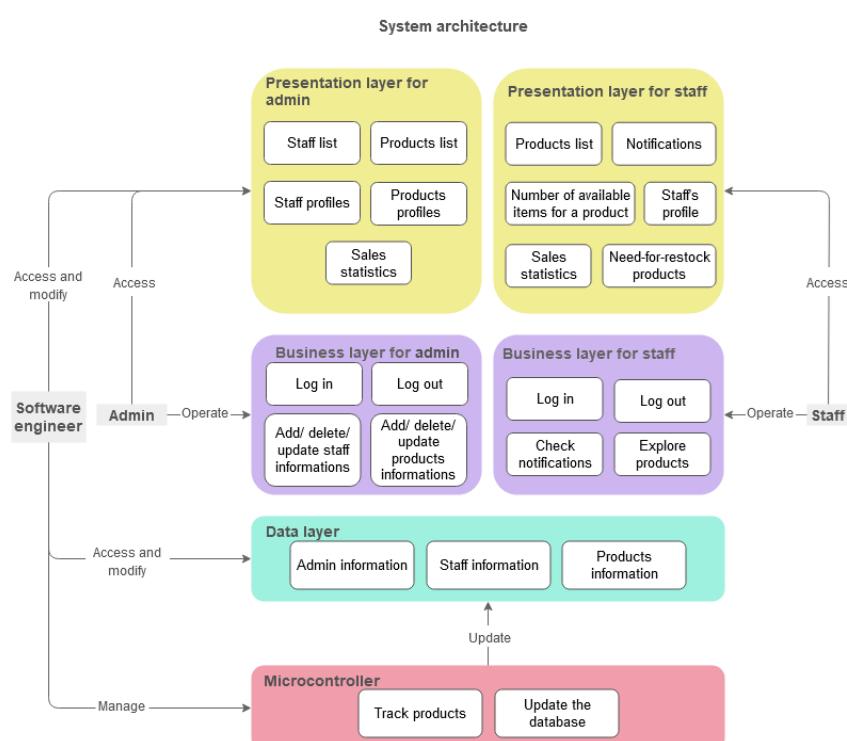


Figure 4.1
System architecture.

Presentation layer: Presentation layer components implement the functionality required to allow users to interact with the application.

Business layer: Is where you put the models and logic that is specific to the business problem you are trying to solve.

Data layer: This is where all the data are stored.

Microcontroller: Track the number of available items in the shelf through sensors and automatically update the specified attribute for that in the database.

4.2 User Interface Design:



Figure 4.2.1
The first interface

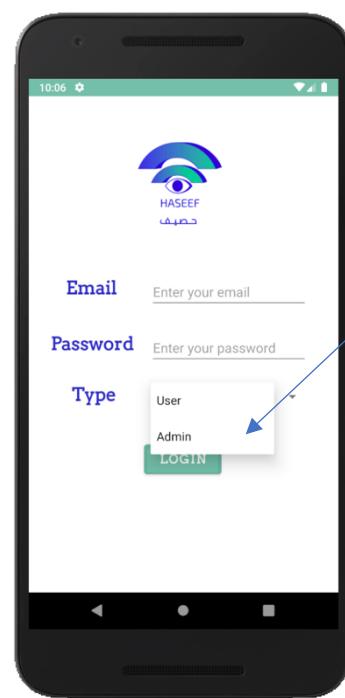


Figure 4.2.2
Log-in interface

The first interface: that will appear to the employees when they entering the app contains the logo and the name of the app. After 3 seconds, it will navigate the user to the log-in interface shown in Figure 4.2.2.

Log-in screen: each employee has an Email and a password assigned by the Admin, there is no registration for a new employee as the registration is done by the Admin.

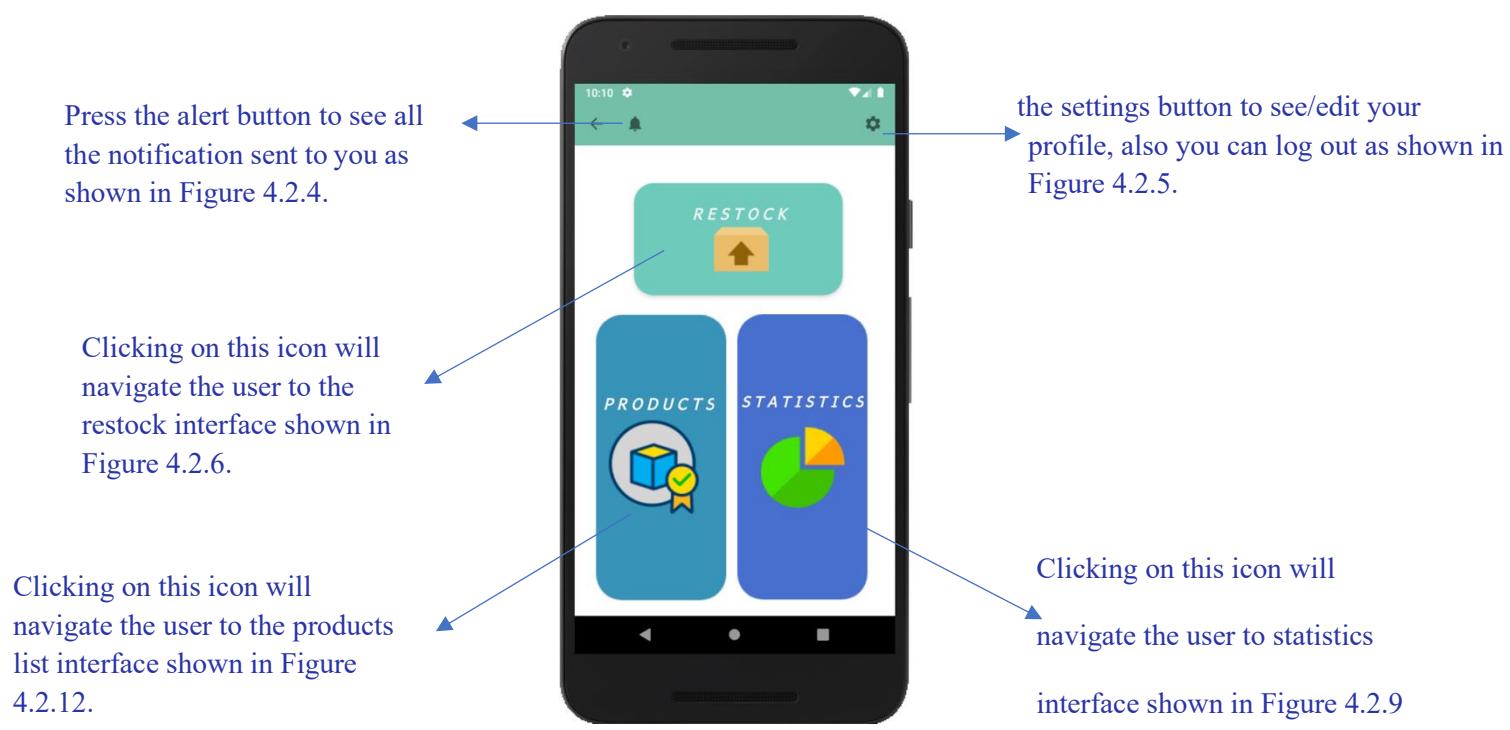


Figure 4.2.3
Home page interface for staff

Home page: The home page includes Restock, Products, Statistic, setting and notifications buttons if you press any one of them the interface will change to what you have pressed.

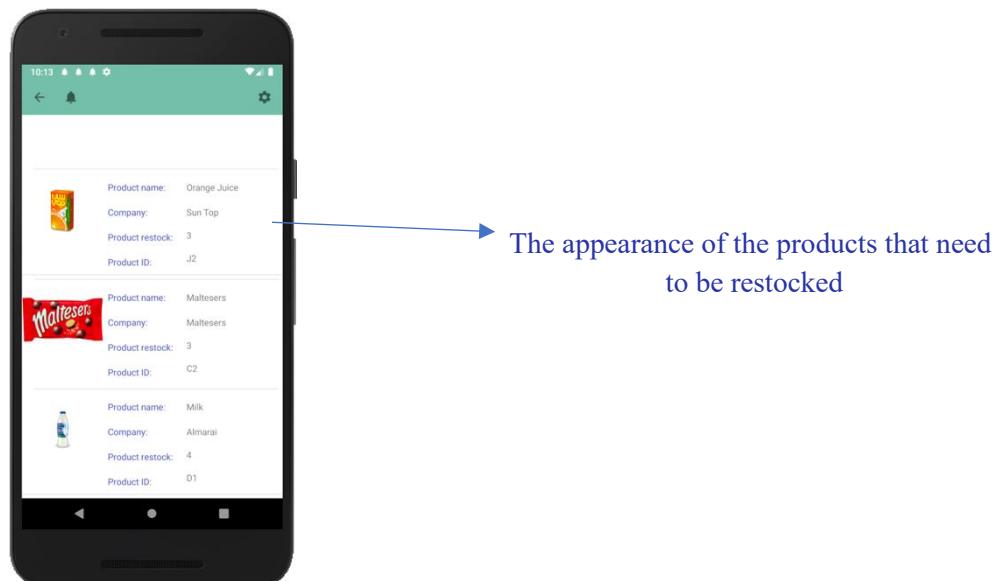


Figure 4.2.4
Notifications interface

Notifications: here we show all the alerts sent to the employee, an alert for an empty shelf.

