Palestine Polytechnic University
College of Information Technology and Computer Engineering

Palestine Postal Automation

Project Team:

Hamza Al-Herbawi
Rannen Wredat
Raghad Salama
Nidaa Nasser Edin

Supervisor:

Dr. Mohammad Abu Taha

To	
Cc	
Bcc	
Subject	

Hebron - Palestine

2021/2022

الإهداء

قال تعالى:} يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ {]المجادلة: ١١[

إلى من أفضلها عن نفسي ولم لا؛ فلقد ضحت من أجلي

إلى صندوق أسراري ....وحضن آمالي

إلى جنتي التي تمشي ... أمي الغالية

إلى صاحب الوجه الطيب، والأفعال الحسنة

إلى الظهر الذي أسند.... إلى القلب الكبير

الذي مهد دربي لطريق العلم

والدي العزيز

إلى ملاذي وسندي ودعمي.. إلى ينبوع التفاؤل

إخوتي وأخواتي

إلى رفقاء الدرب إلى من طارت أحلامنا سويا إلى النجوم

إلى الأصدقاء والزملاء

إلى أرواح الشهداء الخالدين ... إلى من رووا بدمائهم تراب فلسطين

إلى أسرى فلسطين... أسرى الحرية

وإلى من يستحق الإهداء

شكر وتقدير

قال تعالى:} وَمَن يَشْكُرْ فَإِنَّمَا يَشْكُرُ لِنَفْسِهِ {]لقمان:[١٢

"وقال رسوله الكريم صلى الله عليه وسلم: "لا يَشْكُرُ اللَّهَ مَنْ لا يَشْكُرُ النَّاس

نحمد الله سبحانه وتعالى على إحسانه وتوفيقه على ما أسداه علينا من نعم لا تعد ولا تحصى ونصلي ونسلم على نبي هذه الأمة الذي جاء رحمة للعالمين. وبعد، فإننا نتقدم ببالغ شكرنا وتقديرنا للدكتور المشرف محمد أبو طه على ما قدمه من نصح وإرشاد لإنجاز المشروع، ونشكر جميع الأساتذة المحترمين الذين رافقونا في رحلة العلم ولم يدخروا جهداً في أداء أمانتهم، كما نشكر جميع أصدقائنا وزملائنا الذين كانوا لنا خير مساعد ومعين.

وأخيراً، نسأل الله دوام فضله، ونرجو أن يكون من نتاج هذا الجهد المتواضع بعض العلم الذي ينتفع به، وأن يكون هذا العمل خطوة متواضعة لطريق أكثر إثراء لمزيد من الأعمال.

الملخص

في ظل التقدم الكبير للتكنولوجيا في جميع مجالات الحياة، فقد أضحت ذات دور مهم في حياتنا بحيث لا يمكن أن يمر يوم دون استخدامها، وذلك قد انعكس على طريقة التواصل والمراسلة فيما بيننا سواء كنا في نفس المدينة أو خارجها، ومن هنا كانت فكرة تحويل عمليات البريد الورقية إلى عمليات مؤتمتة من خلال إنشاء موقع إلكتروني وتطبيق هاتف يساعد ويحقق هذا الهدف.

نظام البريد الفلسطيني يعتمد في معالجة العمليات البريدية على الحد الأدنى من استخدام التكنولوجيا فيما يتم باقي العمل بشكل ورقي، والنظام المقترح في هذا المشروع يعتمد على استبدال بعض الأعمال الورقية الحالية بأخرى إلكترونية مما يساعد في التحول من العمل الورقي إلى العمل المؤتمت كاستحداث نظام صندوق البريد الافتراضي ونظام تتبع البريد غير المعياري (non-standard) وهو البريد غير المدرج ضمن نظام التتبع العالمي.

وقد تضمن المشروع إنشاء موقع إلكتروني داخلي لنظام البريد الفلسطيني بمستويين اثنين من الصلاحيات لكل من المدراء والموظفين، بحيث أعطي المدراء صلاحيات كاملة على النظام الداخلي فيما اقتصرت الصلاحيات

المتاحة للموظفين على العمليات الروتينية اليومية في استلام البريد وإرسال الإشعارات، كما تضمن المشروع إنشاء تطبيق هاتف لمستخدمي نظام البريد بهدف تسهيل متابعة الطرود الخاصة بهم.

Abstract

In light of the great progress that technology has achieved in all areas of our life, it has play an important role in our lives so that no day passes without using it, and this is reflected in the way of the communication and correspondence between us, whether we are in the same city or outside,

Hence, the idea of transforming mail operations from paper-based processes into automated processes by creating a website and mobile application that helps and achieves this goal.

The Palestinian postal system depends in handling postal operations on a minimal use of technology while the rest of the work is done in paper form,

The proposed system in this project depends on replacing some of the current paperwork with electronic ones, such as the development of the virtual mailbox system and the non-standard mail tracking system - the mail that is not included in the global tracking system-.

Our project included the creation of an internal website for the Palestinian Postal system with two levels of authority for both managers and employees, whereby managers were given full powers over the internal system, while the authority of an employees were limited to daily routine operations of receiving mail and sending notifications. The project also included the creation of a phone application for postal system users in order to facilitate the tracking of their parcels.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

Technology has played an important role in our lives that our day cannot pass without the use of technology, it has become common in all homes, and everything has become connected with technology.

We, in turn, realize the importance of technology at this time and that it has become a huge part of everyone's life. We will automate some of the Palestinian postal operations which are large and complex operations for both employees and clients, such as tracking mail from the moment it arrives at the central mail and its distribution to branch offices in cities, as well as the reservation process of mailboxes, these operations are difficult and when automated it will be easier for the employee and client.

1.1. Problem

1.1.1 Problem statement

Postal service development and its consolidation have led to an increasing number of services for the citizens which means the organization must automate operations if it is to serve them effectively. Postal tracking works in two ways, the first way is to fall under a standard system (UPU), and enter the tracking number to get the tracking result of your shipment, including the dates, locations, country of origin, and destination. The second method uses the EXCEL program where the data for the package is manually added (non-standard) and this cannot be entered into the international tracking system. Additionally, some clients should reserve a mailbox and track this box periodically, which results in some complications and wasted time.

1.1.2 Problem solution

Mail operations will be simplified and employees will be able to track packages more easily through a website/application similar to the international postal system mainly for packages with tracking numbers non-standard where they are exported, distributed, and archived electronically. In addition, the website will allows the client to save time and effort by receiving notifications of any arriving mail to him.

1.2 Overview

Post office mail is a basic communication between members of Palestinian institutions, as there is a continuous and strong need for communication whether between Palestine and other countries or within the country itself.

Whereas the ministry of communications and information technology seeks to meet all the requirements of employees and clients in more than one hundred offices distributed throughout the West Bank to organize exchange operations, documenting, and auditing. Thus, all of these operations will be converted into computerized operations away from the traditional paper-based methods. Also, it seeks to find virtual boxes representing those boxes in the local post office to make it easier for clients to monitor post office boxes.

1.3 Motivation

1. Huge amount of time wasted in entering all that information about dispatches and items into the Excel program.
2. Searches in Excel are not comparable to ease of searching and sorting through the website that we created.
3. The low quality of information archiving services through Excel calls for designing a site specifically for the Palestinian Post to preserve and archive client's information.
4. There are many steps during or after which you must print a receipt, such as "mail delivery Transaction" and these are not available in Excel.
5. The user tends to forget to check his mail, or there is a great burden and wasted time on employees in the process of communicating with clients to remind them to take the mail that reached their inbox. Hence, the need to make an application that contains virtual boxes for users and is linked to the website.

1.4 Scope of the project

First, we hope to use the project as a tracking system for packages headed from the central mail located in Jericho to mail located in Hebron, and later we hope to apply the tracking system to the rest of the mail located in all governorates of Palestine.

1.5 Objective

- Save time and effort for employees so that unofficial mail is archived automatically.
- Facilitate dealing with mailboxes to save time and effort for citizens.
- Create a computerized mail system that reduces the use of Excel.
- Save time and burden on employees by creating an application to simulate mailboxes and send notifications and reminders to its owners.
- Ease of filling in the package receipt and printing it directly from the site.
- We also facilitate the process of searching for relevant packages through serial numbers and archiving data.

1.6 System description

We are creating a website and an application for Palestinian mailboxes that contains an interface on the site of the Palestinian Post. There are a number of boxes in the mail so that the boxes reserved for clients can be identified. The mail application sends an SMS message when a package arrives.

At the end of the box reserve period, the local postal office sends a special reminder to the client that the period has expired.

The website of the Palestinian Post also tracks (non-standard) packages that arrive at the central mail (which they cannot track through the international mail system IPS). According to the cities of the West Bank, the packages that belong to one city are sorted and sent to the main center in the city. When each package arrives at the sorted cities, an SMS is sent inside the site that the packages have been delivered to their correct location and all of this is tracked through the system that we work on, which It will be affiliated with the Palestinian Postal System.

TimeLine /Project Scheduling

Table 1: Project tasks

Task number	Task names	Time required in week
1.	System planning and information gathering	4
2.	System requirements determining	4
3.	Describe and analyze website requirements	4
4.	System design	6
5.	Programming and system development	8
6.	System testing	4
7.	Documenting	Along the working period

Table 2: Project timeline

task	Time in weeks																
	First semester								🟨	Second semester							
	2	4	6	8	10	12	14	16	🟨	2	4	6	8	10	12	14	16
System planning and information gathering	🟩	🟩							🟨								
System requirements determining			🟩	🟩					🟨								
Describe and analyze website requirements					🟩	🟩			🟨								
System design							🟩	🟩	🟨	🟩							
Programming and system development									🟨		🟩	🟩	🟩	🟩			
System testing									🟨			🟩	🟩	🟩	🟩	🟩	
Documenting	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
🟩 Estimated time to finish the task									🟨 Holiday between semesters								

1.7 Summary

This chapter consists of a discussion of the most important requirements, which encompass the general idea of the project and the system we build. The project describes the main problem and how it can be solved, the goals we wish to achieve, and the benefits that will accrue to the citizenry through this system (website).

Chapter 2: Requirements Specification

2.1. Introduction

In this chapter, users of the website, functional and non-functional requirements, use case diagram, class diagram and detailed description of website requirements will be presented.

2.2. Alternative solutions

2.2.1. Desktop Application

1. Features of this desktop application:
    - Support for Big Files.
    - Integration with the client OS.
    - Easier to work with offline.
2. Disadvantages of this desktop application:
    - Difficult to update and maintenance is the responsibility of the end user.
    - Diversity of operating systems and you'll need to develop different versions.
    - Not portable.

2.2.2. Website

1. Features of this website:
    - Easy to use by users.
    - Access the site from any type of device (computers and smart devices).
    - Ease update at any time.
    - Can be opened from a browser running on any operating system.
2. Disadvantages of this website:
    - Permanent need to connect to the Internet.

2.2.3. Mobile Application

1. Features of this website:
    - Ease of use.
    - Available to the user always through the mobile phone.
    - Easy to update.
2. Disadvantages of this website:
    - Permanent need to connect to the Internet.

2.3. Proposed system

After looking at the solutions that can be selected, a website has been chosen to solve the irregular mail problem, because it is the best solution for office work inside post offices and is easier to maintain and update. a mobile application was chosen to solve the problem of package tracking because it is easy to access any time.

2.4. Users

1. Website:
    1. User:
        - Employees can monitor packages and add information about the package.
        - Employees can add information about the post office box and send a SMS to the client.
    2. Admin:
        - can create a new user, enable user and disable the user.
2. Mobile Application:
    1. User (visitor):
        - client can add their own serial number to know the status of your package.

2.5. Requirements

2.5.1 functional requirements

- Website functional requirements
    - For User

1. Ability to log in to his account.
2. Ability to view all packages.
3. Ability to add information about the package.
4. Ability to delete a package.
5. Ability to edit a package.
6. Ability to view all postal boxes.
7. Ability to add information about postal boxes.
8. Ability to delete a postal box.
9. Ability to edit a postal box.
10. Ability to view all invoices.
11. Ability to add an invoice.

12. Ability to edit an invoice.
13. Ability to print an invoice.
14. Ability to delete an invoice.
15. Ability to view all areas.
16. Ability to add an area.
17. Ability to edit an area.
18. Ability to delete an area.
19. Ability to send SMS about postal boxes.
20. Ability to log out from his account.

- For Administrator:

1. Ability to log in to his account
2. ability to perform all the functions that the users can do.
3. Ability to add a user's account.
4. Ability to add roles to users.
5. Ability to delete any account.
6. Ability to log out to his account.

- Mobile application functional requirements

  - For visitors:

    1. Search for their own serial number to know the status of your package.
    2. Ability to communicate with the Palestinian Post.
    3. Ability to browse the Palestinian Postal Services.

2.5.2 Nonfunctional requirements

Non-functional requirements are the basis for the success of the functional requirements because they are based on their support and quality, either directly or indirectly.

Non-functional requirements include the following:

1. Ease of dealing with the website/Application.
2. The website/Application design interface should be easy to handle and use.
3. The interface should be simple, with comfortable colors for the user

4. The website should be responsive when it is opened from a mobile browser.
5. The website should be provided with filters that facilitate the search for used items.

Security: The website/Application have been able to maintain trust between users by following these methods during the establishment of the website:

1. Prevents SQL injection attacks and XSS attacks, this is achieved by Laravel framework.
2. Every user can provide his personal information to prove his identity to other users.
3. Ensure that package tracking processes and parcel information are correct.

2.6. functional requirements specifications

- Website functional requirements

Functional requirements for user:

1. Ability to log in to his account.

requirement name	Login
Main user	- User
Target	- Enable the user to access permission available to him in the website
Conditions	- To have a account/permission from admin
Procedures	- open the main screen of the website. - Write your username and password. - Click on the login button. - Login to the website and explore the website
Exceptions	- The username or password is incorrect. - the account has been disabled - Inability to create a session between the server and the user's browser
Exception’s solution	- Re-enter the correct data. - connect with your administrator - Modify browser settings.