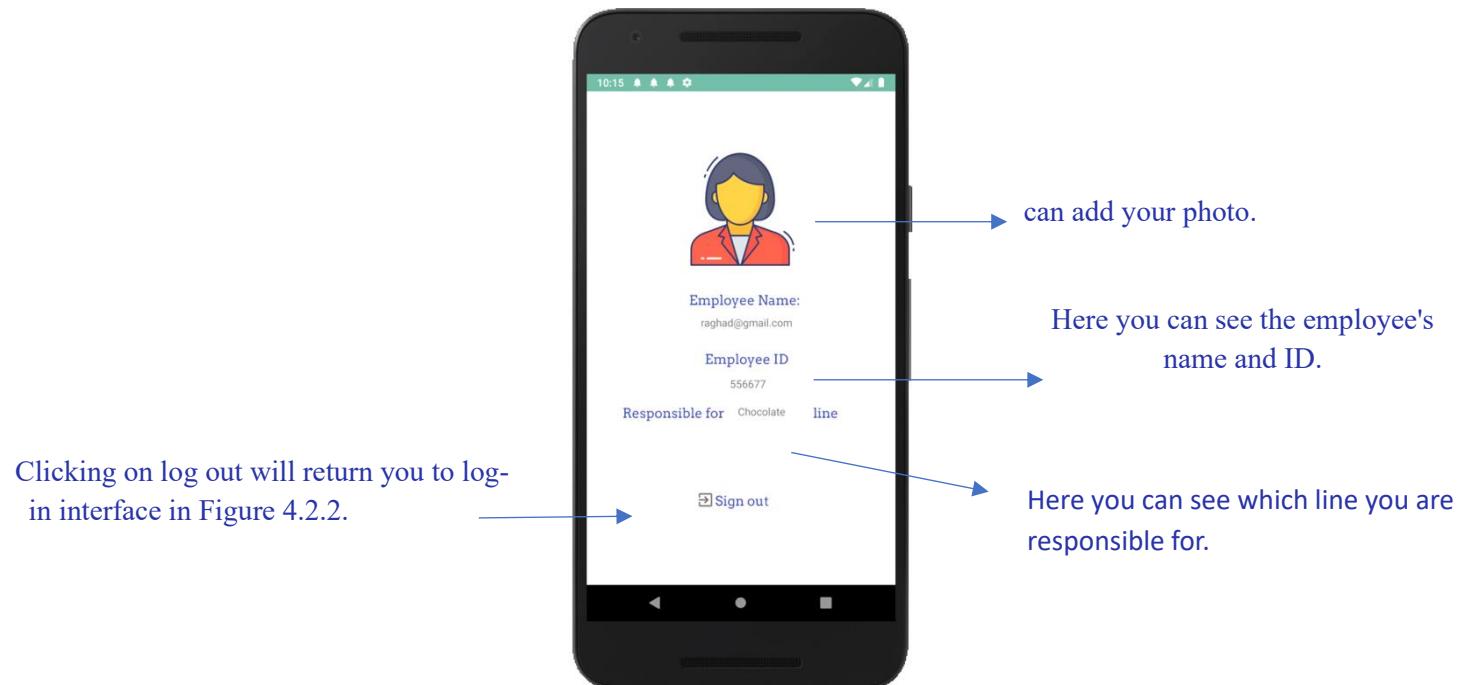


Figure 4.2.5
Profile interface

Settings: you can see the employee's name, ID and the line he was assigned to. Also, there are log out and help buttons. Help page well show you some important information to help you. log out button is to log out from the application.

First choose the type and then
the products that need
restocking will appear
in this page.

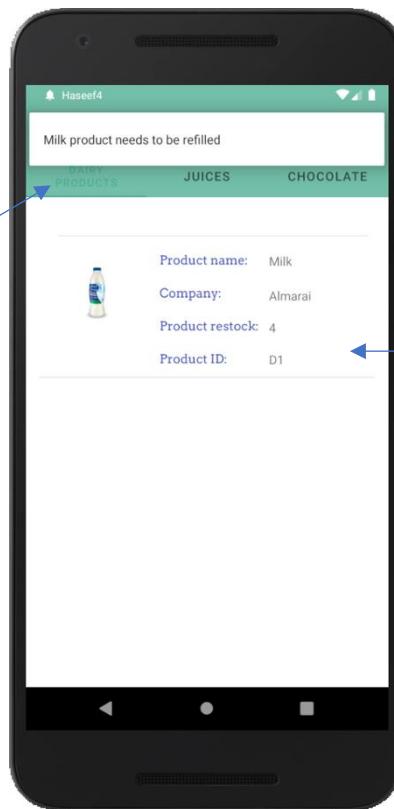


Figure 4.2.6
Dairy products line restock interface

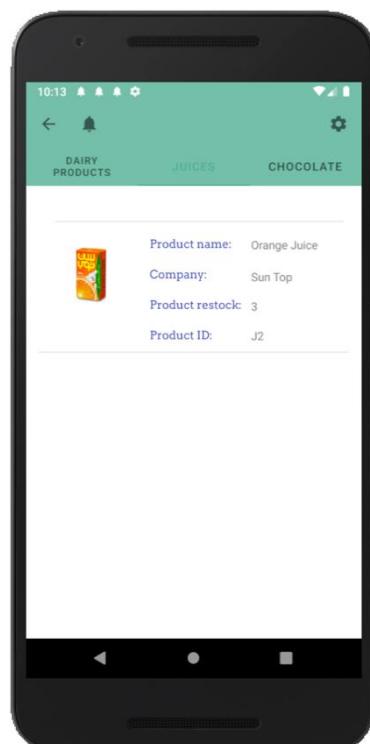


Figure 4.2.7
Juices line restock interface

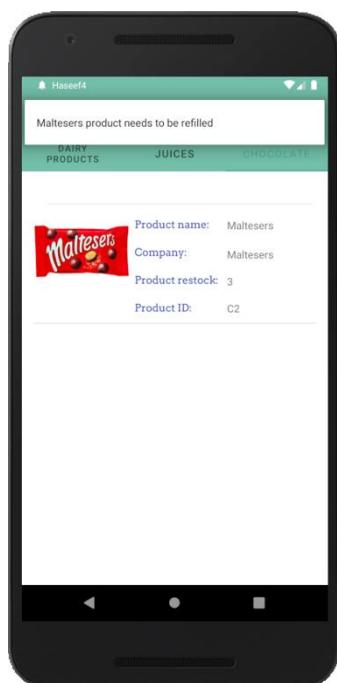


Figure 4.2.8
Chocolate line restock interface

Restock: shows products that need to restock and illustrates the line and shelf number.



Figure 4.2.9
Statistics

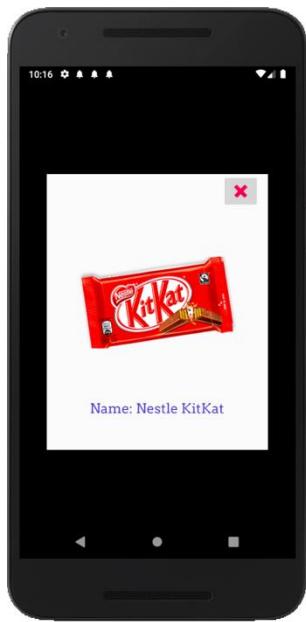
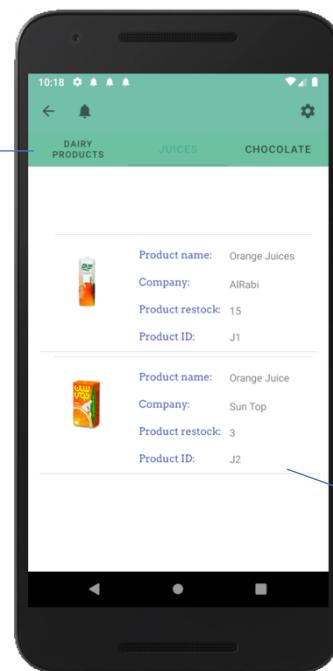


Figure 4.2.10
The best-selling product of the day



Figure 4.2.11
The least-selling product of the day

Statistics: if you press on the Statistic, you can see the best and least seller during days, months, and years



Choose a type of products to explore. ←

→ Here you can see a list of all products by their lines.

Figure 4.2.12
Products interface for staff

Products: the product page well show you all the products and their information. Also, there is a list in the left contains all the lines, you can click on any line to show the products contained in this line