2. Ability to view all packages / invoices / postal boxes.

requirement name	- view all packages/ invoices / postal boxes
Main user	- User
Target	- Enable the user to view all packages invoices / postal boxes.
Conditions	- Be logged in.
Procedures	- login into the website. - choose “Dashboard” from the menu. - click on “view all packages, invoices or postal boxes “.
Exceptions	- you don't have permission to “view all packages, invoices or postal boxes “ from the administrator.
Exception’s solution	- Communicate with your administrator.

3. Ability to add information about the package.

requirement name	- add a package information
Main user	- user
Target	- Add and control package operations and related information.
Conditions	- logged in with a permission.
Procedures	- Login to his account. - enter the home page. - click a “package “page. - press the ‘add’ button.

4. Ability to delete a package.

requirement name	- delete a package
Main user	- User
Target	- Enable them to delete a package from the website
Conditions	- You have permission to “delete a package” from the administrator
Procedures	- Login to your account. - enter the home page. - click a “package “page. - choose the package that you want. - press the ‘delete’ button.
Exceptions	- you don't have permission to “delete a package” from the administrator.
Exception’s solution	- Communicate with your administrator.

5. Ability to edit a package.

requirement name	- Edit a package
Main user	- User
Target	- Enable them to edit a package from the website
Conditions	- must have a permission to “edit a package” from the administrator
Procedures	- Login to your account. - enter the home page. - click a “package “page. - choose the package that you want. - press the ‘edit’ button.
Exceptions	- you don't have permission to “edit a package” from the administrator.
Exception’s solution	- Communicate with your administrator.

6. Ability to add information about postal boxes.

requirement name	- Add information to mailboxes.
Main user	- User
Target	- Add information to reserved mailboxes.
Conditions	- Be logged in
Procedures	- Go to the boxes tab. - Click on the new button - enter a number, phone, username, and area for a specific user. - press saves.
Exceptions	- The number is out of the specified range
Exception's solution	- Re-enter and choose the box number between 1-1000.

7. Ability to delete a postal box.

requirement name	- delete a postal box.
Main user	- User
Target	- Enable them to delete a postal box from the website
Conditions	- You have permission to "delete a postal box" from the administrator
Procedures	- Login to your account. - enter the home page. - click the "boxes "page. - choose the package that you want. - press the 'delete' button.
Exceptions	- you don't have permission to "delete a package" from the administrator.
Exception's solution	- Communicate with your administrator.

8. Ability to edit a postal box.

requirement name	- Edit a postal box
Main user	- User
Target	- Enable them to edit a postal box from the website
Conditions	- You have permission to “edit a postal box” from the administrator
Procedures	- Login to your account. - enter the home page. - click the “boxes “page. - choose the box that you want. - press the ‘edit’ button.
Exceptions	- you don't have permission to “edit a box” from the administrator.
Exception’s solution	- Communicate with your administrator.

9. Ability to add an invoice.

requirement name	- Add an invoice.
Main user	- User
Target	- To add an invoice
Conditions	- Be logged in
Procedures	- choose “invoices” from the menu. - Click on the new button - enter a ‘Package Num’,’ Amount Paid’, and ‘Recipient’. - press saves.
Exceptions	- The number is out of the specified range
Exception’s solution	- Re-enter and choose the ‘Amount’ between 3-10.

10. Ability to edit an invoice.

requirement name	- Edit an invoice
Main user	- User
Target	- Enable them to edit an invoice from the website
Conditions	- must have permission to “edit an invoice” from the administrator
Procedures	- Login to your account. - click the “invoices “page. - choose the invoice that you want. - press the ‘edit’ button.
Exceptions	- you don't have permission to “edit an invoice” from the administrator.
Exception’s solution	- Communicate with your administrator.

11. Ability to delete an invoice.

requirement name	- delete an invoice.
Main user	- User
Target	- Enable them to delete an invoice from the website.
Conditions	- must have a permission to “delete an invoice” from the administrator.
Procedures	- Login to your account. - click the “invoices “page. - choose the invoice that you want. - press the ‘delete’ button.
Exceptions	- you don't have permission to “delete an invoice” from the administrator.
Exception’s solution	- Communicate with your administrator.

12. Ability to add an area.

requirement name	- Add an area.
Main user	- User
Target	- To add an area
Conditions	- Be logged in
Procedures	- Choose “areas” from the menu. - Click on the new button - Enter a ‘Name’. - Press saves.
Exceptions	- The name must not be empty.
Exception’s solution	- Enter a name.

13. Ability to edit an area.

requirement name	- Edit an area
Main user	- User
Target	- Enable them to edit an area from the website
Conditions	- Must have permission to “edit an area” from the administrator
Procedures	- Login to your account. - Click the “areas “page. - Choose the area that you want. - Press the ‘edit’ button.
Exceptions	- You don't have permission to “edit an area” from the administrator.
Exception’s solution	- Communicate with your administrator.

14. Ability to delete an area.

requirement name	- delete an area.

Main user	- User
Target	- Enable them to delete an area from the website
Conditions	- must have a permission to “delete an area” from the administrator
Procedures	- Login to your account. - click the “areas “page. - choose the area that you want. - press the ‘delete’ button.
Exceptions	- you don't have permission to “delete an area “from the administrator.
Exception’s solution	- Communicate with your administrator.

15. Ability to print an invoice.

requirement name	- Print an invoice.
Main user	- User
Target	- Enable them to print an invoice from the website
Conditions	- Must have a permission to “print an invoice” from the administrator
Procedures	- Login to your account. - Click the “invoices “page. - Choose the invoice that you want. - Press the ‘print’ button.
Exceptions	- You don't have permission to “print an invoice” from the administrator.
Exception’s solution	- Communicate with your administrator.

16. Ability to send SMS messages about postal boxes.

requirement name	Send SMS messages.

Main user	User.
Target	Send SMS messages to the subscribers.
Conditions	Be logged in as a user.
Procedures	- User login with his account. - enter on a boxes page. - enter a number of boxes that need to send to it a SMS in a text field search. - click on the box that is shown. - write a SMS message content - click on send “send ”.
Exceptions	- the box number not found.
Exception’s solution	- re-enter a box number correctly.

17. Ability to log out from his account

requirement name	- Log out.
Main user	- User
Target	- To log out from the user account.
Conditions	- Be logged in.
Procedures	- click on the profile picture - click on the log out button on the box.
Exceptions	- The session was not successfully deleted from the browser.
Exception’s solution	- Try to log out again.

Functional requirements for website administrator:

1. Ability to log in to his account.

requirement name	- Login.
Main user	- Admin
Target	- Enable the admin to access full permission to the website.
Conditions	- To be registered on the website as an Admin.
Procedures	- open the main screen of the website. - Write the username and password. - Click on the login button. - Login to the website and explore the home page.
Exceptions	- The username or password is incorrect. - Inability to create a session between the server and the user's browser
Exception's solution	- Re-enter the correct data. - Modify browser settings.

2. Ability to add a user account and give him roles

requirement name	- Add a user account and give him a role
Main user	- Admin
Target	- Allows creating accounts for employees and making them users of the website with permissions (give him roles)
Conditions	- To be logged in as an admin
Procedures	- login in the website - enter the user's page - Click the" Add new "button - Enter information such as name, password, Role, etc. - Click on the "save "button.
Exceptions	- Username already exists

	- You forgot to add a required field - The username is not acceptable - User password is too short - The admin password is incorrect
Exception’s solution	- Enter another non-duplicate username - Make sure to fill in the required fields - Choose an acceptable username that does not start with numbers or symbols - Re-enter the admin password correctly

3. Ability to delete any account.

requirement name	- Delete accounts.
Main user	- Admin
Target	- The admin can delete the user accounts.
Conditions	- To be logged in as an admin.
Procedures	- login into the website as an admin. - enter the user’s page. - Click on a specific account. - Click the” Disable account” - Click on “Confirm Disable” button
Exceptions	- The account has already been disabled.
Exception’s solution	

4. Ability to log out to his account.

requirement name	- Log out.

Main user	- Admin
Target	- To log out of the account.
Conditions	- Be logged in.
Procedures	- click on the profile picture - click on the log out button on the box .
Exceptions	- The session was not successfully deleted from the browser.
Exception’s solution	- Try to log out again.

- Mobile Application functional requirements

Functional requirements for user (visitors):

1. search for their own serial number to know the status of your package.

requirement name	- package status search.
Main user	- Visitors (user).
Target	- The visitor can know the status of his package by entering the serial number in the search box.
Conditions	
Procedures	- open the main screen of the app. - Click on the search box. - Enter the serial number. - Click on the search icon. - Then see the status of the package.
Exceptions	- the serial number not found.
Exception’s solution	- Re-enter the correct data.

2. The ability to communicate with the Palestinian Post

requirement name	- communicate with the Palestinian Post
Main user	- User (visitors).
Target	- Users of the application can access contact information with the Palestinian Post from
Conditions	Download the application
Procedures	- open the main screen of the app. - open the menu. - Click on the "Contact with the Palestinian Post " - Choose the method of communication you want from the page, whether Facebook, e-mail, or through the phone number
Exceptions	
Exception's solution	

3. The ability to browse postal services.

- Bill Payment Service
- Stamp collecting service
- Non-judgmental service
- Post box service

requirement name	- browse postal services
Main user	- User (visitors).
Target	- Users of the application can access and browse postal services from the menu
Conditions	- Download the application

Procedures	- open the main screen of the app. - open the menu. - Click on the any section you want from the “Postal services “and browse it
Exceptions	
Exception’s solution	

2.7. use case diagram

- website and application use case diagram:

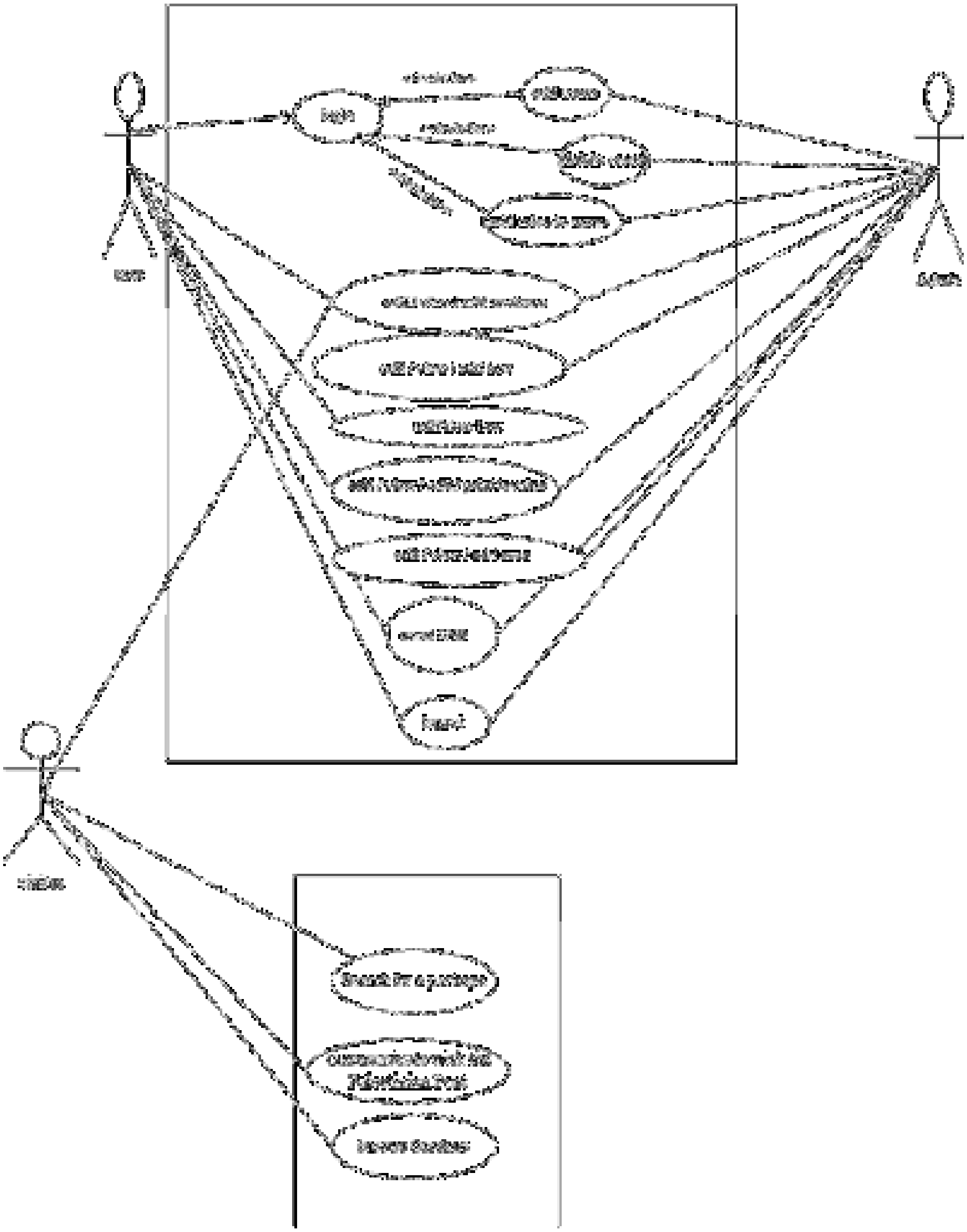


Figure 1: use case diagram

2.8. Summary

This chapter consists of a discussion of the most important requirements, which encompasses functional and non-functional requirements, and detailed description of website requirements and application.