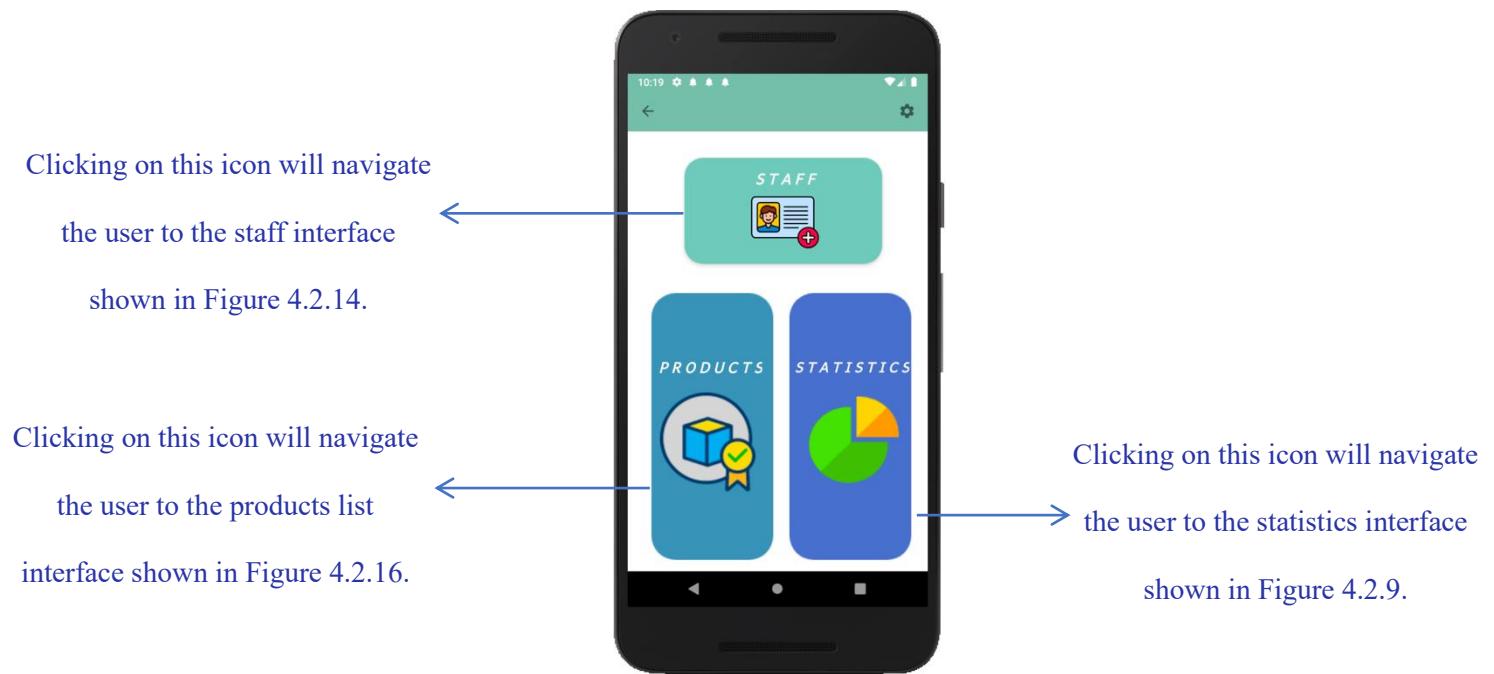


Figure 4.2.13
Home page interface for admin

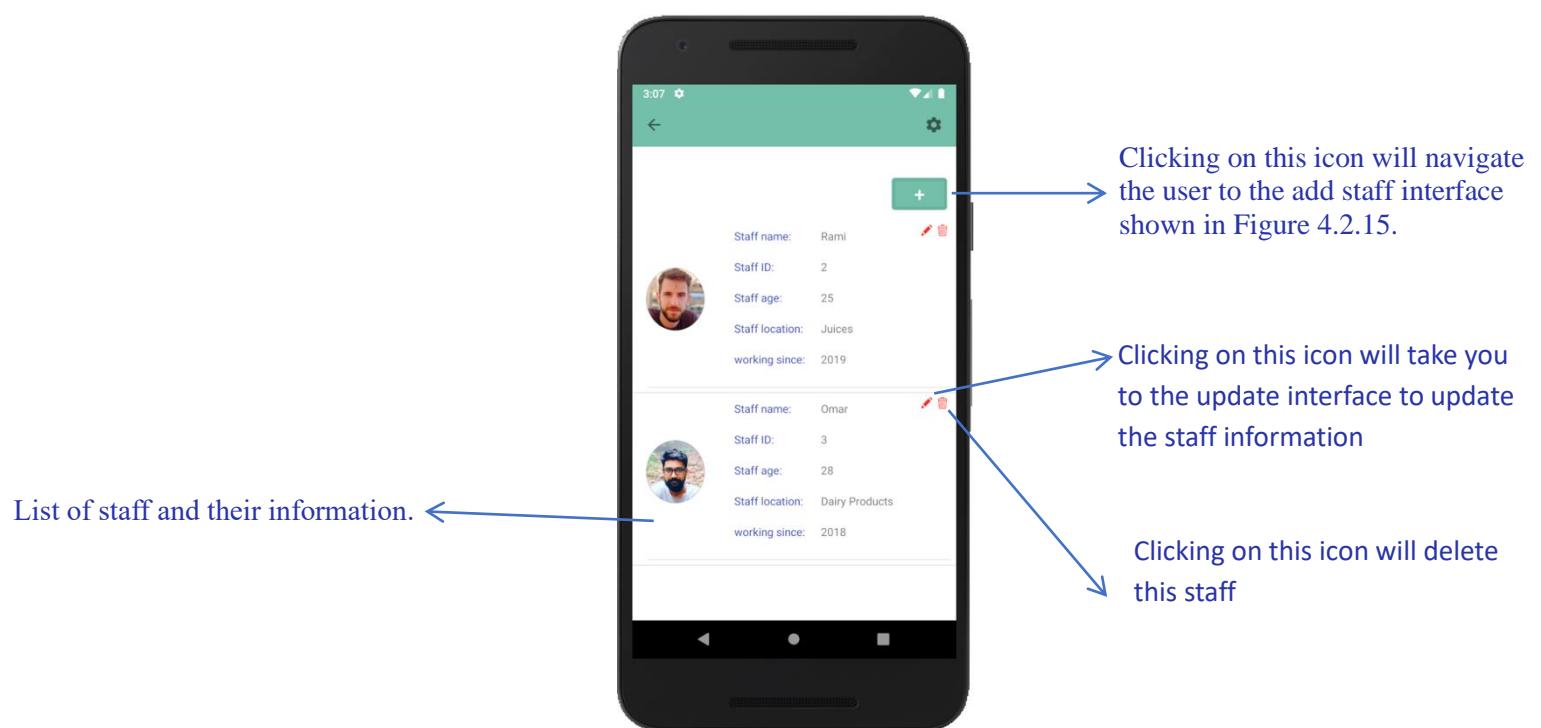


Figure 4.2.14
Staff list interface

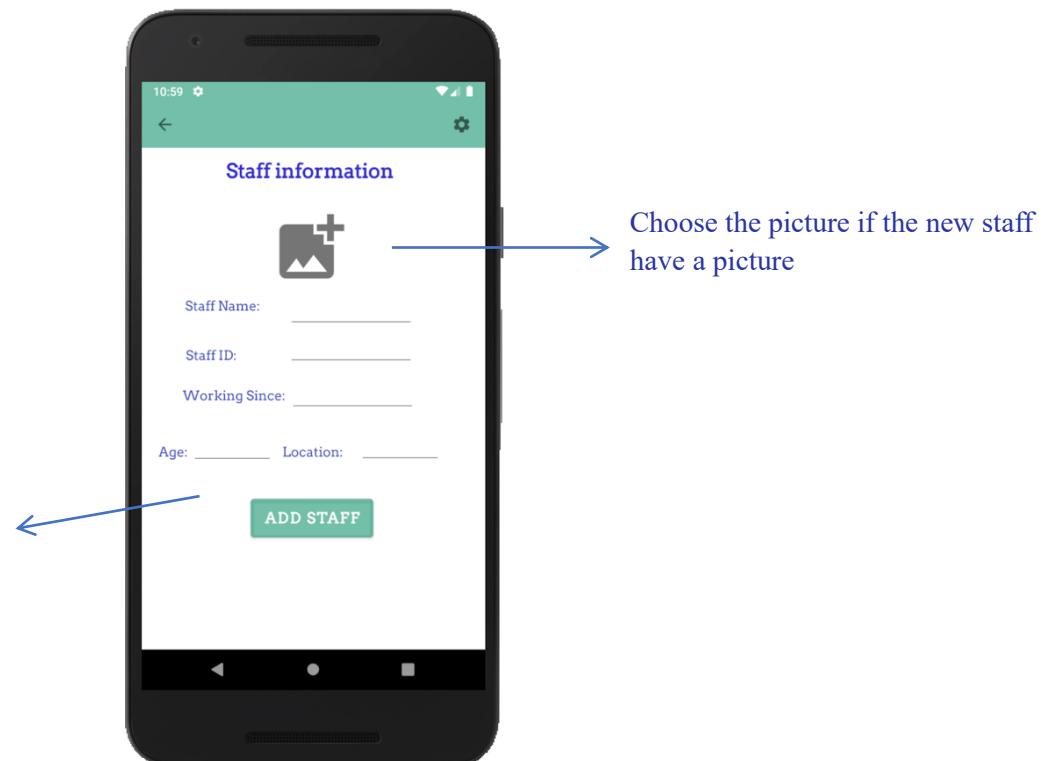


Fig 4.2.15
Adding new staff member interface

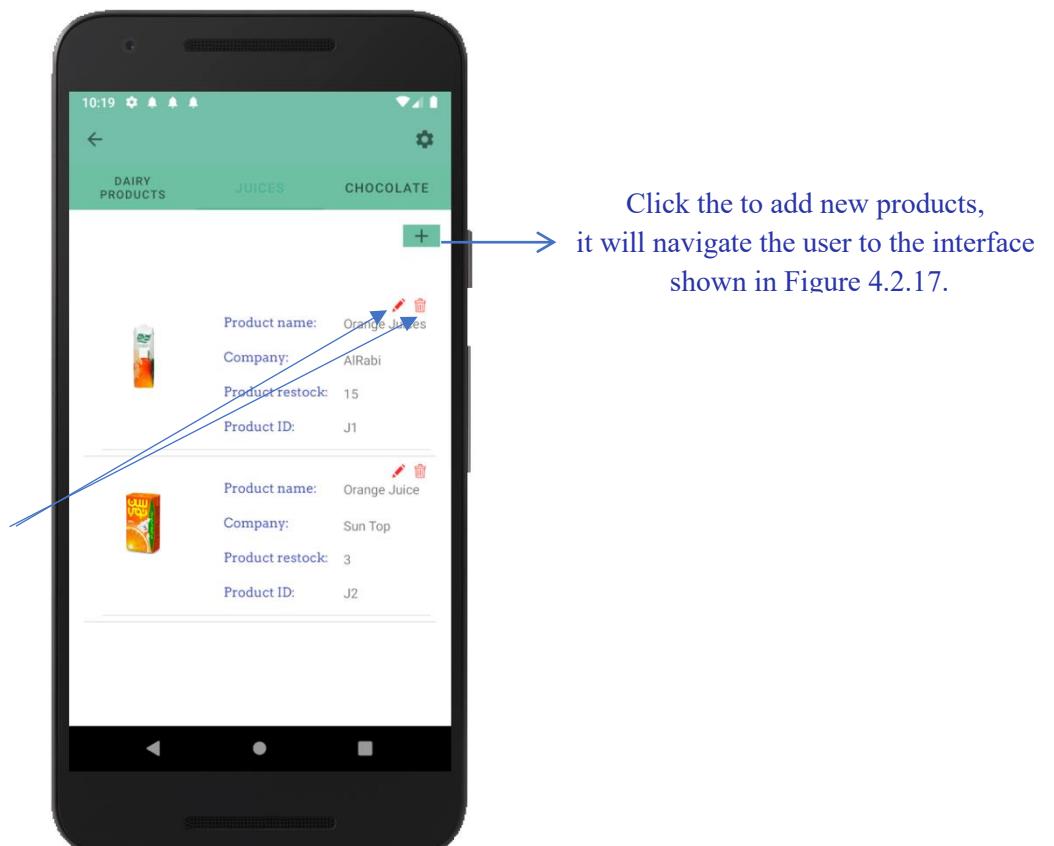


Figure 4.2.16
Products interface for admin

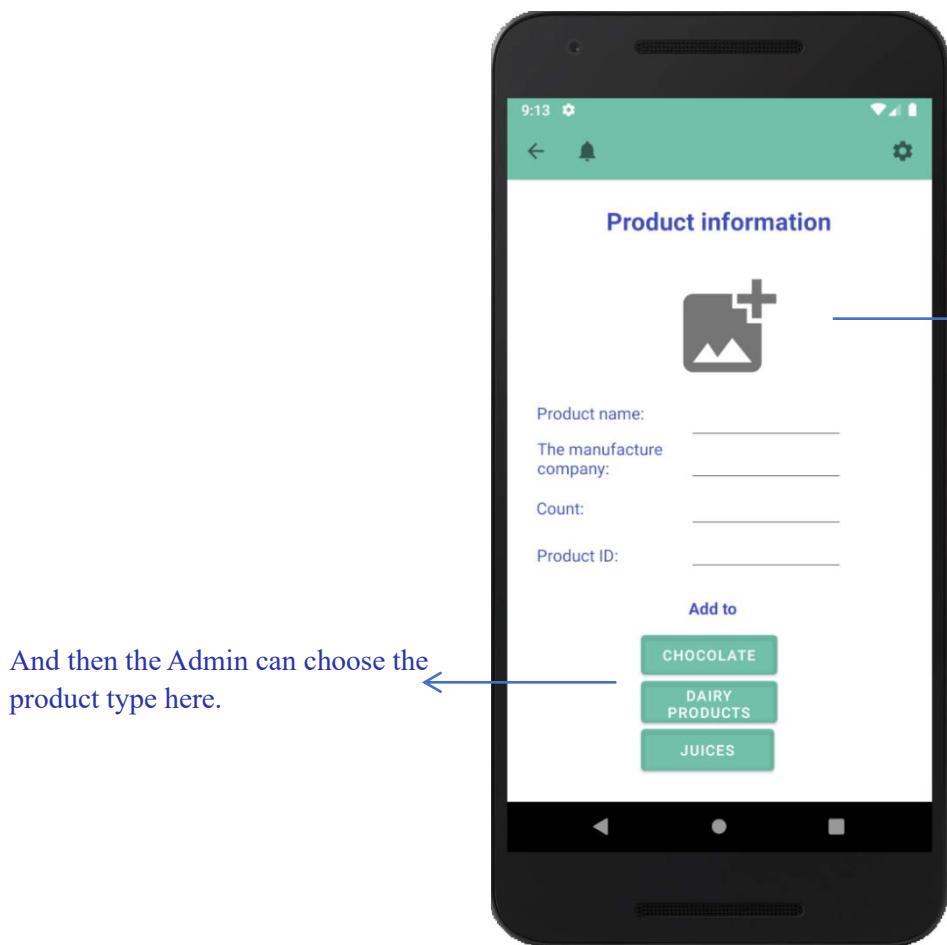


Figure 4.2.17
Adding new product interface

CHAPTER 5

IMPLEMENTATION

5.1 Introduction

This chapter is going to cover the implementation requirement and the implementation details that help us achieve this project, we are also going to show the Input/output screens to help demonstrate how the application is going to function.

5.2 Implementation Requirements

5.2.1 Hardware Requirements

Our implementation for this project did require us to use special hardware are:

1-Load cell: is a force transducer. It converts a force such as tension, compression, pressure, or torque into an electrical signal that can be measured and standardized. As the force applied to the load cell increases, the electrical signal changes proportionally [1].

2-Hx711: is an electronic scale module, whose working principle is to convert the measured changes in resistance value changes, through the conversion circuit into electrical output [2].

3- Wi-Fi (ESP8266):is a low-cost Wi-Fi chip, with full TCP / IP stack and microcontroller capacity [3].

4- PC with a high-quality processor to develop the system.

5.2.2 Software requirements

Software requirements to implements explained in detail in the following lines:

1- Arduino IDE: for uploading the program's functions on the Arduino and handling its inputs.