Chapter 3: Software Design

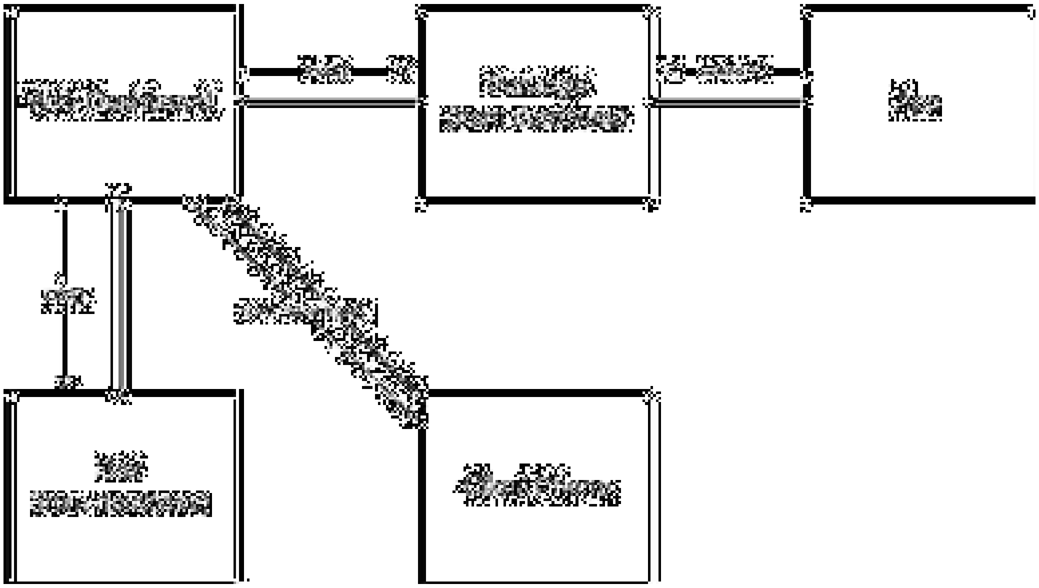
3.1. Introduction

In this chapter, we will talk about the block diagram, the website architecture, the website database description, the database table's description, as well as the design of the website and the application. We will detail its components and parts, as this stage is considered one of the most important stages in building the project because it gives a complete idea of all parts of the system and drawings that show the function of each part and its components.

Using the block diagram, the website architecture, and the website database description is easy for the programmer to do the process of understanding and building the system quickly and correctly.

3.2. Block diagram

It shows general block diagram of the whole system, and how the system's equipment's work with each other to achieve the needed Requirements of our project.



3.3. Design decision

3.3.1. Architectural pattern

An architectural pattern is a description of a good design practice that has been tried and tested in different environments like MVC pattern, layered pattern, and client-server.

We chose this pattern because it is the most suitable one for both the web and the application. The MVC design pattern serves to separate the presentation layer from the business logic, and makes model classes reusable without modification.

In this pattern separates concerns into one of 3 buckets:

A. Model

B. View

C. Controller

3.3.2. architecture for website and mobile application

The architecture of MVC (Model, View, Controller) has been adopted. Since back-end is the Model (RESTful APIs), and front-end is the view and controller, as shown in the following figure:

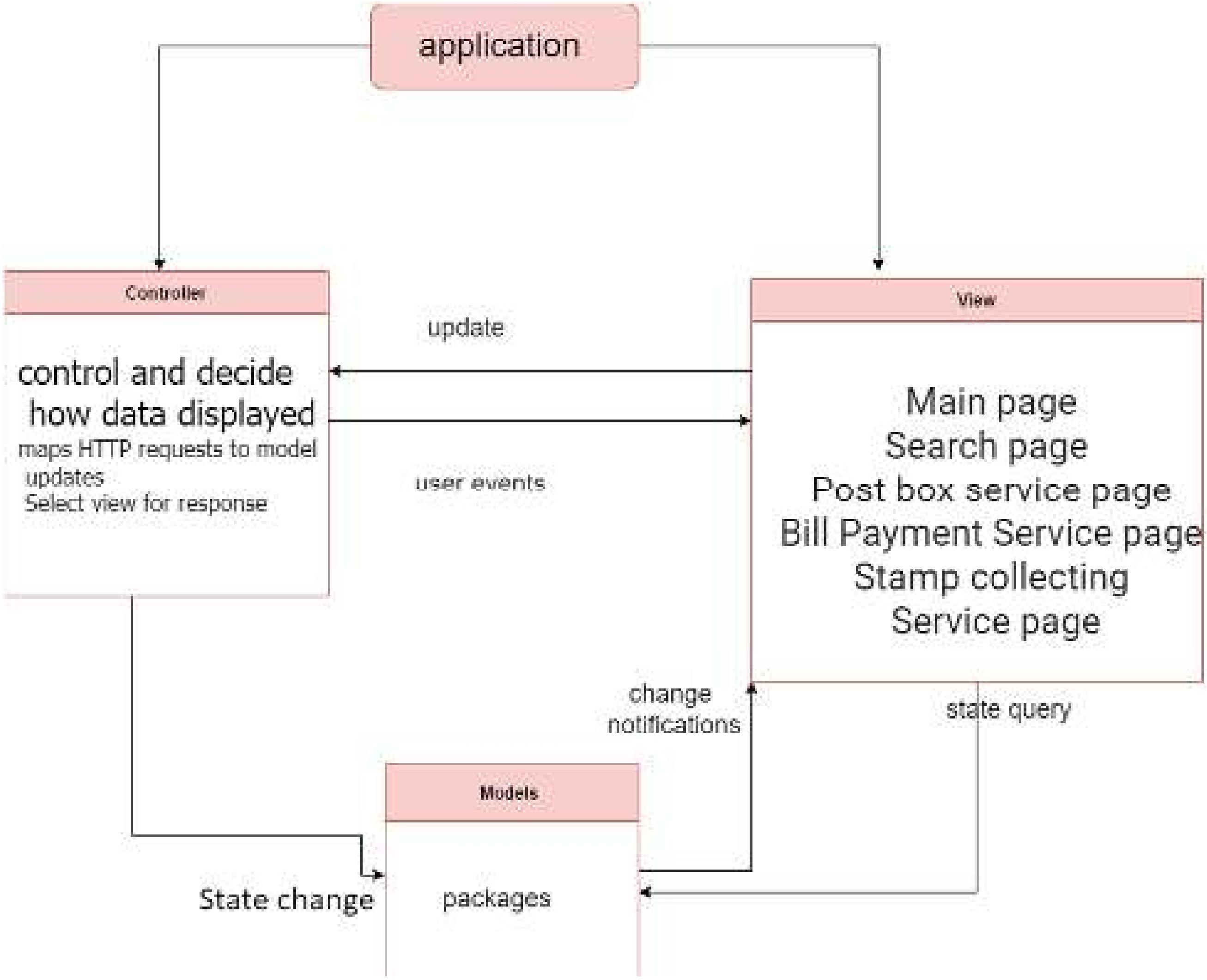
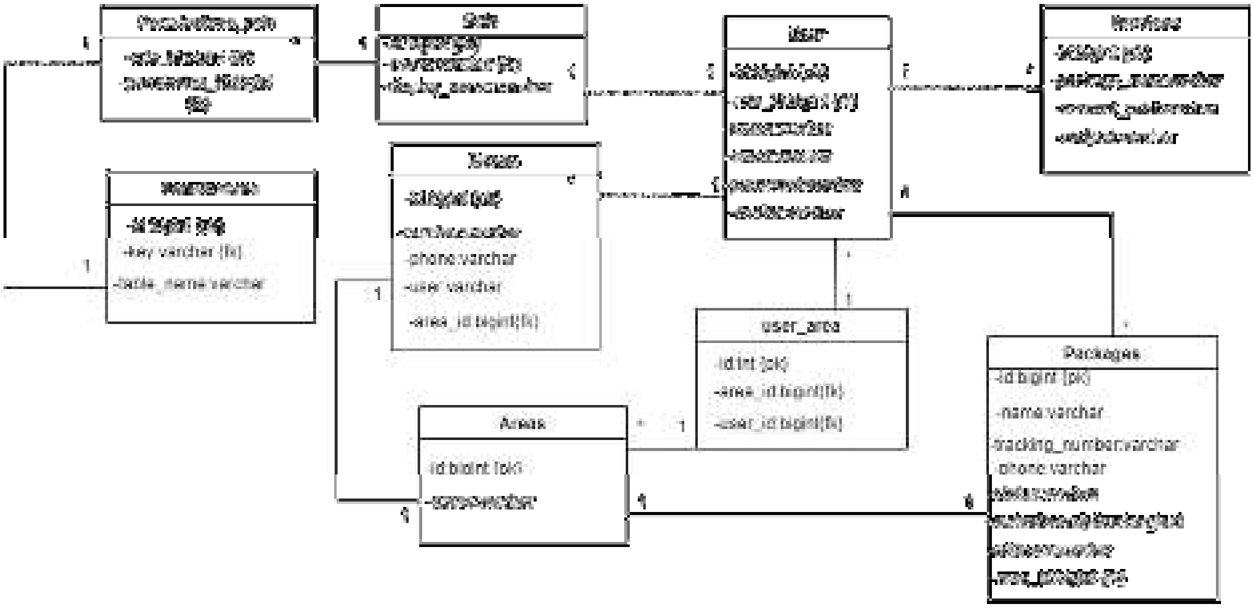


Figure 4: Architecture for mobile application

3.4. Architecture

1. **Model:** It works to manage data and operations related to the website databases and application; this is the normalized database:



Description of the system database: There are several tables associated with the system that are linked through a database.

We will clarify the relationships in this part using the database that reveals the details of the inputs to the system and that shows them in detail

The database tables and relationships between the tables form the basis for building the system.

Table 3: Database tables

The name of the table in the database	**Table description**
Users	Store data related to users monitoring and users' box. Store data related to admins. Store data related to clients.
Packages	Store data related to package that users add
Boxes	Store data related to boxes owned by clients
Invoices	Store commercial invoices for packages
Area	Store the cities that clints and box

Database tables description:

Table 4: Users

Field name	Data type	Size	Null	Unique	Description
Id	Int	11	No	Yes	PK
role_id	Varchar	30	No	No	
Name	Varchar	30	No	No	
email	Varchar	100	Yes	Yes	
password	Varchar	255	No	No	
avatar	Varchar	255	Yes	No	

Table 5: Boxes

Field name	Data type	Size	Null	Unique	Description
id	Int	20	No	Yes	PK
number	Varchar	255	No	yes	
phone	Varchar	16	No	no	
user	Varchar	255	No	no	
area_id	Int	20	No	no	

Table 6: Packages

Field name	Data type	Size	Null	Unique	Description
Id	Int	20	No	Yes	PK

Name	Varchar	255	No	Yes	
tracking_number	Varchar	255	No	Yes	
Status	Varchar	255	No	no	
StatusDescription	Longtext	255	No	no	
Phone	Varchar	16	No	No	
Address	Varchar	255	No	No	
area_id	Int	20	no	no	FK

Table 7: Invoices

Field name	**Data type**	**Size**	**Null**	**Unique**	**Description**
id	Int	20	No	Yes	PK
package_num	Varchar	255	No	No	
amount_paid	Varchar	255	No	no	
recipient	Varchar	255	No	No	

Table 8: Areas

Field name	Data type	Size	Null	Unique	Description
id	Int	20	No	Yes	PK
name	Varchar	255	No	Yes	

Table 9: Permissions

Field name	Data type	Size	Null	Unique	Description
id	Int	20	No	Yes	PK
key	Varchar	255	No	no	
tabel_name	Varchar	255	No	Yes	

Table 10: Permissions role

Field name	Data type	Size	Null	Unique	Description
role_id	Int	20	No	Yes	PK
permission_id	Int	20	No	Yes	

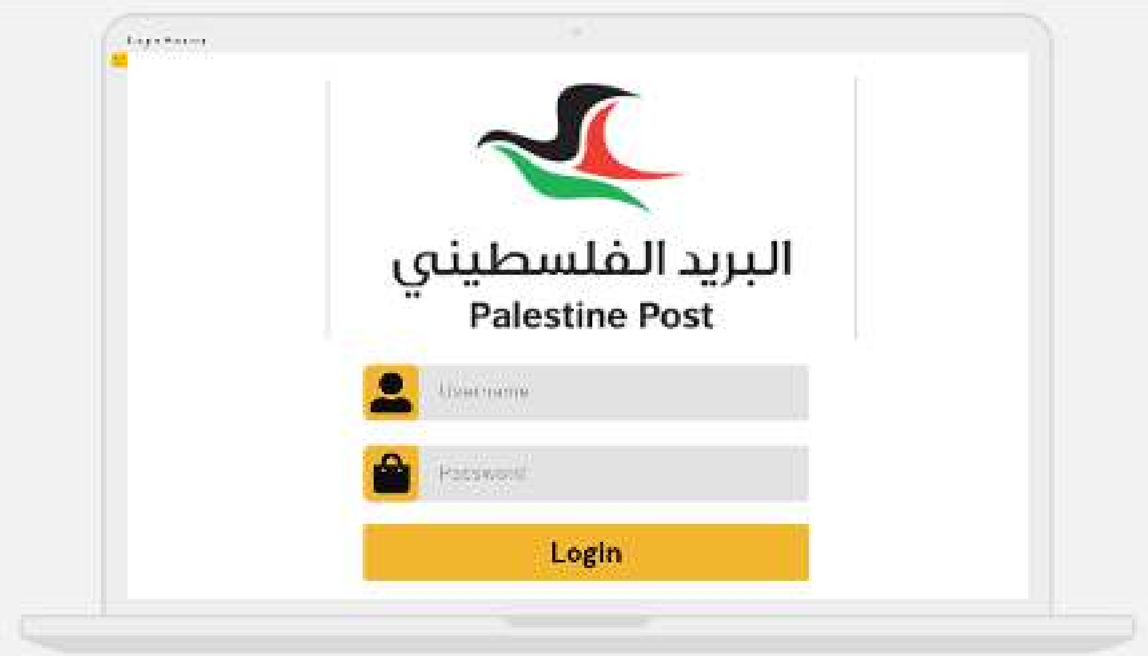
Table 11: Role

Field name	Data type	Size	Null	Unique	Description
id	Int	20	No	Yes	PK
name	Varchar	255	No	no	
display_name	Varchar	255	No	Yes	