2- Android studio: is an integrated development environment (IDE) for Google's Android operating system, we will use it to design and develop the mobile application [4]. Specifically, Java programming languages for write the code, and the code testing, we will use Android Emulators.

3- Firebase: is a platform developed by Google for creating mobile applications that help us you quickly develop high-quality apps [5].

5.3 Implementation details

Implementation We are working in Haseef to improve products managing in commercial stores, by making it an easy and enjoyable experience.

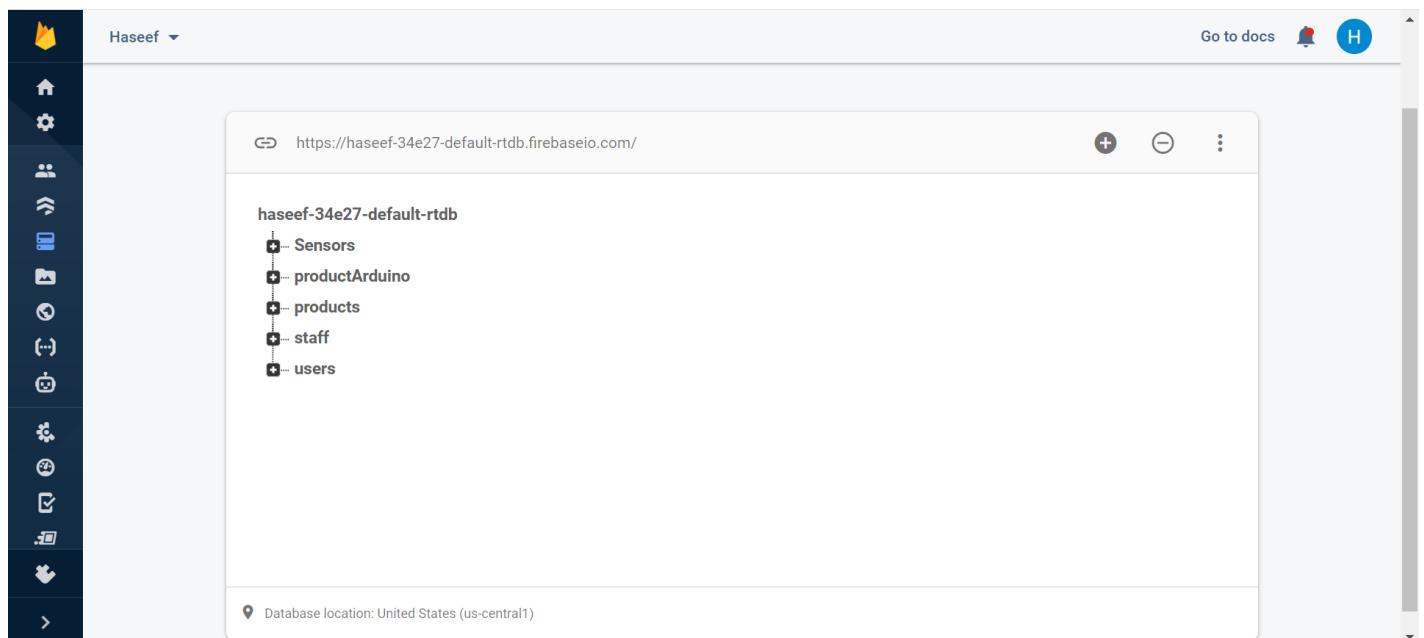
The implementation process starts with:

- 1- Install android studio with required tools
- 2- Install java libraries
- 3- Install Arduino IDE
- 4-Connecting the firebase with the app
- 5- Connect the firebase with the microcontroller
- 6- Coding with Java, C++
- 7- Designing the interfaces of the application

The sample code and database of our project is the following:

- Firebase Realtime database:

In Realtime database, we are going to store the admin and staff account, available and restock products, staff information and weight sensor. (see Figure 5.2.1).



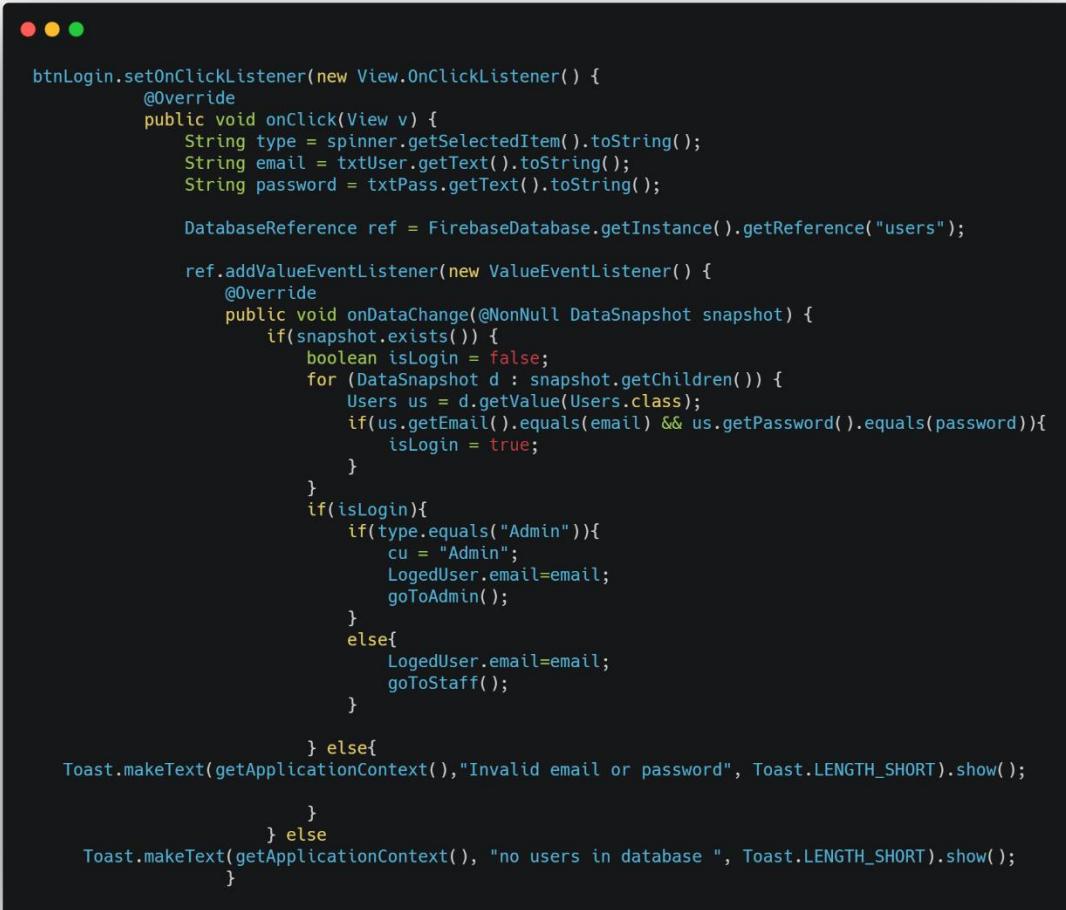
The screenshot shows the Firebase Realtime Database console. On the left is a sidebar with various icons for different services. The main area displays the database structure under the project name "Haseef". The root level contains the following nodes: Sensors, productArduino, products, staff, and users. Each node has a plus sign next to it, indicating they can be expanded. At the bottom of the main area, it says "Database location: United States (us-central1)". The top right corner has links for "Go to docs", a bell icon for notifications, and a user profile icon.

Figure 5.3.1 Realtime database

- **Application:**

1- Login

Admin or staff must fill the email and password to sign in the account, the information will be validated the database (see Figure 5.2.2).



```

btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String type = spinner.getSelectedItem().toString();
        String email = txtUser.getText().toString();
        String password = txtPass.getText().toString();

        DatabaseReference ref = FirebaseDatabase.getInstance().getReference("users");

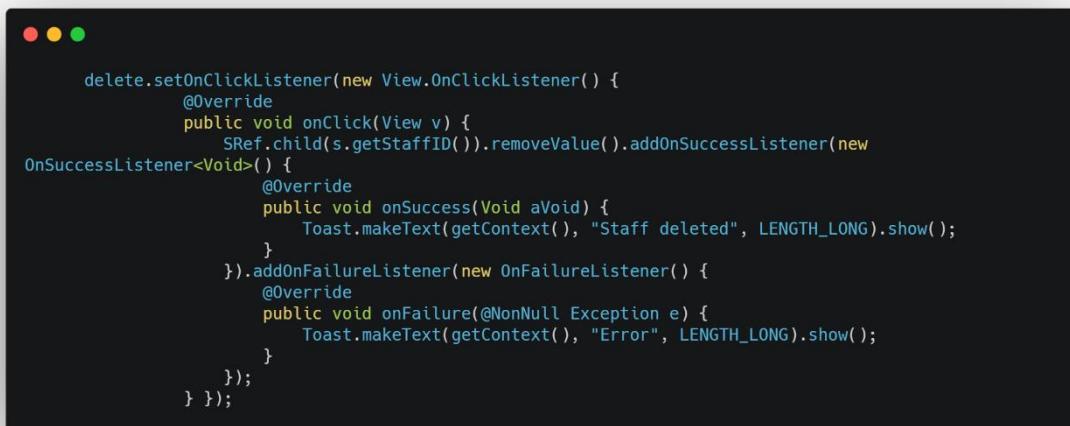
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.exists()) {
                    boolean isLoggedIn = false;
                    for (DataSnapshot d : snapshot.getChildren()) {
                        Users us = d.getValue(Users.class);
                        if(us.getEmail().equals(email) && us.getPassword().equals(password)){
                            isLoggedIn = true;
                        }
                    }
                    if(isLoggedIn){
                        if(type.equals("Admin")){
                            cu = "Admin";
                            LogedUser.email=email;
                            goToAdmin();
                        }
                        else{
                            LogedUser.email=email;
                            goToStaff();
                        }
                    } else{
                        Toast.makeText(getApplicationContext(),"Invalid email or password", Toast.LENGTH_SHORT).show();
                    }
                } else{
                    Toast.makeText(getApplicationContext(), "no users in database ", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}

```

Figure 5.3.2 Login code

2- Delete Staff

when admin clicks delete staff icon, it will be deleted from the database. (see **Figure 5.2.3**).



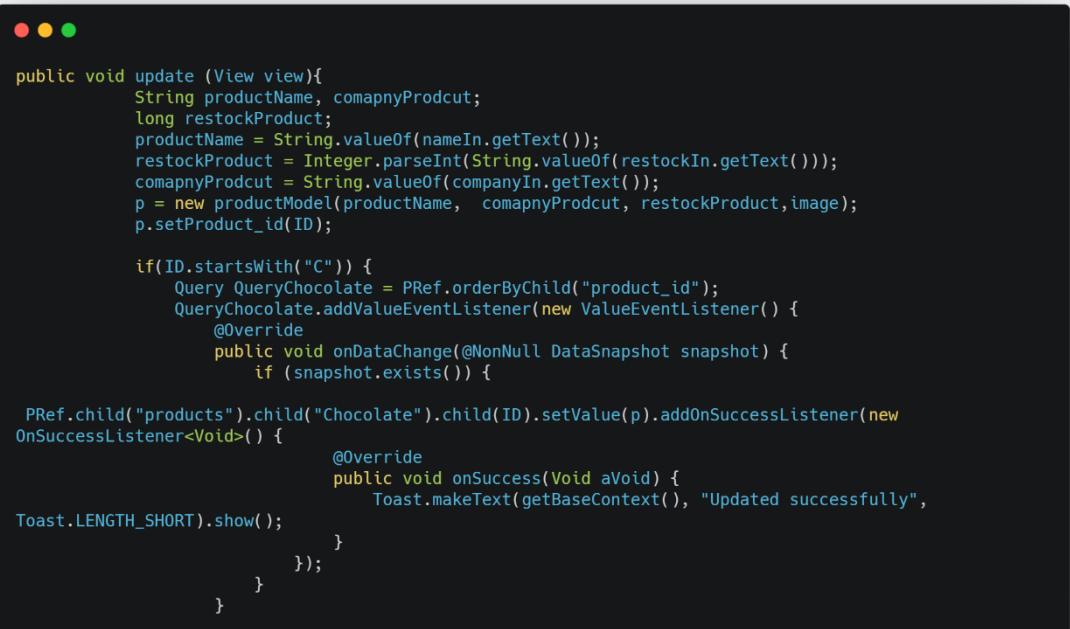
```

delete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        SRef.child(s.getStaffID()).removeValue().addOnSuccessListener(new
OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            Toast.makeText(getApplicationContext(), "Staff deleted", LENGTH_LONG).show();
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Toast.makeText(getApplicationContext(), "Error", LENGTH_LONG).show();
        }
    });
}
});
```

Figure 5.3.3 Delete Staff code

3- Update product

when admin clicks update product icon, it will be updated in the database. (see **Figure 5.2.4**).



```

public void update (View view){
    String productName, comapnyProdcut;
    long restockProduct;
    productName = String.valueOf(nameIn.getText());
    restockProduct = Integer.parseInt(String.valueOf(restockIn.getText()));
    comapnyProdcut = String.valueOf(companyIn.getText());
    p = new productModel(productName, comapnyProdcut, restockProduct,image);
    p.setProduct_id(ID);

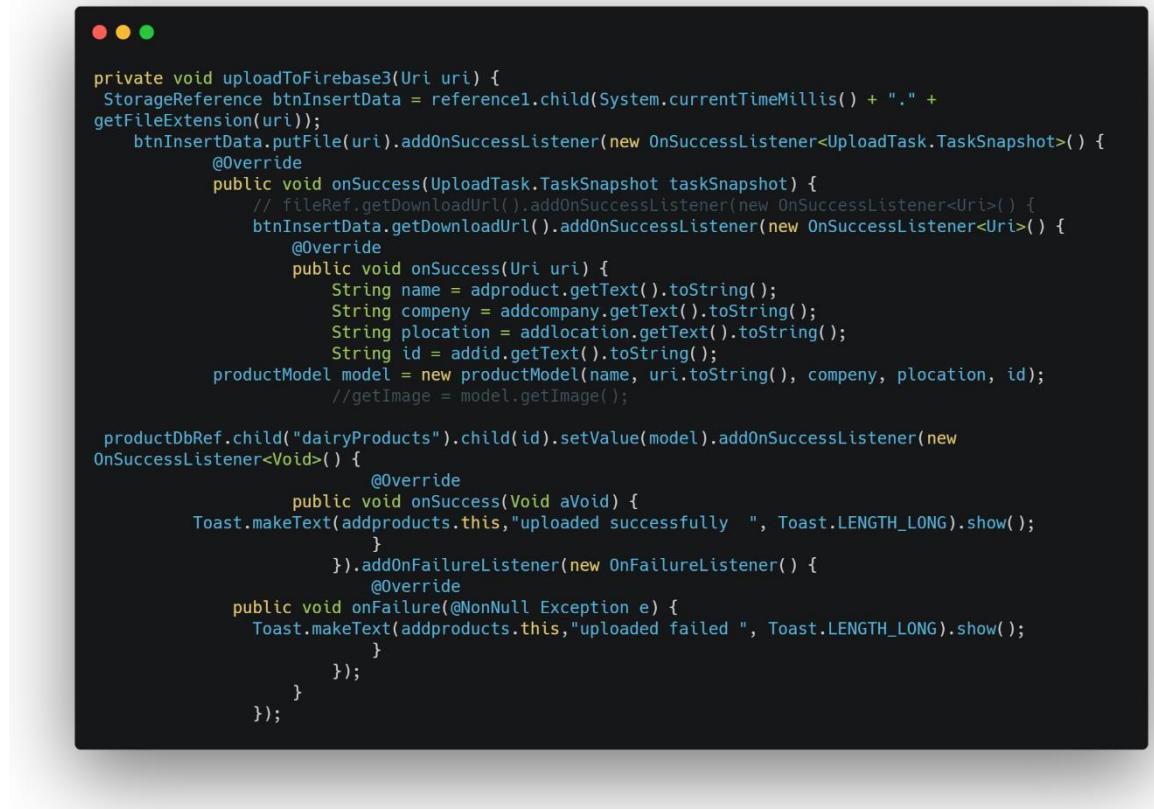
    if(ID.startsWith("C")) {
        Query QueryChocolate = PRef.orderByChild("product_id");
        QueryChocolate.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {

PRef.child("products").child("Chocolate").child(ID).setValue(p).addOnSuccessListener(new
OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {
                Toast.makeText(getApplicationContext(), "Updated successfully",
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Figure 5.3.4 update product code

4- Add new product

When the admin goes to the add products page, the product category must be specified. The figure below shows the insert to dairy products category code. (see **Figure 5.2.5**).