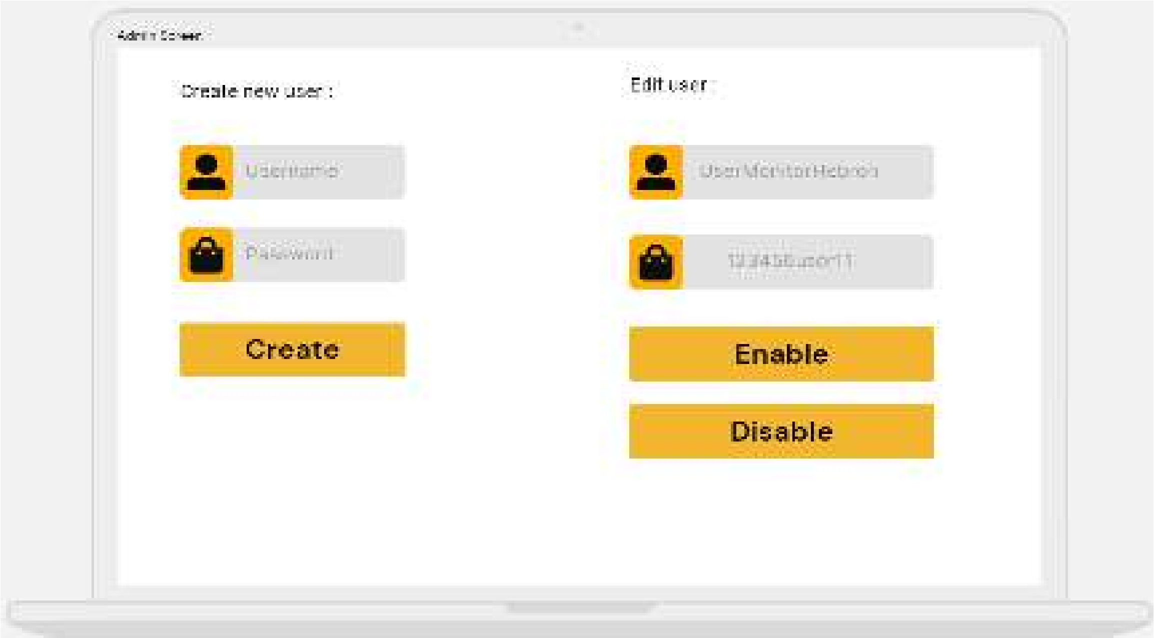
3.5. Interfaces

3.5.1. Interfaces Web

1. The main interface is to enter the admin, user monitor, and postal box user.



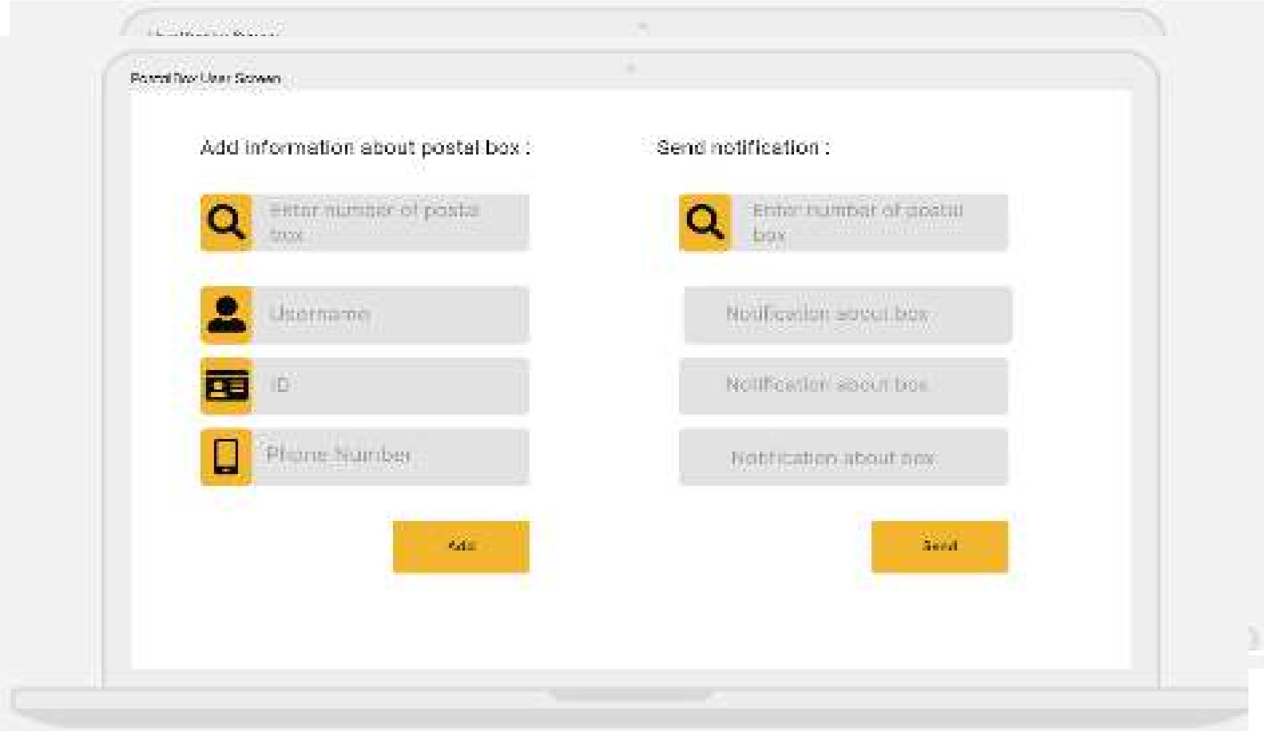
2. The admin's interface after logging in.



UserMonitorHebron

123456user11

3. The user’s monitor interface after logging in.
4. The postal box user’s interface after logging in.



3.5.2. interfaces Application

1. The main interface.

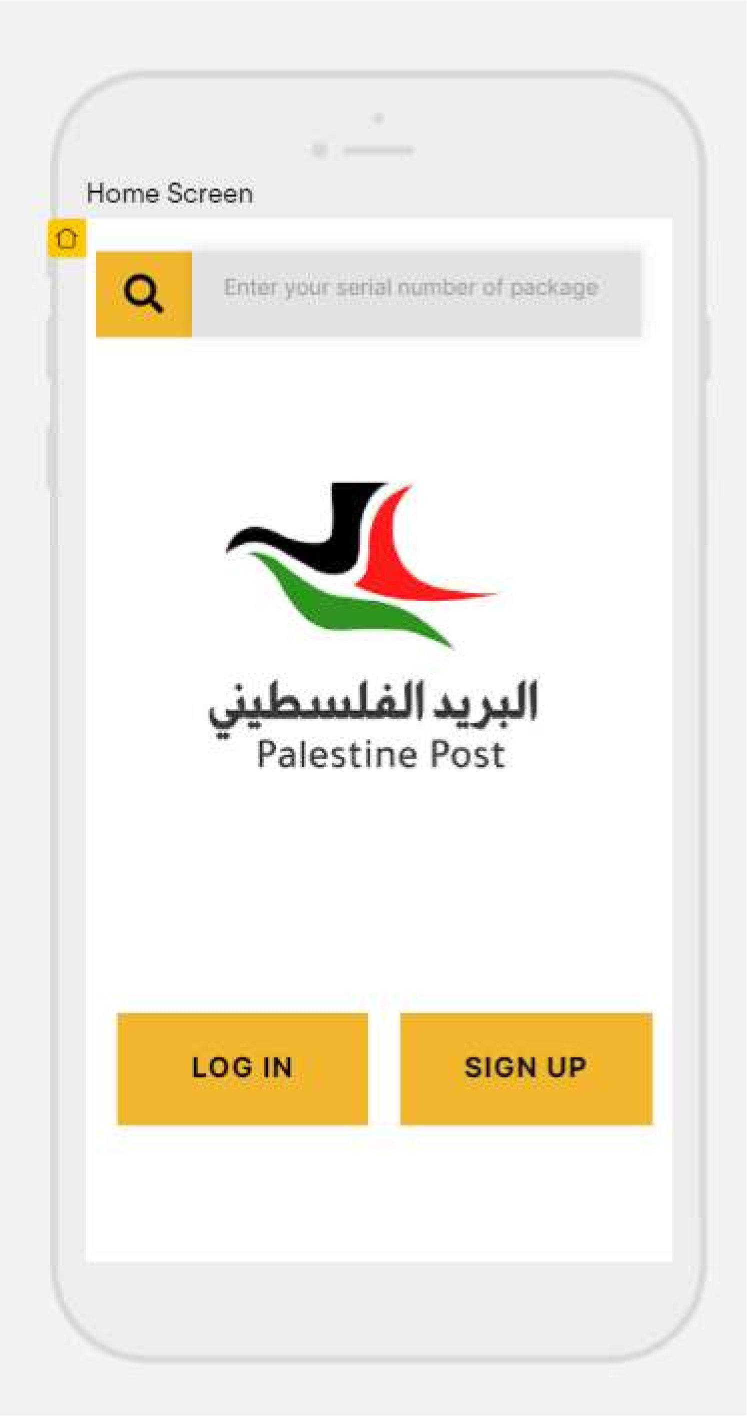


Figure 10: Home page

2. When you select SIGN UP.

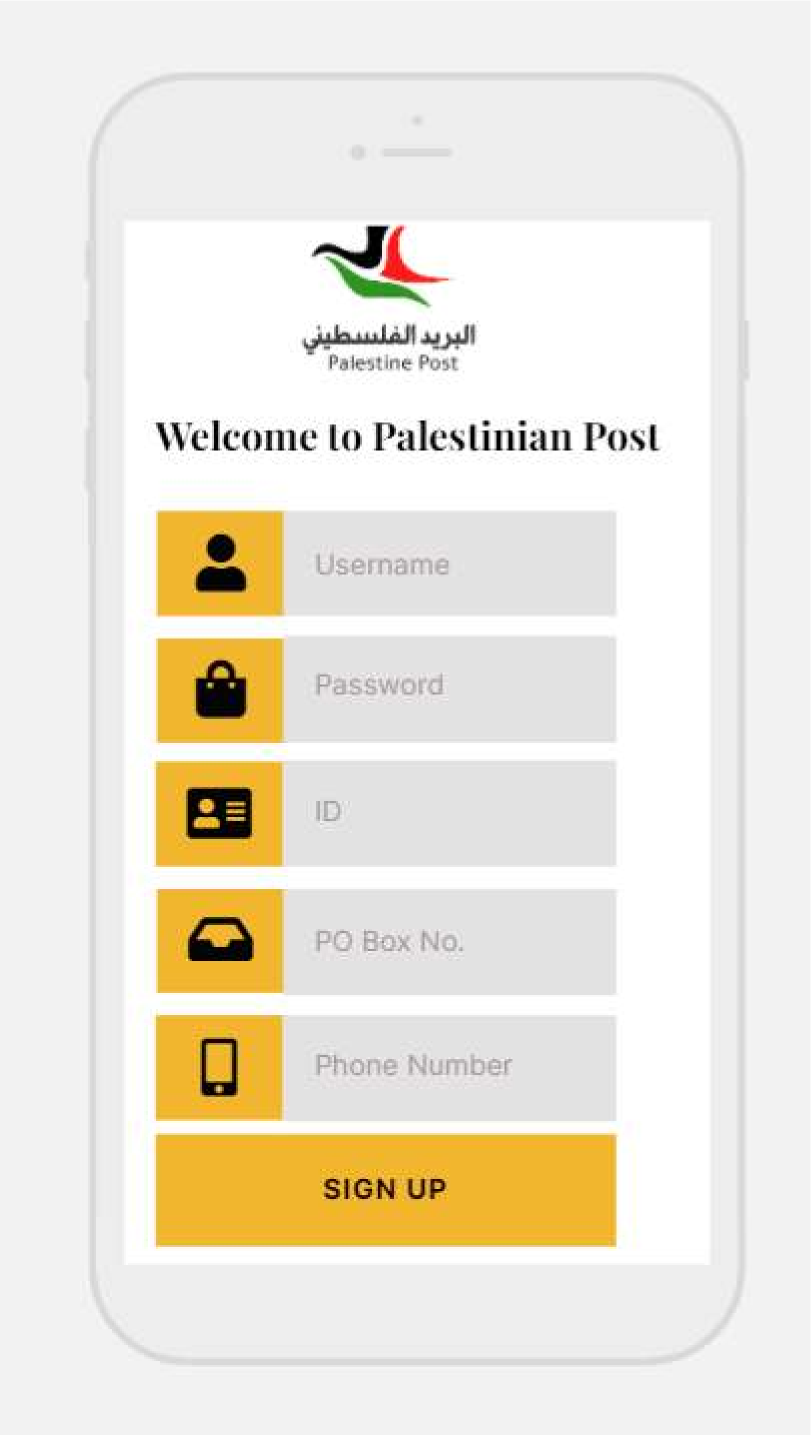


Figure 11: SIGN UP page

3. When you select LOG IN.

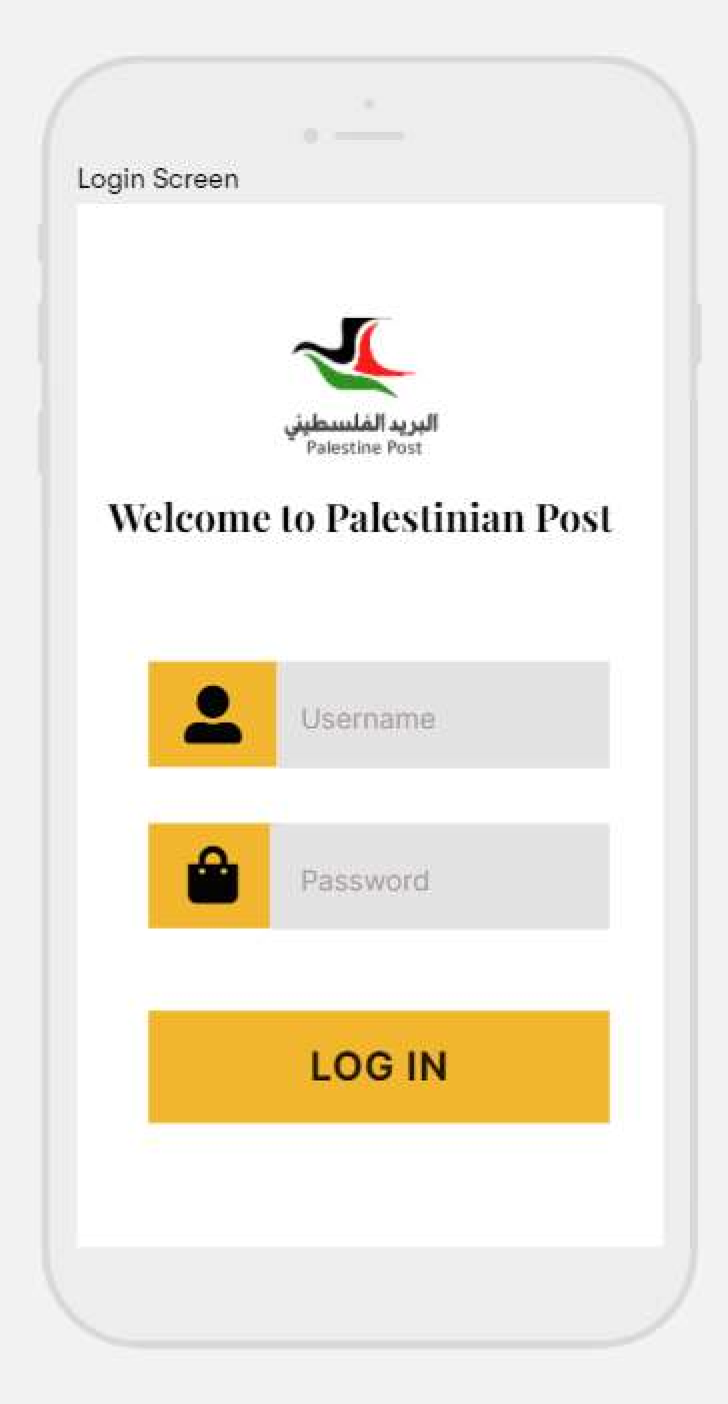
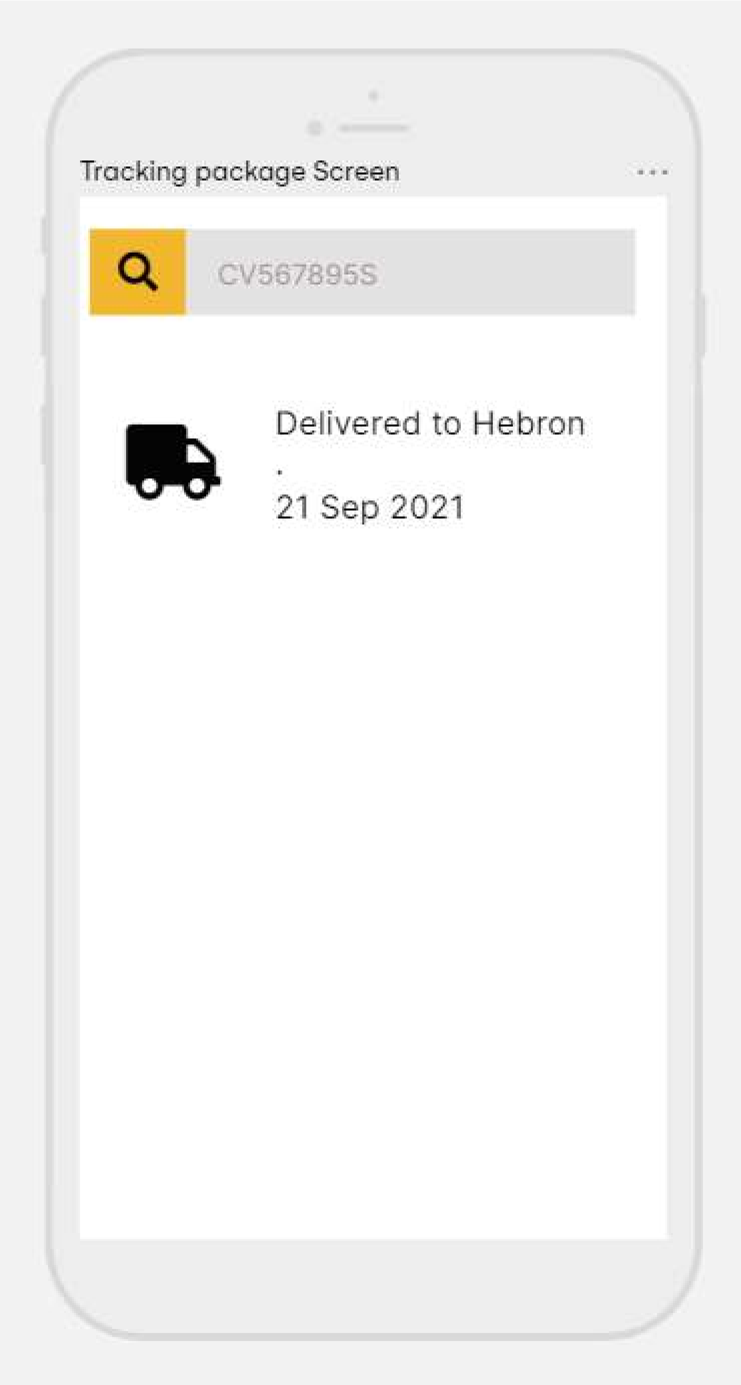


Figure 12: LOG IN page

3. When you search for your package.

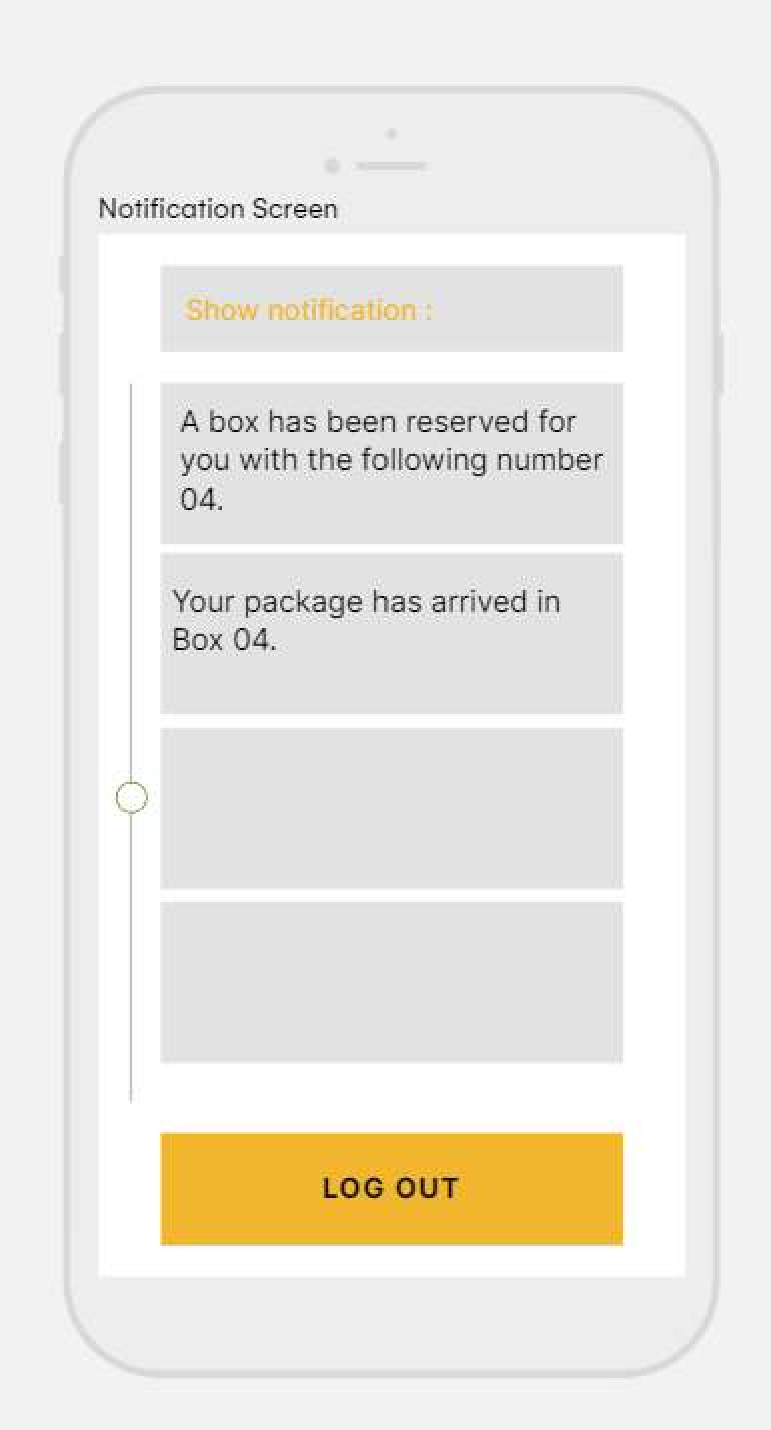


CV567895S

Delivered to Hebron
.
21 Sep 2021

Figure 13: Search page

4. When your Login is completed.



Notification Screen

A box has been reserved for you with the following number 04.

Your package has arrived in Box 04.

Figure 14: Notification page

3.6. Summary

In this chapter we discussed several topics related to design, including: architectural design, database design. Description of design decisions and why we adopt it. and some of the interfaces that we build.

Chapter 4: Building and implementing the system

4.1. Introduction

In this chapter, the stages of building and implementing the system that are moving from the paper stage to the electronic application stage will be identified and to ensure that the system operates properly without problems and that the system operates in full, as this chapter will be presented Supplies necessary to implement the project, and some important screens for the system will be displayed.

4.2. System software resources

The site was developed using the PHP language, in addition to the use of several programs shown as follows:

- **Windows 10**

It is a version of the operating systems for Windows computers, produced by Microsoft Corporation, this system is characterized by its fast performance and support for most software applications in general and Internet software in particular, this is what made it the best in our time, and it also supports software resources that were used in building, developing and operating the electronic system.

Figure 15: windows 10 interface

• **VS Code**

It is a free and open-source code editor developed by Microsoft for many systems "Windows, Linux and macOS". This editor supports many languages and contains a large number of plugins that facilitate the work of writing, editing and analyzing code.

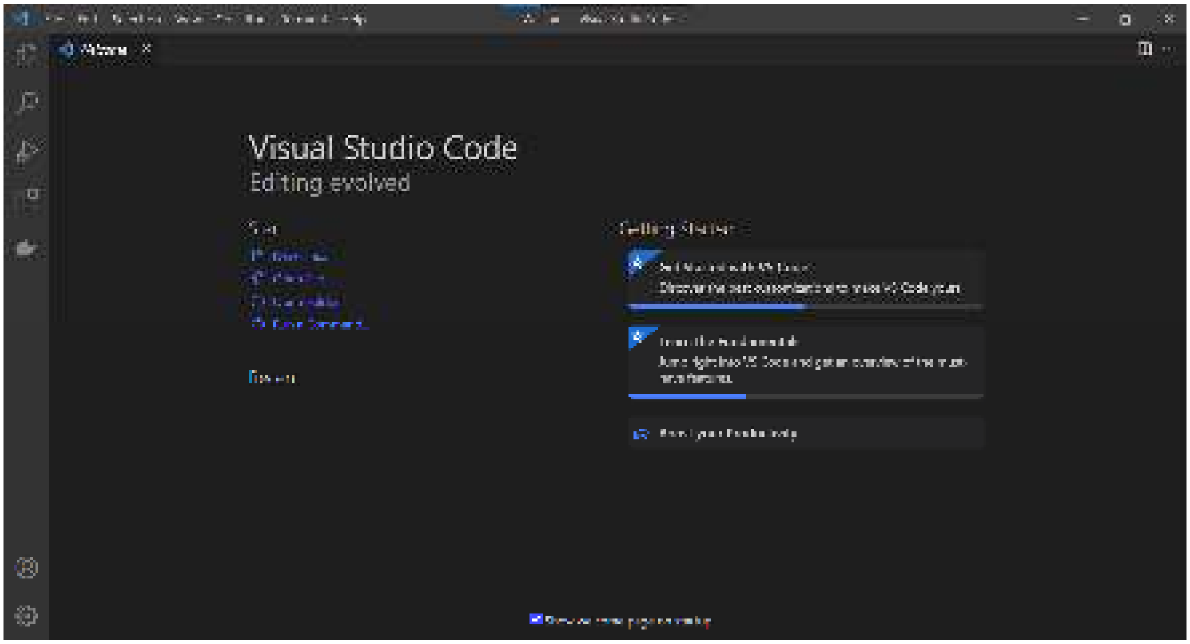


Figure 16: VS Code

• **google docs**

It was used to document the progress of the project.

• · **XAMPP Server**

It is a software package that allows a programmer to run a web server and (MySQL) database server on a personal machine.

- **Android studio**

Android studio is one of the fastest tools for building apps on every type of android device. We used it because it provides world-class code editing, debugging, performance tooling, edible build system, and an instant build/deploy system all allow developers to focus on building unique and high-quality apps. Android Studio comes bundled with:

1. Android SDK tools.
2. Android platform tools.
3. Latest android platform.
4. Latest android system image for emulator.

Figure 17: Android studio interface

- **draw.io**

It is an online or computer diagramming website and software for making flowcharts, diagrams, organizational charts, UML, ER, network diagrams, and more.

- **Composer Program**

It is a text editor dedicated to writing the source code for computer programs and websites.

- **Google Chrome**

It is a web browser that works on most operating systems developed by Google. It was built on the open-source browser, which contains some ready-made components of open source, and was used to ensure that the site works on the most common browsers in the world, in addition to that it has features that facilitate programmers to develop interfaces. front end.