```

private void uploadToFirebase3(Uri uri) {
    StorageReference btnInsertData = reference1.child(System.currentTimeMillis() + "." +
        getFileExtension(uri));
    btnInsertData.putFile(uri).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            // fileRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
            btnInsertData.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                @Override
                public void onSuccess(Uri uri) {
                    String name = adproduct.getText().toString();
                    String compeny = addcompany.getText().toString();
                    String plocation = addlocation.getText().toString();
                    String id = addid.getText().toString();
                    productModel model = new productModel(name, uri.toString(), compeny, plocation, id);
                    //getImage = model.getImage();

                    productDbRef.child("dairyProducts").child(id).setValue(model).addOnSuccessListener(new
                    OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void aVoid) {
                            Toast.makeText(addproducts.this,"uploaded successfully ", Toast.LENGTH_LONG).show();
                        }
                    }).addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {
                            Toast.makeText(addproducts.this,"uploaded failed ", Toast.LENGTH_LONG).show();
                        }
                    });
                }
            });
        }
    });
}

```

Figure 5.3.5 Add new product code

- **Microcontroller:**

1- Connect to Wi-Fi router

```
● ● ●

float weight1 = 0.0;
void setup() {
scale1.begin(D1, D2);
// setup code, to run once:
Serial.begin(115200);
delay(10);

Serial.print("Connecting");
WiFi.disconnect();
Serial.println("disconnected");
WiFi.begin(ssid, pass); // Connect to WiFi router

// Wait for connection

while (WiFi.status() != WL_CONNECTED) {
Serial.print(".");
ESP.wdtFeed(); //will ensure the reset of the watchdog.
delay(10);

}
```

Figure 5.3.6 connect to Wi-Fi router code

2-Connect to Realtime database

```

}
//If successfully connected to the wifi router, the name of router will be displayed in the serial
monitor.

Serial.print("Successfully connected to : ");
Serial.println("WIFI_SSID");
Serial.printIn(WiFi.SSID());

//Firebase Realtime Database Configuration.

scale1.set_scale(calibration_factor);
Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
scale1.tare() ;
}

```

Figure 5.3.7 connect database code

3- Calculate the average weight and update the database

```

void loop() {
    weight1 = scale1.get_units();
    Serial.println(weight1);
    float avgweight1 = 0;
    if (weight1 > threshold) { // Takes measures if the weight is greater than the threshold //0
        float weight01 = scale1.get_units();
        for (int i = 0; i < num_measurements; i++) { // Takes several measurements
            delay(100);
            weight1 = scale1.get_units();
            avgweight1 = avgweight1 + weight1;
            if (abs(weight1 - weight01) > threshold1) {
                avgweight1 = 0;
                i = 0;
            }
            weight01 = weight1;
        }
        avgweight1 = avgweight1 / num_measurements; // Calculate average weight
    }
    Serial.print("Measured weight for Sensor 1: ");
    Serial.printf("%.2f",avgweight1, 1);
    Serial.println(" g");

    char result[2]; // Buffer big enough for 7-character float
    dtostrf(avgweight1, 6, 1, result); //Decimal to String Float