- **php**

It is a programming language that was primarily created to be used for developing and programming websites.

- **Laravel framework:**

Laravel is a web application framework with expressive, elegant syntax. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects, such as authentication, routing, sessions, and caching.

Why do we develop Laravel in our project?

Laravel is a modern, robust, feature-rich and open-source PHP framework for rapid web development and is used largely to provide structure and business logic for web applications. Here are a few benefits of developing web apps with its help:

1. Faster, more parallel development.
2. Build the API once, use it for multiple frontends.
3. Faster, more parallel development.
4. Easier to add functionality later on.
5. Cleaner and more robust code.

Figure 18: Laravel logo

1. Laravel uses in the project:

   - code for edit user:

Figure 19: Example of using laravel

- Flutter framework:

  We used this platform to develop our application.

  - Flutter was chosen due to its adoption of the Dart language, which gives it greater power in building application interfaces and speed in development.

  - It is based on (Widget Base), which facilitates and organizes the application management process.

  - It has the ability to produce applications that work on all platforms (web, desktop, IOS, Android).

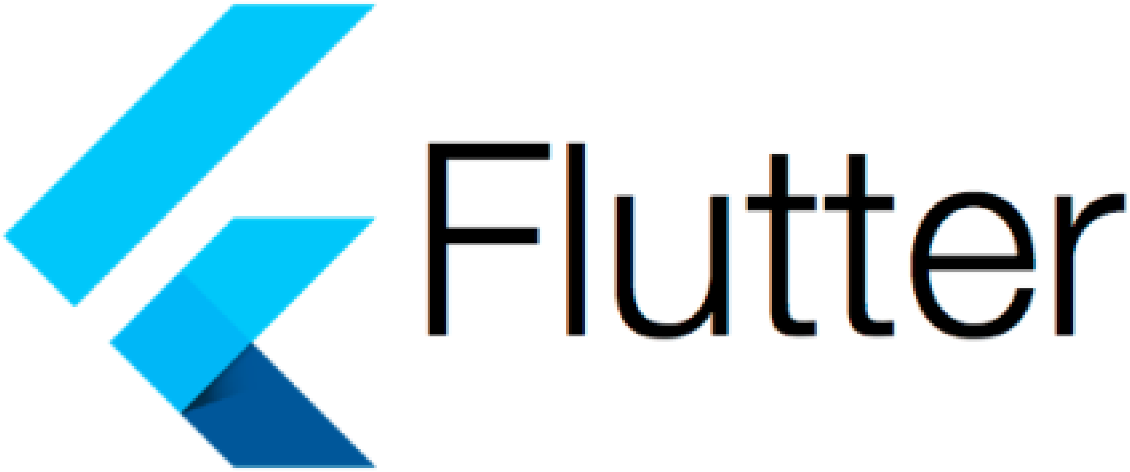


Figure 20: flutter logo

4.3. Software environment and system architecture

The work team relied on building the system architecture on the MVC architecture (and the sixth version of the framework was used).
Laravel (which depends on the PHP language), and the programming environment and how to build the system will be explained in this part:

1. Controllers folder:

> This folder contains the files of the controllers, as each controller contains Several methods for models, and operations on them, such as: deleting, adding and modifying.

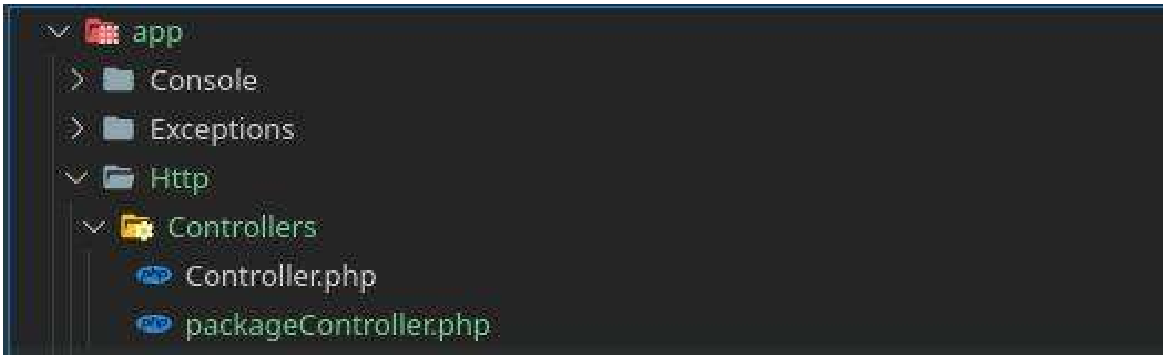


Figure 21: Controller

2. Models:

Model files are files that represent all the tables in the system's database.

The name of the model must be a singular name from the name of the table in order for the controller to deal with it.

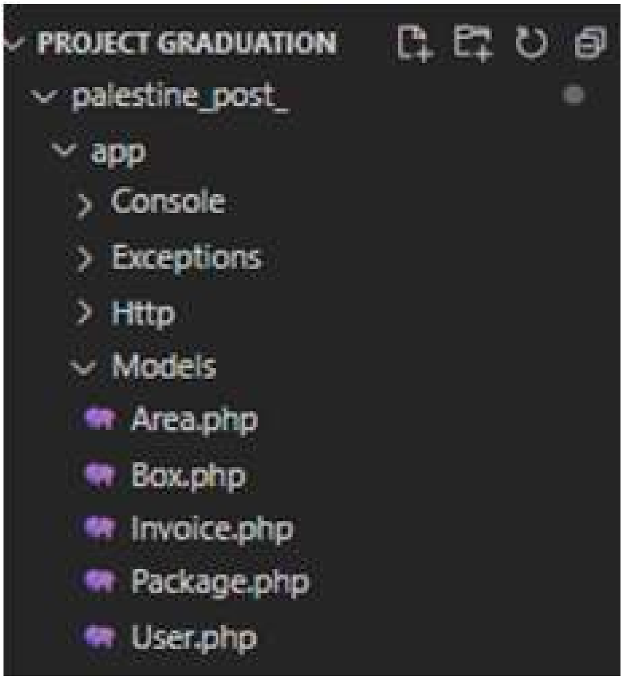


Figure 22: Models

3. Views:

A folder containing files divided into system screens, and for each file there is a set of Private screens have the extension (php. Blade) where the name of the screen is appended to this extension in order to link the screen

with its own controller.

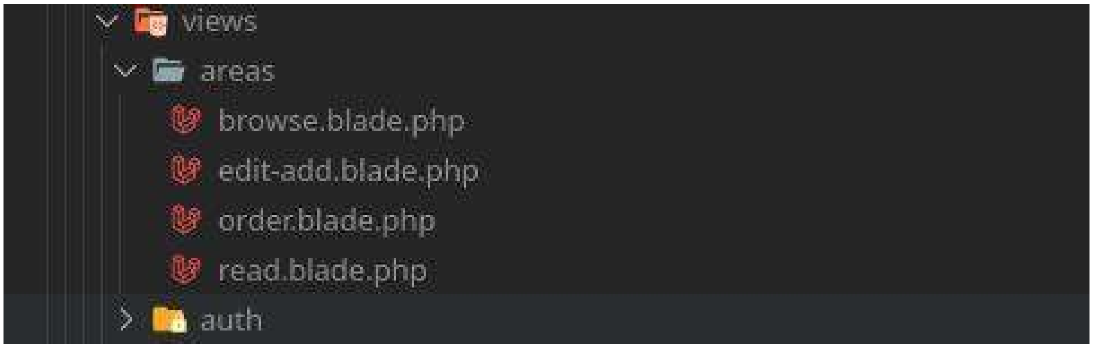


Figure 23: Views

4. Routes:

In this folder all the paths in the path files (php.Web) are specified, were the screens are linked with the models and method in the controllers.

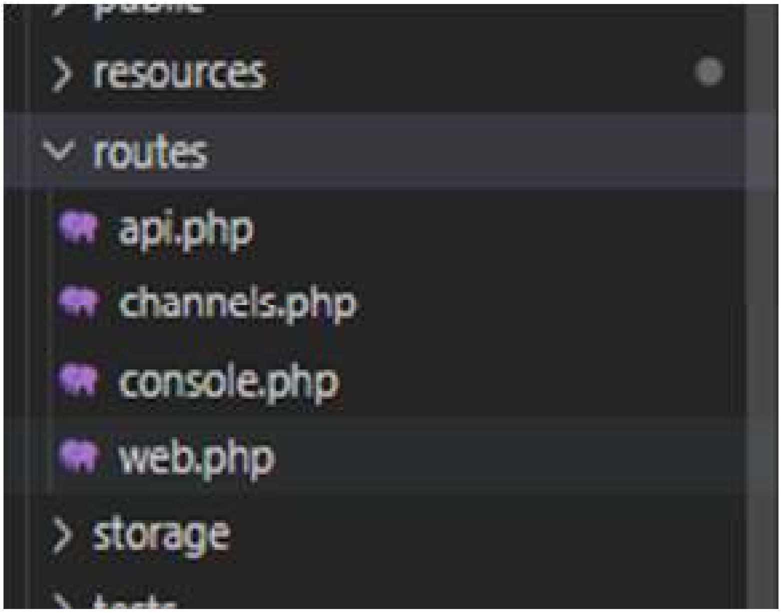
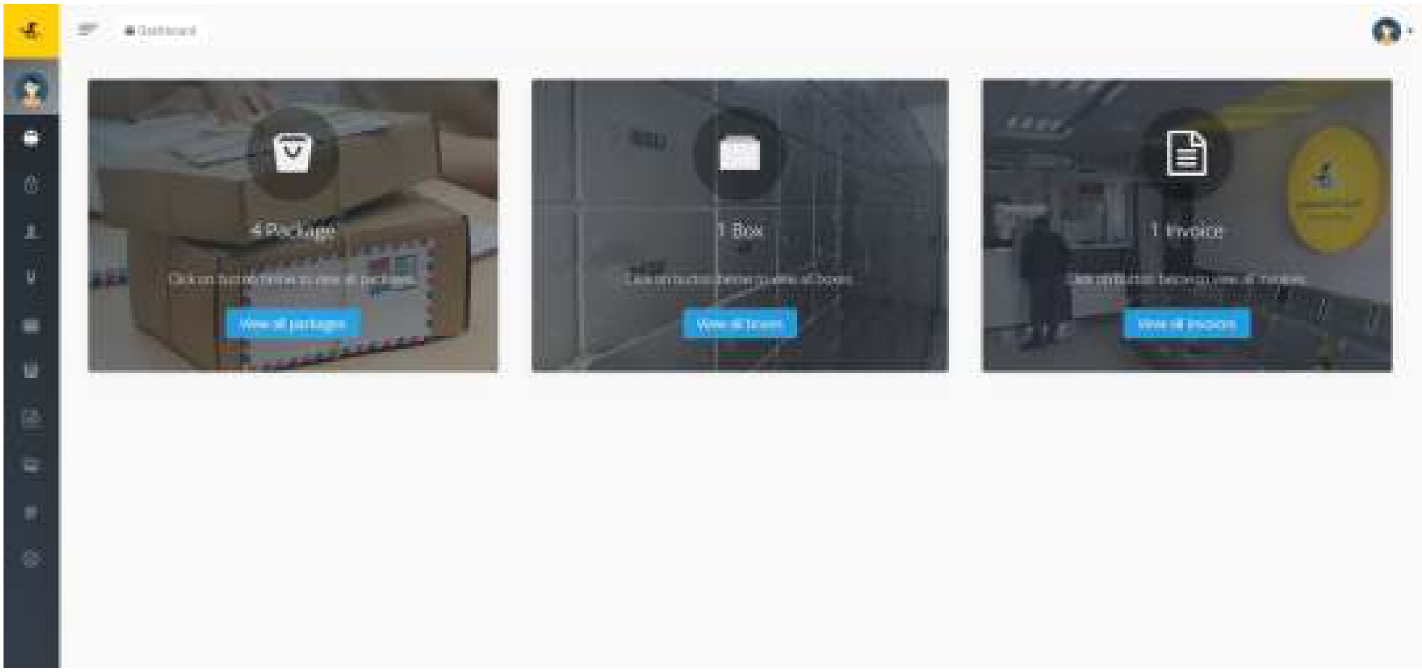


Figure 24: Routes

4.4. System interfaces (views):

Defines and manages how the data is presented to the user.

4.4.1. Interfaces for Palestine postal website:



4 Package

Click on button below to view all packages

1 Box

Click on button below to view all boxes

1 Invoice

Click on button below to view all invoices

4.4.2. Interfaces for Palestine postal application:

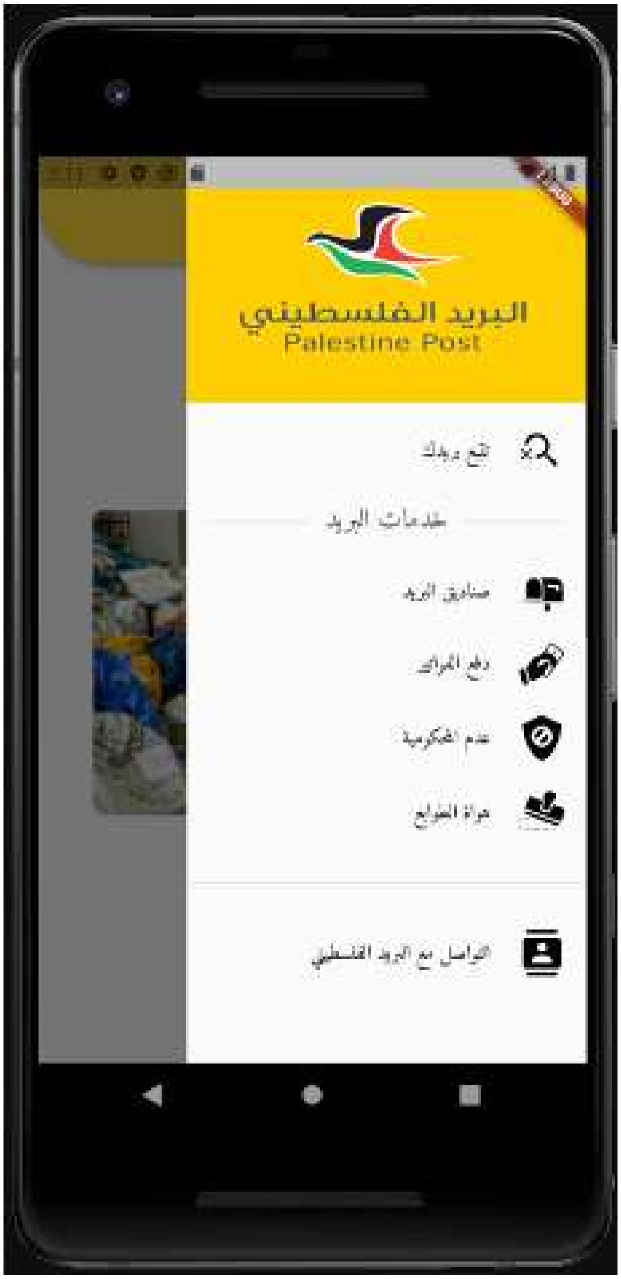


Figure 26 : Menu in App

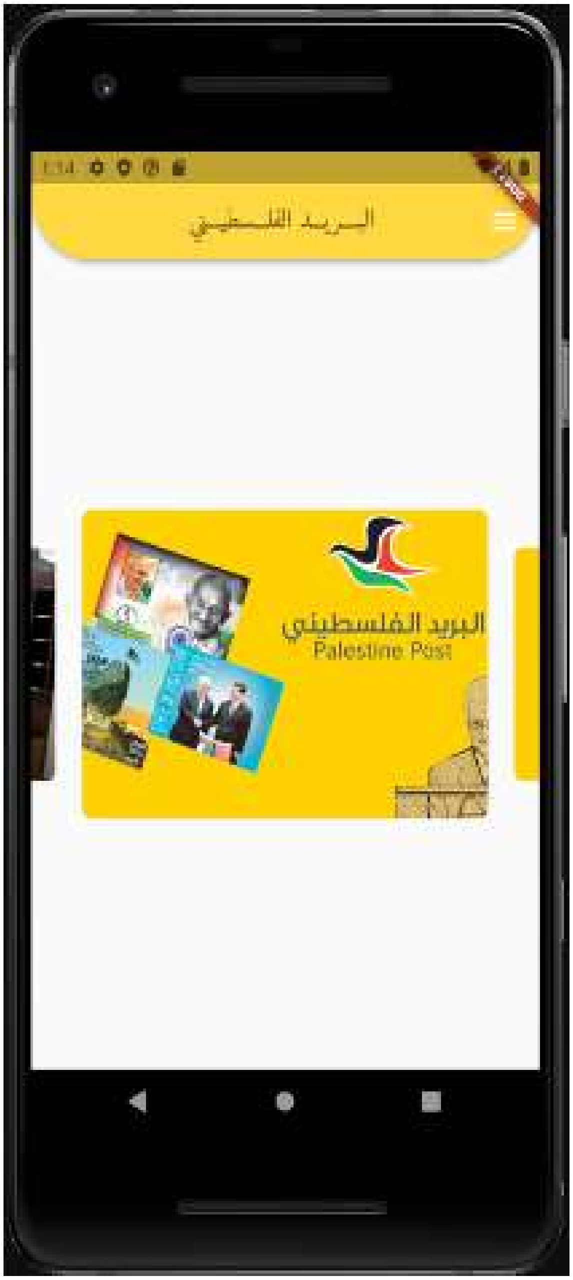


Figure 27 : Main Page

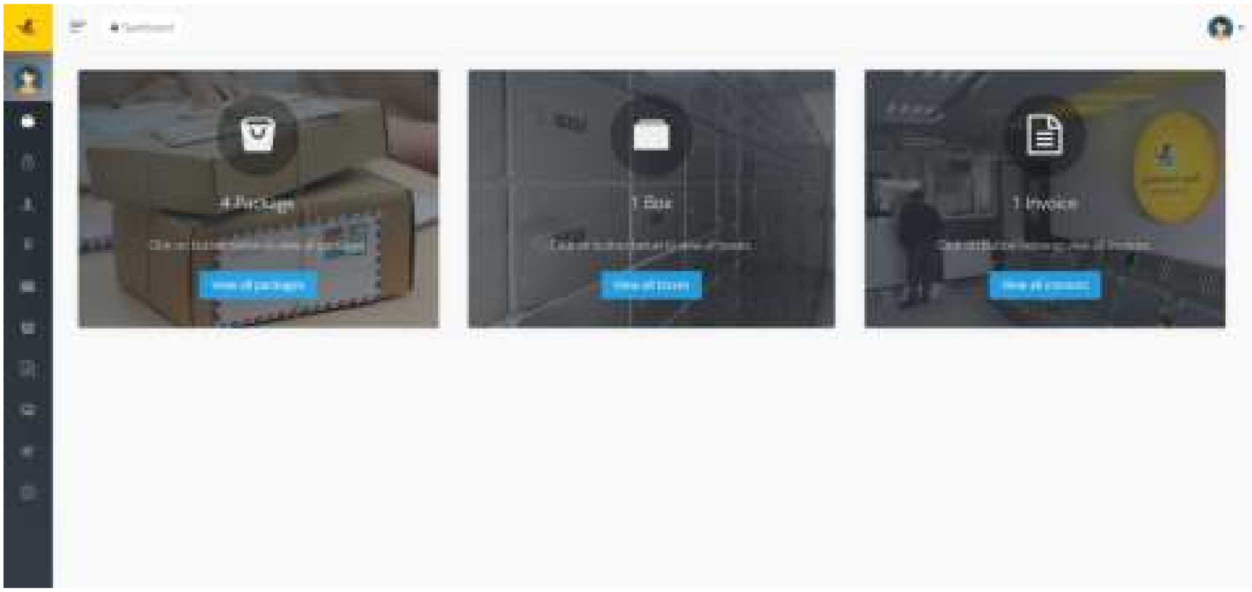
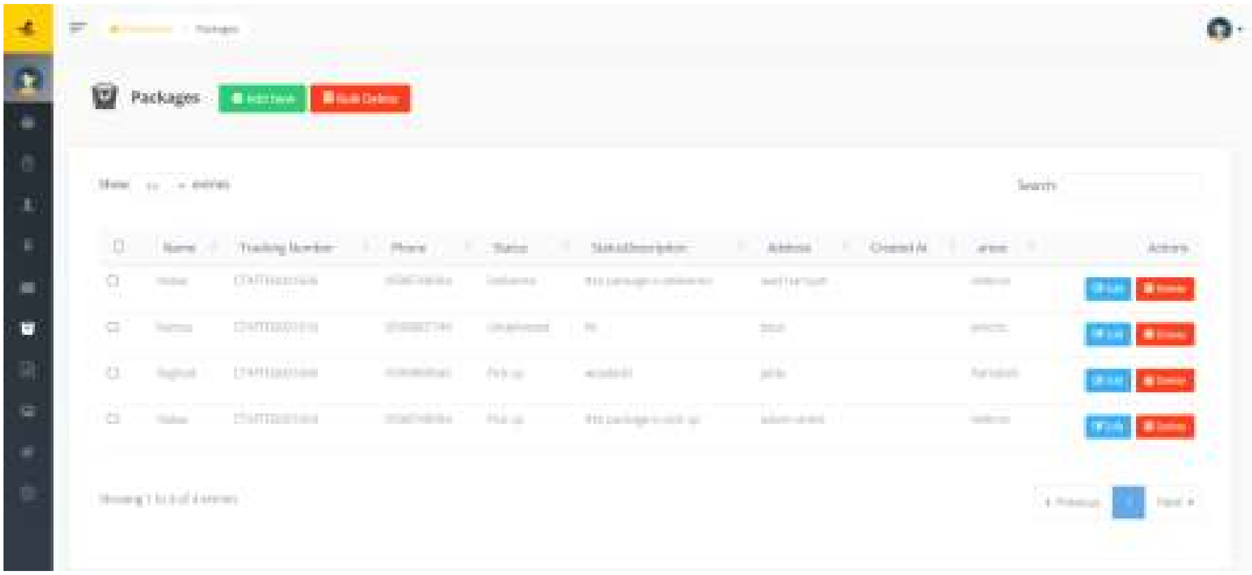


Figure 28 : Dashboard



□	Name	Tracking Number	Phone	Status	StatusDescription	Address	Created At	areas	Actions
□	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]	Edit Delete
□	[illegible]	[illegible]	[illegible]	Undelivered	[illegible]	[illegible]		[illegible]	Edit Delete
□	[illegible]	[illegible]	[illegible]	Pick up	[illegible]	[illegible]		[illegible]	Edit Delete
□	[illegible]	[illegible]	[illegible]	Pick up	[illegible]	[illegible]		[illegible]	Edit Delete

Showing 1 to 4 of 4 entries

Figure 29 : Packages Page

4.5. Operating the system

After completing all the stages of programming, it becomes possible to operate the system and prepare it for the process of testing.

Chapter 5: Testing

5.1. Introduction:

In this chapter, the processes of testing the system will be discussed, and some pictures showing the success of the operations will be included in addition to the explanation, description, and status of each.

5.2. Functional requirements testing

5.2.1. System check mechanism:

The system scan process incorporates mechanisms such as user entering wrong data and correct data to ensure that the system scan is accurate.
The system meets the requirements, and it will notify you if there are any errors.

5.2.2. Functional requirements testing:

In the following table are shown the processes that have been tested and the results of each process that include functional requirements for the site.

Table 12: functional testing

Test case	Scenario	Input data	Expected Output	Actual Output	Test result
Login	Users enter this email and password correctly.	Email:ImadTomaizy@email.com Password:12iMaD*	direct user to dashboard Successful	direct user to dashboard Successful	Pass
	Users enter email with the wrong password.	Email:ImadTomaizy@email.com Password:12iMaD1	Red warning message: If you have any issues please contact with your manager.	Red warning message: If you have any issues, please contact your manager.	Pass
Create user account	Admin click button add new in page users.	Add name , email , password , default role and image, click button save	success message: Successfully added new user	success message: Successfully added new user	Pass

	Admin click button add new	Add empty name Or wrong email: nidaaemail.com or wrong password: 1234m	Red warning message: The Name field is required. Warning: Please include an email “@” in the email address The Password must be at least 6 characters. The Password format is invalid.	Red warning message: The Name field is required. Warning: Please include an email “@” in the email address The Password must be at least 6 characters. The Password format is invalid.	Pass
Delete user account	Admin select user and click button delete in page users.	Delete user	Red warning message: Are you sure you want to delete this user?	Red warning message: Are you sure you want to delete this user?	Pass
Edit user account	Admin select user and click button edit in page users	Edit user name or email or password and image	success message: Successfully edited user	success message: Successfully edited user	Pass

		Add empty name Or wrong email: nidaaemail.com or wrong password: 1234m	Red warning message: The Name field is required. Warning: Please include an email “@” in the email address The Password must be at least 6 characters. The Password format is invalid.	Red warning message: The Name field is required. Warning: Please include an email “@” in the email address The Password must be at least 6 characters. The Password format is invalid.	Pass
Search user	Admin search user in search bar in page users	Add email or name in search bar.	Show the user matching the search results	Show the user matching the search results	Pass
Create new package	User and admin click button add new in page packages	Add name, tracking number, phone , status , status descriptions, address and areas correct, click button save	success message: Successfully added new package	success message: Successfully added new package	Pass
		Add wrong tracking number: cn789645323145678 Add wrong phone: 0566	Red warning message: The Tracking Number must not be greater than 14 characters. The Phone must be 10 digits.	Red warning message: The Tracking Number must not be greater than 14 characters. The Phone must be 10 digits.	Pass

Create new box	User and admin click button add new in page boxes	Add number, phone, user, area correct.	success message: Successfully added new boxes	success message: Successfully added new boxes	Pass
		Add wrong tracking number: -1, 1001	Red warning message: The Number must be between 0 and 1000.	Red warning message: The Number must be between 0 and 1000.	Pass