    Firebase.setFloat("productArduino/dairyProducts/D1/weight2",avgweight1);
}

```

Figure 5.3.8 average weight code



5.4 I/O Screens

Figure 5.4.1 in Hassef application, first thing you will see is ‘welcome page’ it also contains log-in option so you can enter to the homepage.

Figure 5.4.2 log-in Screen contains log-in interface need to insert E-mail, Password and choose (Admin- user) to log-in.



Figure 5.4.1 Welcome I/O screen

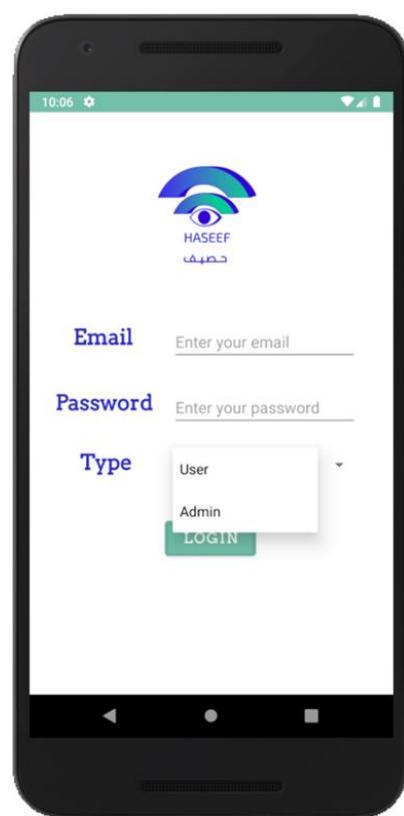


Figure 5.4.2 Log-in I/O screen

Figure 5.4.3 when they Log-in to Hassef as user then home page will open which we call it "staff home page " that include (restock- products- statistic) icons.

Figure 5.4.4 in the notification page, will show to the staff the product that need to re-fill.



Figure 5.4.3 Home page 'user' I/O screen

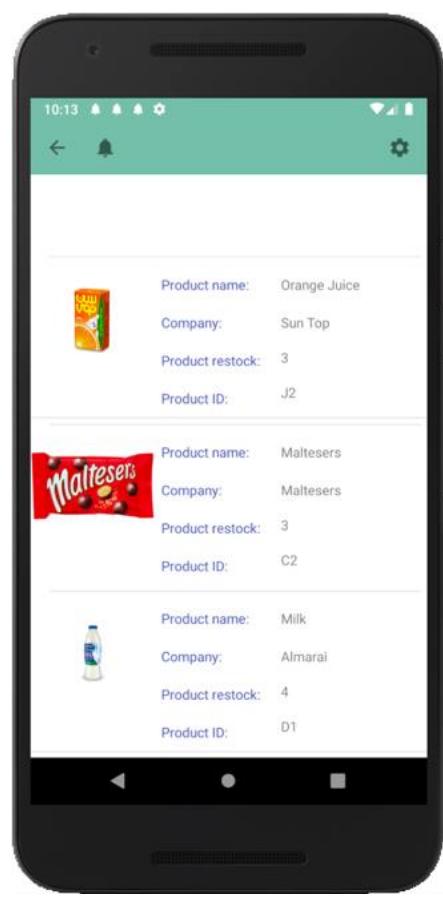


Figure 5.4.4 Notification I/O screen

Figure 5.4.5 settings page will show the information about the user either admin or user, they can also sign-out.

Figure 5.4.6 when choose restock, will show all the product that less than 5 .

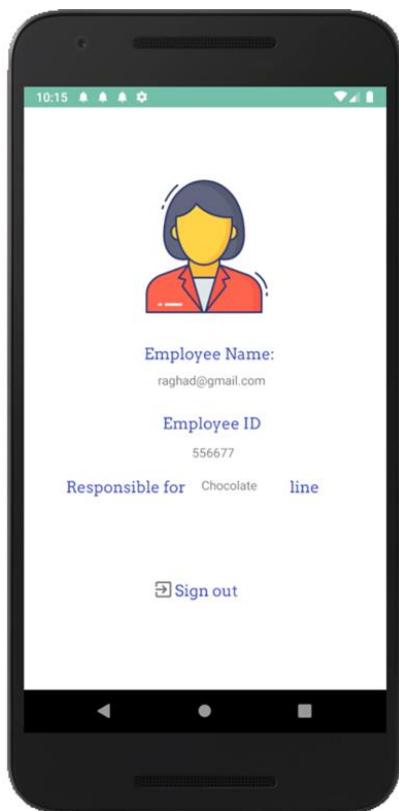


Figure 5.4.5 Settings I/O screen

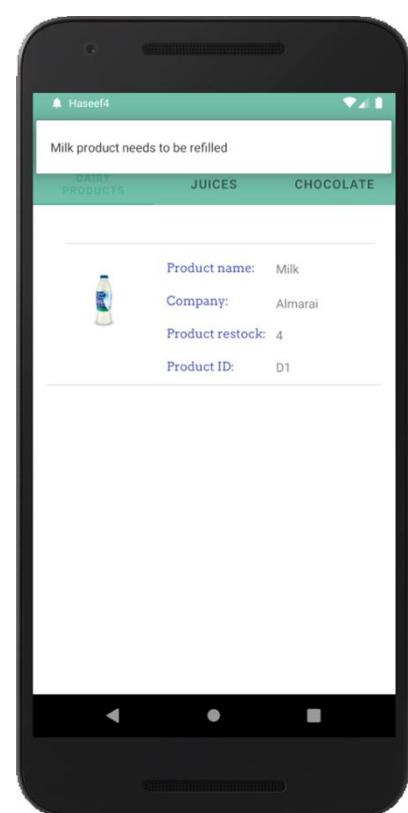


Figure 5.4.6 Restock I/O screen

Figure 5.4.7 when choose Statistics, the page going to show the "best – least" selling for “day-month-year” and witch product is the best-least.



Figure 5.4.7 Statistics I/O screen

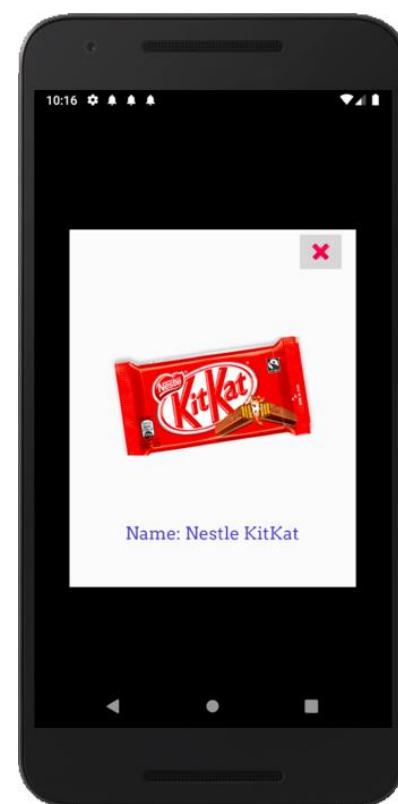


Figure 5.4.8 pop-up I/O screen

Figure 5.4.9 when choose product, the page going to show the category for the products and can choose one of category and will display all the products and can know the information about the product.

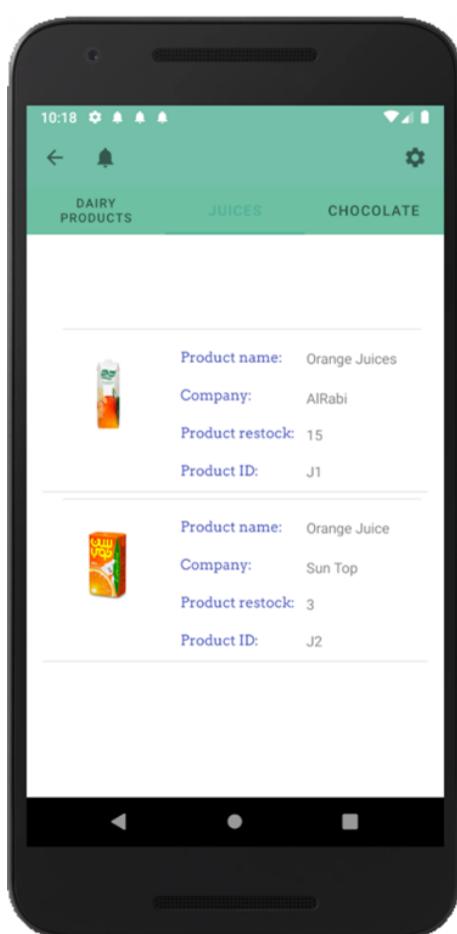


Figure 5.4.9 Product I/O screen

Figure 5.4.10 When they Log-in to Hassef as admin then home page will open which we call it ‘admin home page’ that include (staff- products- statistic) icons.

Figure 5.4.11 when choose staff, all the staff will show in the staff page also can add a new staff, update information about the staff and delete the staff.



Figure 5.4.10 Home page ‘Admin’ I/O screen

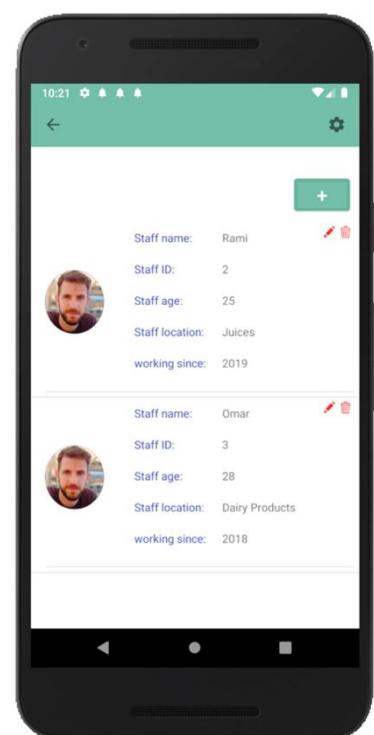


Figure 5.4.11 Staff I/O screen

Figure 5.4.12 when you choose add, you can add a staff information than press ‘add staff’ to added.

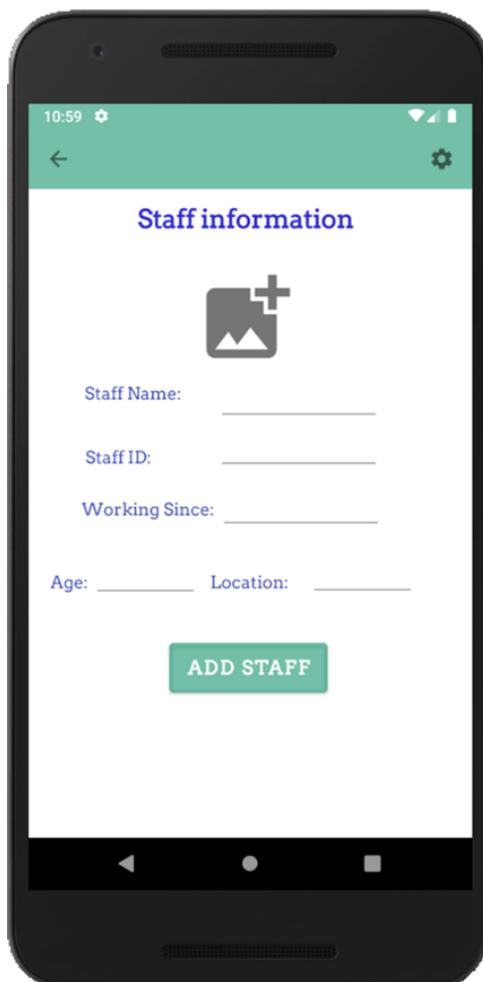


Figure 5.4.12 Add staff I/O screen

Figure 5.4.13 when choose products, all the products will show in the product page also can add a new product, update information about the product and delete the product

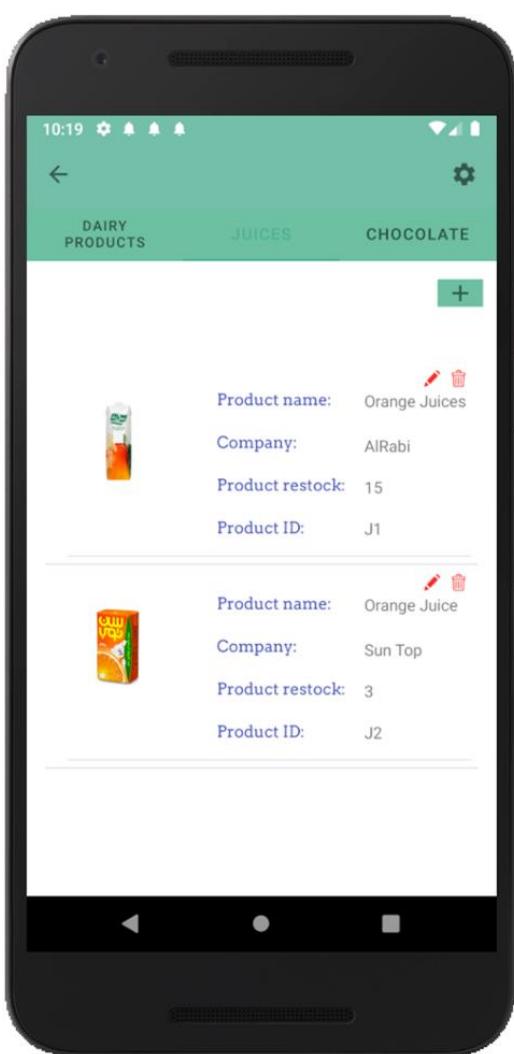


Figure 5.4.13 Product 'Admin' I/O screen

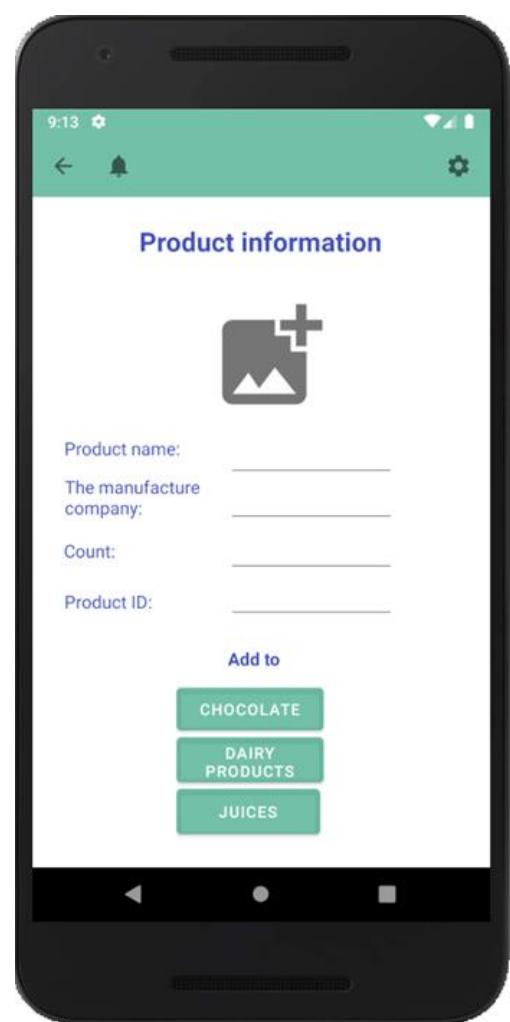


Figure 5.4.14 Add product I/O screen

CHAPTER 6

TESTING

6.1 Test plan

Testing strategy:

our strategy is to break down the testing process into levels. A test level is a group of test activities that are organized and managed together. [6]

Haseef testing process has 3 levels as shown in Fig 6.1.1.



Fig 6.1.1 Haseef testing process

Unit testing: is the process of testing individual application components (methods, classes) to ensure that the units are working as defined and are integrated in a correct way and working together as expected. [6] It helps developers to find bugs in the early stages of the development cycle and usually done at the time of adding a new feature.

System testing: testing an integrated system to verify that it meets specified requirements. System testing is concerned with the behavior of the whole system. Its objective is to ensure that the system is correctly behaving as defined in the requirements.

Acceptance testing: is the testing with respect to user needs, requirements, and business processes, which is conducted to determine whether a system satisfies the acceptance criteria (usability, functional completeness, availability) and to determine whether to accept the solution or not. [6]

Features to be tested:

For admin:

1. For login process, test the validation process and the process of navigating the user to the admin UI.
2. For the staff page, test the display of each staff's information.
3. Add staff to the database.
4. Delete staff from the database.
5. Update staff in the database.
6. For the products page, test the display of each product's information.
7. Add product in the database.
8. Delete product from the database.
9. Update product in the database.
10. For the settings page, test the access to the user profile and check if the displayed information is belonging to the current user.
11. Logout from the app in the settings page.

For staff:

1. For login process, test the validation process and the process of navigating the user to the staff UI.
2. For the restock page, test if all products with less than or equal to 5 items left are displayed.

3. For the products page, test the display of each product's information.
4. Test the receiving of notifications in case of out-of-stock events.
5. Test the display of the notification history in the notification page.
6. For the settings page, test the access to the user profile and check if the displayed information is belonging to the current user.
7. Logout from the app in the settings page.

In the microcontroller side:

1. Test the loadcell measurements.
2. Test the Wi-Fi module connection with the Firebase database.
3. Test storing of the measurements to the Firebase.

Schedules of test activities:

Table 6.1

Test activities

No.	Test activity	Time
1	Unit testing	In time of adding a new feature. [6]
2	System testing	After completing the whole project. [6]
3	Acceptance	Before the deployment time. [6]

Table 6.2

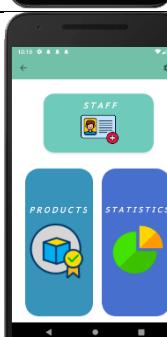
Testing tasks:

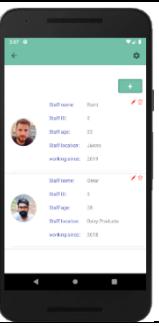
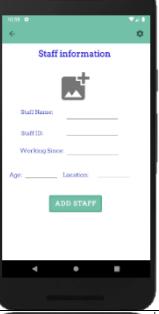
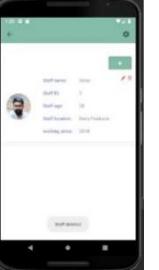
Testing tasks

Task	Location
Write a test plan	Test plan section
Define test items	Test plan section > test items
Define features to be tested	Test plan section >features to be tested
Define Unit Test	Test plan section > test strategy
Define System Test	Test plan section > test strategy
Define Acceptance Testing	Test plan section > test strategy
Build test cases	Test cases section
Document test results	Test results section

Table 6.3

Test cases for admin

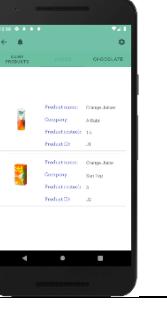
Test case	Check item	Test case's objective	Test Data /input	Expected result	Test result	Actual result
1	Logo	Appearance of logo	-	Appearance of logo	pass	
2	Login page	Log in to the system	Email, Password	Logged in	pass	
3	login with incorrect email and password	Admin with incorrect email and password cannot enter the system	Wrong email or password	Toast message, Invalid email, or password	pass	
4	Admin home	Admin should be able to choose one of the actions	click any icons in the page and it should work	Appearance of icons	pass	

5	Choose Staff page	Should show all the staff information to the admin	Click on staff icon	The appearance of all staff information	pass	
6	Update staff	Admin should be able to choose update icon.	Staff name Staff ID Staff age Staff location Working since image	The ability to fill in all fields	pass	