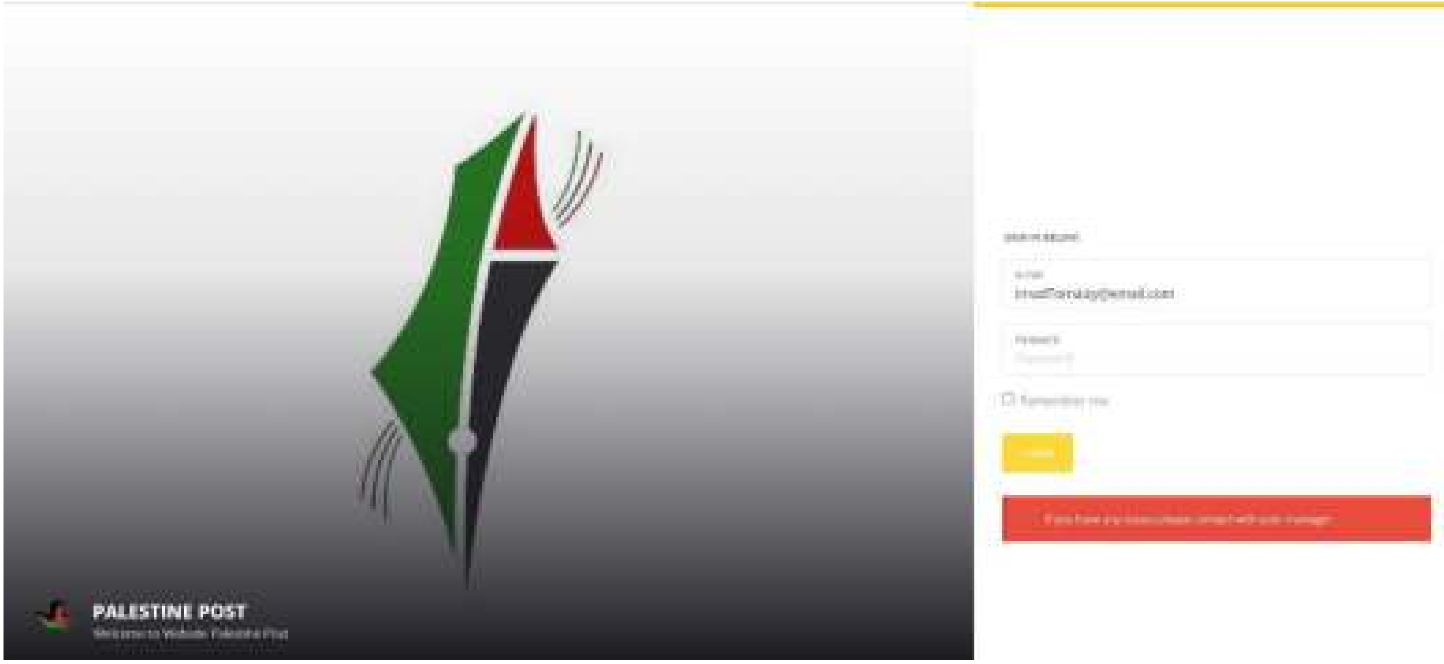
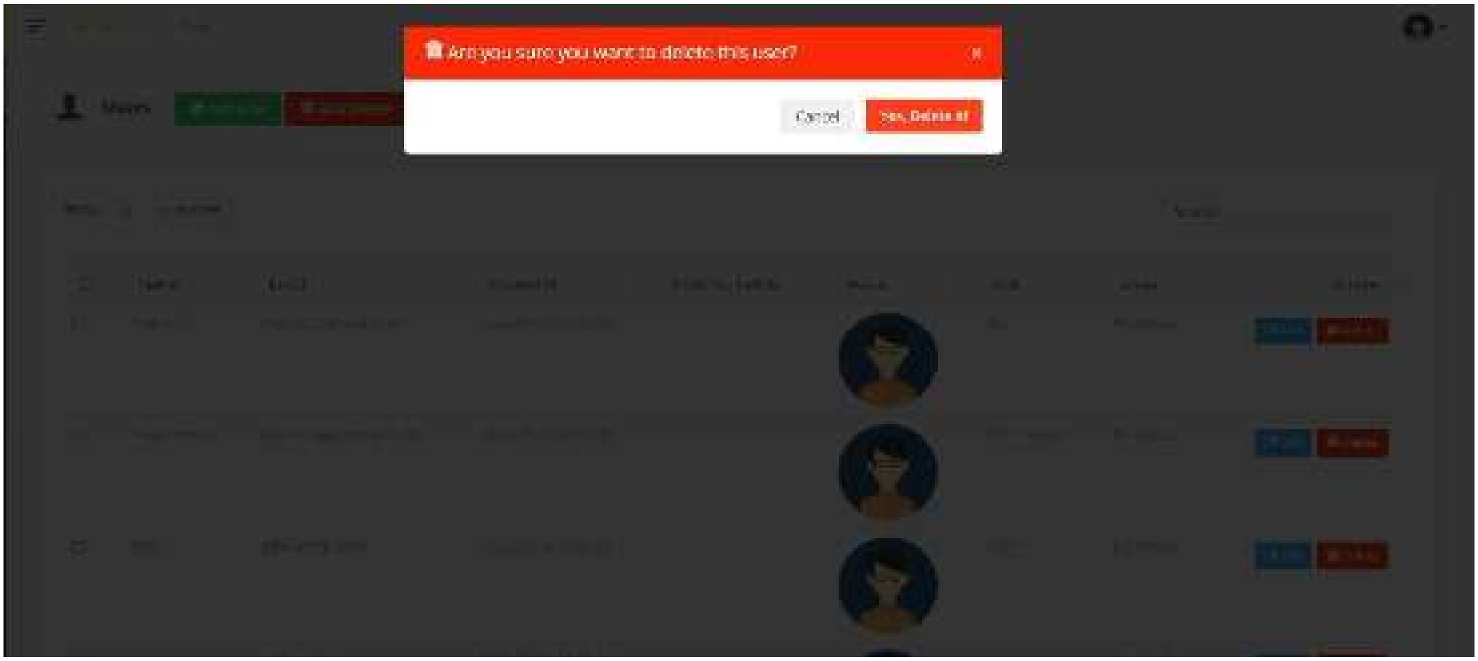
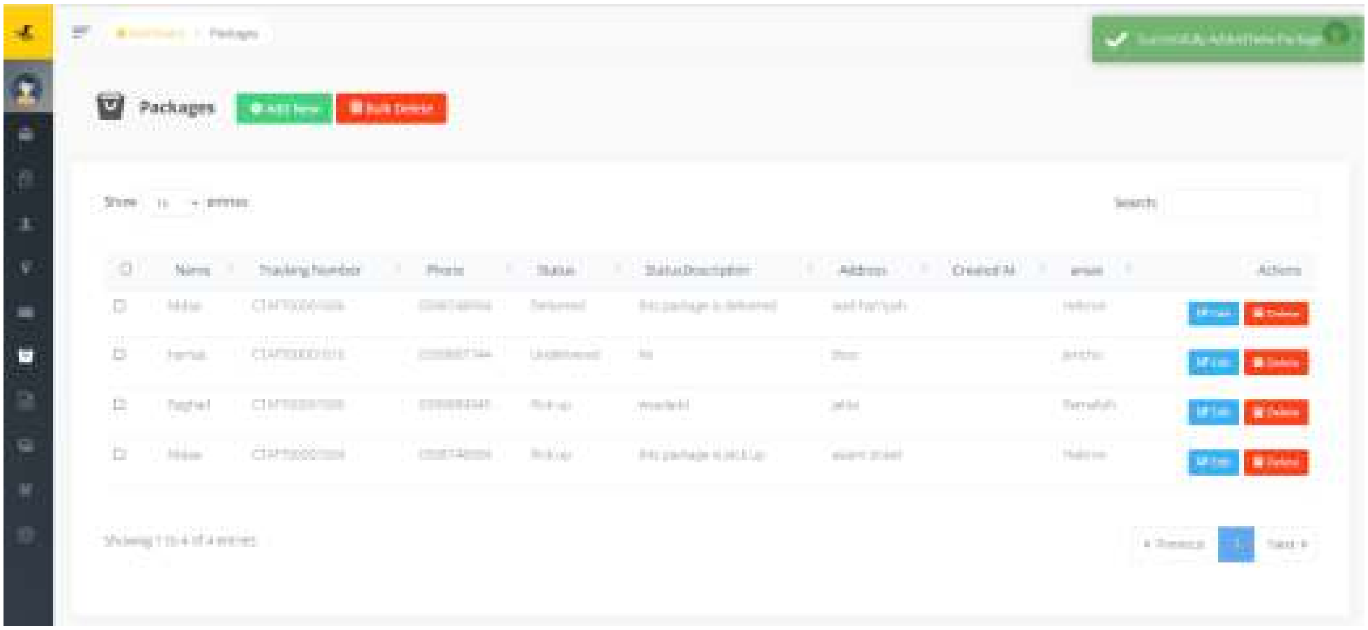


Figure SEQ Figure * ARABIC 30: Login Enter wrong password



Are you sure you want to delete this user?

Figure 31: Message When delete user



Successfully Added New Package

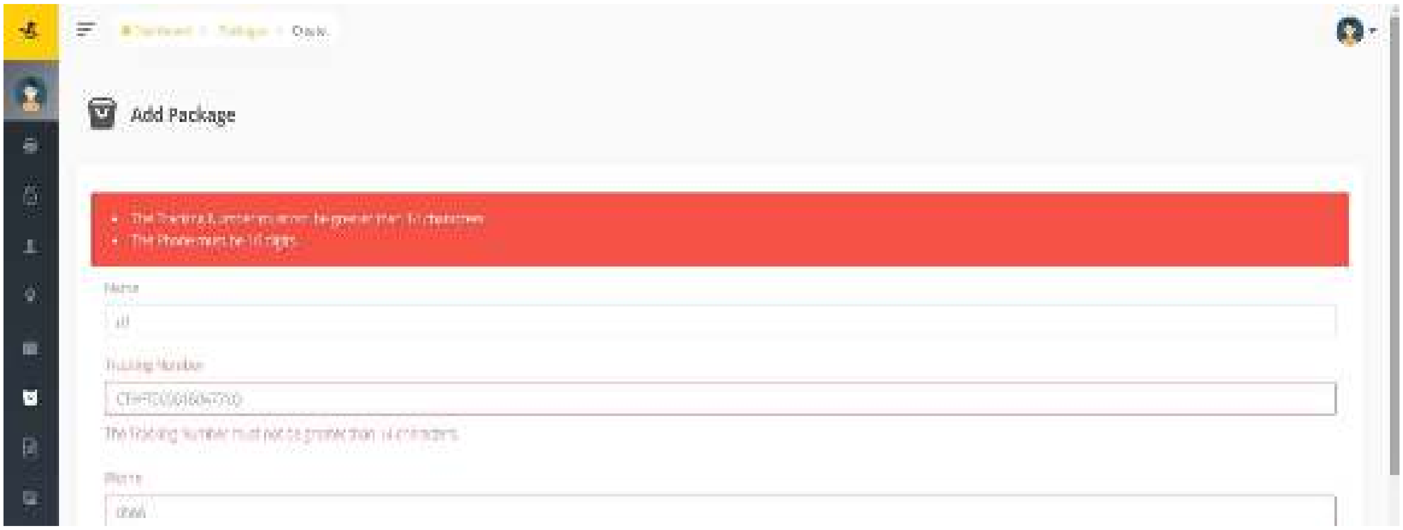
Packages

Show 10 entries

Search:

☐	Name	Tracking Number	Phone	Status	Status Description	Address	Created At	email	Actions
☐	[illegible]	[illegible]	[illegible]	Delivered	this package is delivered	[illegible]		[illegible]	Edit Delete
☐	hamza	[illegible]	[illegible]	Undelivered	[illegible]	[illegible]		[illegible]	Edit Delete
☐	[illegible]	[illegible]	[illegible]	Pick up	[illegible]	[illegible]		[illegible]	Edit Delete
☐	[illegible]	[illegible]	[illegible]	Pick up	this package is pick up	[illegible] street		[illegible]	Edit Delete

Showing 1 to 4 of 4 entries



Add Package

- The Tracking Number must not be greater than 14 characters.
- The Phone must be 11 digits.

Name

ali

Tracking Number

[illegible]

The Tracking Number must not be greater than 14 characters.

Phone

[illegible]

5.2.3. Unit testing for APIs

Unit testing is the most popular and important type of testing. A unit or component of a system is tested to determine if it works as expected. If no input is required, you can either call the component or provide an input and determine the output. When testing REST APIs, a unit is a single endpoint request, and how you write the unit test depends on what you want to test in the response.

We test our APIs using a postman testing tool.

1. GET: /api/packages
    - Expected output: json object contains first all packages on the website with json pager object.
    - Actual output:

```
[
  {
    "id": 4,
    "name": "Nidaa",
    "tracking_number": "CTAFT00001604",
    "phone": "0598748994",
    "status": "Pick up",
    "StatusDescription": "this package is pick up",
    "address": "aslam street",
    "area_id": 1,
    "created_at": null,
    "updated_at": null
  },
  {
    "id": 5,
    "name": "Raghad",
    "tracking_number": "CTAFT00001606",
    "phone": "0599889945",
    "status": "Pick up",
    "StatusDescription": "wsadadd",
    "address": "jalda",
    "area_id": 3,
    "created_at": null,
    "updated_at": null
  },
  {
    "id": 6,
    "name": "hamza",
    "tracking_number": "CTAFT00001616",
    "phone": "0599887744",
    "status": "Undelivered",
    "StatusDescription": "hii",
    "address": "btoo",
    "area_id": 2,
    "created_at": null,
    "updated_at": null
  },
  {
    "id": 8,
    "name": "Nidaa",
    "tracking_number": "CTAFT00001606",
    "phone": "0598748994",
    "status": "Delivered",
    "StatusDescription": "this package is delivered",
    "address": "wad harriyah",
    "area_id": 1,
    "created_at": null,
    "updated_at": null
  }
```

```
]
```

- Passed/failed: Passed

2. GET: /api/search/package
   - Input: /api/search/CTAFT00001616
   - Expected output: details about the package tracking_number = CTAFT00001616
   - Actual output:

```
[
  {
    "id": 6,
    "name": "hamza",
    "tracking_number": "CTAFT00001616",
    "phone": "0599887744",
    "status": "Undelivered",
    "StatusDescription": "hii",
    "address": "btoo",
    "area_id": 2,
    "created_at": null,
    "updated_at": null
  }
]
```

   - Passed/failed: Passed

Chapter 6: Conclusion and Future Work.

In the end, we created a website belonging to the Palestinian Post and an application for mobile phones, where the accounts for postal employees are created by the admin to carry out their work electronically. Postal boxes, delete and modify, send a message over the phone to the citizen, create an invoice, print it, modify or delete it, and add the postal areas by the admin.
We also wanted to serve citizens in obtaining information about their package that do not comply with the standard of international package by searching for the tracking number of their package by adding in the application search box, inquiring about postal services, and communicating with the Palestinian Post.

In the future, a mobile application can be developed to add login and notification access.

After confirming the readiness of the system and its strength in meeting the operational requirements and ensuring the readiness of the site, it will be put on the local market as an initial step to take advantage of it and develop it in advanced stages to include new varieties and other ideas.

Reference

- Laravel:

1. https://laravel.com/docs/9.x.
2. https://voyager-docs.devdojo.com/.

- flutter:

3. https://www.fluttericon.com/.
4. https://dart.dev/.
5. https://docs.flutter.dev/.

- stack overflow:

6. https://stackoverflow.com/.