7	Successful staff information update	Staff information should be updated in the database	Staff name Staff ID Staff age Staff location Working since	Toast message, Staff updated	pass	
8	Add staff	Admin should be able to choose add icon	Staff name Staff ID Staff age Staff location Working since image	The ability to fill in all fields	pass	
9	Successful staff information Add	Staff information should be added in the database	Staff name Staff ID Staff age Staff location Working since image	Toast message, uploaded successfully	pass	
10	Successful staff information Delete	Admin should be able to choose delete icon	Click on delete icon	Toast message, Staff deleted	pass	

11	choose product page	Should show all the product information to the admin	Click on product icon	The appearance of all product information	pass	
12	Update product	Admin should be able to choose update icon	Product name Product ID Number of items Company	The ability to fill in all fields	pass	
13	Successful product information update	product information should be updated in the database	Product name Product ID Number of items Company	Toast message, updated successful	pass	
14	Add product	Admin should be able to choose add icon	Product name The manufacture company count Product ID image	The ability to fill in all fields	pass	
15	Successful product information Add	product information should be added in the database	Product name The manufacture company count Product ID image	Toast message, uploaded successfully	pass	

16	Successful product information Delete	Admin should be able to choose delete icon	Click on delete icon	Toast message, product deleted	pass	
17	choose statistic page	Should show all the best and least product in day, month, year and year to the admin	Click on the best (day, month, year) or least product	The appearance of all statistic information	pass	
18	Clicking on setting	admin can be able to show his/her information	Click on setting icon	Admin see his/her information	pass	
19	Clicking on sign out	The system goes to login page	Clicking on sign out icon	Go to login page	pass	

Table6.4 Test cases for staff

Test case	Check item	Test case's objective	Test Data /input	Expected result	Test result	Actual result
1	Logo	Appearance of logo	-	Appearance of logo	pass	
2	Login page	Log in to the system	Email, Password	Logged in	pass	
3	login with incorrect email and password	Staff with incorrect email and password cannot enter the system	Wrong email or password	Toast message, Invalid email, or password	pass	
4	Staff home	Staff should be able to choose one of the actions	click any icons in the page and it should work	Appearance of icons	pass	
5	choose product page	Should show all the product information to the staff	Click on product icon	The appearance of all product information	pass	

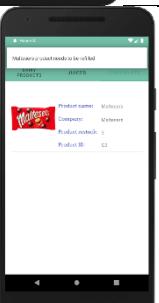
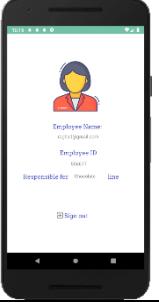
6	choose statistic page	Should show all the best and least product in day, month, and year to the staff	Click on the best (day, month, year) or least product	The appearance of all statistic information	pass	
7	choose Restock page	Should shows products that need to restock and illustrates the line and shelf number	Click on restock icon	The appearance of the product that need to be restocked	pass	
8	Notifications	Should show all the alerts sent to the employee for an alert for an empty shelf	Click on notifications icon	The appearance of the product that need to be restocked	pass	
9	Clicking on setting	staff can be able to show his/her information	Click on setting icon	staff see his/her information	pass	
10	Clicking on sign out	The system goes to login page	licking on sign out icon	Go to login page	pass	

Table6.1 Test cases for staff

6.3 Test results

Software testing is a test required at the end of any software project to check and make sure the software is working in the right way.

In our project "haseef", we applied the system testing and the acceptance testing.

As shown in appendix A we drew a pie chart to show the passed and the failed tasks in our Application. We had 46 tasks and all of them worked successfully as we expected.

As we applied the acceptance test, we asked 2 users to perform the most important 5 tasks in our Application, 4 of the tasks were clear to the users and only one task that was not very clear to the users, it was the sign out button, one of the users suggested that if we had a sign out button in the tool bar that would be easier.

CHAPTER 7

CONCLUSION

7.1 Evaluation

Haseef is designed to be applied in commercial stores, its objective is to notify the staff about out-of-stock events to enhance the shelves management and to improve the quality of customer service using IoT technology.

Haseef has fulfilled its objective, and all of its functions are behaving correctly. However, it has one remaining hardware issue, which is the inaccuracy of the measurements due to the instability and high sensitivity of the HX711 module.

Overall, Haseef has a friendly and simple UI and is achieving its purpose.

7.2 Future work

1. Improve the accuracy of the measurements.
2. Upload dust sensor attached with the weight sensor to notify the cleanliness of the shelf to the staff.
3. To reduce cost, weight sensors can be replaced with cameras to track the products using the image processing technique.
4. Analyze the out-of-stock events to predict the customer habits and get insights for marketing plans such as the best time to sell a specific product and the associations between products.

References

- [1] Oracle, "What Is IoT?," may 2020. [Online]. Available: <https://www.oracle.com/internet-of-things/what-is-iot.html>.
- [2] "Python Advantages and Disadvantages – Step in the right direction," 2020. [Online]. Available: <https://techvidvan.com/tutorials/python-advantages-and-disadvantages/>.
- [3] Walmart, "Walmart's New Intelligent Retail Lab Shows a Glimpse into the Future of," 2020. [Online]. Available: <https://corporate.walmart.com/newsroom/2019/04/25/walmarts-new-intelligent-retail-lab-shows-a-glimpse-into-the-future-of-retail-irl>.
- [4] "Kannita, shelf content and consumer behavior analytics," 2020. [Online]. Available: <https://www.startuphub.ai/startups/kannita/>.
- [5] AWM, "retail solutions," 2020. [Online]. Available: <https://smartshelf.com/index.html#clients>.
- [6] ICT Quality Management officer, "ICT Test Strategy," 2018. [Online]. Available: file:///C:/Users/hyyof/AppData/Local/Temp/EN-9.+OJ_2018_ICT_9753_Quality+Assurance_Technical+Annex+E+%E2%80%93++ICT+Test+Strategy_001.pdf.
- [7] "Advantages of Java," [Online]. Available: https://www.ibm.com/support/knowledgecenter/ssw_aix_71/performance/advantages_java.html.
- [8] JetBrains , "PyCharm Get started," [Online]. Available: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>. [Accessed Oct 2020].
- [9] "Android developers," Google, [Online]. Available: <https://developer.android.com/>. [Accessed Oct 2020].
- [10] "Arduino - Software," [Online]. Available: <https://www.arduino.cc/en/main/software>. [Accessed Oct 2020].
- [11] P. Walpita, "Software Architecture Patterns — Layered Architecture," 9 jul 2019. [Online]. Available: <https://medium.com/@priyalwalpita/software-architecture-patterns-layered-architecture-a3b89b71a057>.

Appendix A

Test cases planned	Test cases executed	pass	fail
29	29	29	0

