Software Requirements Specification

Project: Packaging Management System

Document file

Date	Revision	Author	Verified quality dep.
09/19/2024	0.2	Herrera Luevano Jesus Arturo Monarrez Barron Polo Alejandro Suastegui Leyva Yerlan Axel Toledo Herrera Neyzer Joel	

Document validated by the parts on date: 09/19/2024

By the client:	By the supplier company:
Technological University of Tijuana	DrimDev
Signed by	Signed by

Content

1 Introduction **3**
1.1 Purpose 4
1.2 Scope 4
1.3 People Involved 4
1.4 Glossary 5
1.5 References 5
1.6 Overview 5
2 General Description **6**
2.1 Product Perspective 6
2.2 Product Functions 6
2.3 User Characteristics 7
2.4 Constraints 7
2.5 Assumptions and Dependencies 7
2.6 Evolution of the System 7
3 Specific Requirements **7**
3.1 Common Interface Requirements 8
3.1.1 User Interfaces 8
3.1.2 Hardware Interfaces 8
3.2 Functional Requirements 9
3.2.1 Functional requirement 1 9
3.2.2 Functional requirement 2 9
3.2.3 Functional requirement 3 10
3.2.4 Functional requirement 4 11
3.2.5 Functional requirement 5 11
3.3 Non-Functional Requirements 11
3.3.1 Performance Requirements 11
3.3.2 Security 12
3.3.3 Reliability 12
3.3.4 Availability 12
3.3.5 Maintainability 12
3.3.6 Portability 13
4 Appendix **13**
4.1 Entity-Relationship Diagram 13
4.2 Relational Model 13
4.3 Use Case 13
4.4 Activities 13
4.5 Sequence 14
4.6 Communication 14
4.7 Deployment 14
4.8 Components 14

1 Introduction

The following content describes the specification of functional and non-functional requirements of the Packaging Management System. This software will be developed to optimize a company device packaging process, guaranteeing that the process is carried out in an efficient and standardized manner, allowing storage and generation of reports. The Software Requirements Specification will serve as a reference for the development team and ensure that all interested parties understand the capabilities and limitations of the system.

1.1 Purpose

The purpose of this document is to describe exhaustively and clearly the functional and non-functional requirements for the development of the Packaging Management System. This document will serve as a technical reference for all phases of software development, from design to implementation and testing, ensuring that the final product meets the expectations and needs of stakeholders.

- **Development Team:** To use as a technical guide during the design, development and implementation of the software.
- **Clients and Interested Parties:** To ensure that system requirements are aligned with business needs and applicable regulations.

1.2 Scope

The software to be developed will be called Packaging Management System (SGE). This system aims to manage materials, verify the packaging process and traceability of packages, allowing companies to maintain control over how their products are packaged so they can be stored and later shipped.

1.3 People Involved

Name	Toledo Herrera Neyzer Joel
Role	Team Leader, Developer
Professional Category	TSU in IT Area Multiplataform Software Development
Responsibilities	Documentation, team organization, networking, frontend and backend programming, data base, documentation
Contact Information	0323105969@ut-tijuana.edu.mx
Approval	

Name	Herrera Luevano Jesus Arturo
Role	Backend Leader
Professional Category	TSU in IT Area Multiplataform Software Development
Responsibilities	Backend structure, frontend programming, database, documentation
Contact Information	0323105941@ut-tijuana.edu.mx
Approval	

Name	Monarrez Barron Polo Alejandro

Role	Frontend Leader
Professional Category	TSU in IT Area Multiplataform Software Development
Responsibilities	Frontend structure, backend programming, database, documentation
Contact Information	0323106434@ut-tijuana.edu.mx
Approval	

Name	Suastegui Leyva Yerlan Axel
Role	Database Leader
Professional Category	Database structure, frontend and backend programming, documentation
Responsibilities	Documentation, team organization
Contact Information	0322103828@ut-tijuana.edu.mx
Approval	

1.4 Glossary

DB: Database, an information structure.
FN: Functional Requirement
NFR: Non-Functional Requirement
DBMS: Database Management System
PMS: Packaging Management System

1.5 References

Reference	Title	Rout	Date	Author
01	IEEE Recommended Practice for Software Requirements Specifications, Std. 830-1998		1998	IEEE

1.6 Overview

This document consists of three sections. In the first section, a introduction to it and provides an overview of the specification of system resources.

In the second section of the document a general description of the system is made, In order to know the main functions that it must perform, the data associated and the factors, constraints, assumptions and dependencies that affect the development, without going into excessive details.

Finally, the third section of the document is the one in which they define in detail the requirements that the system must satisfy.

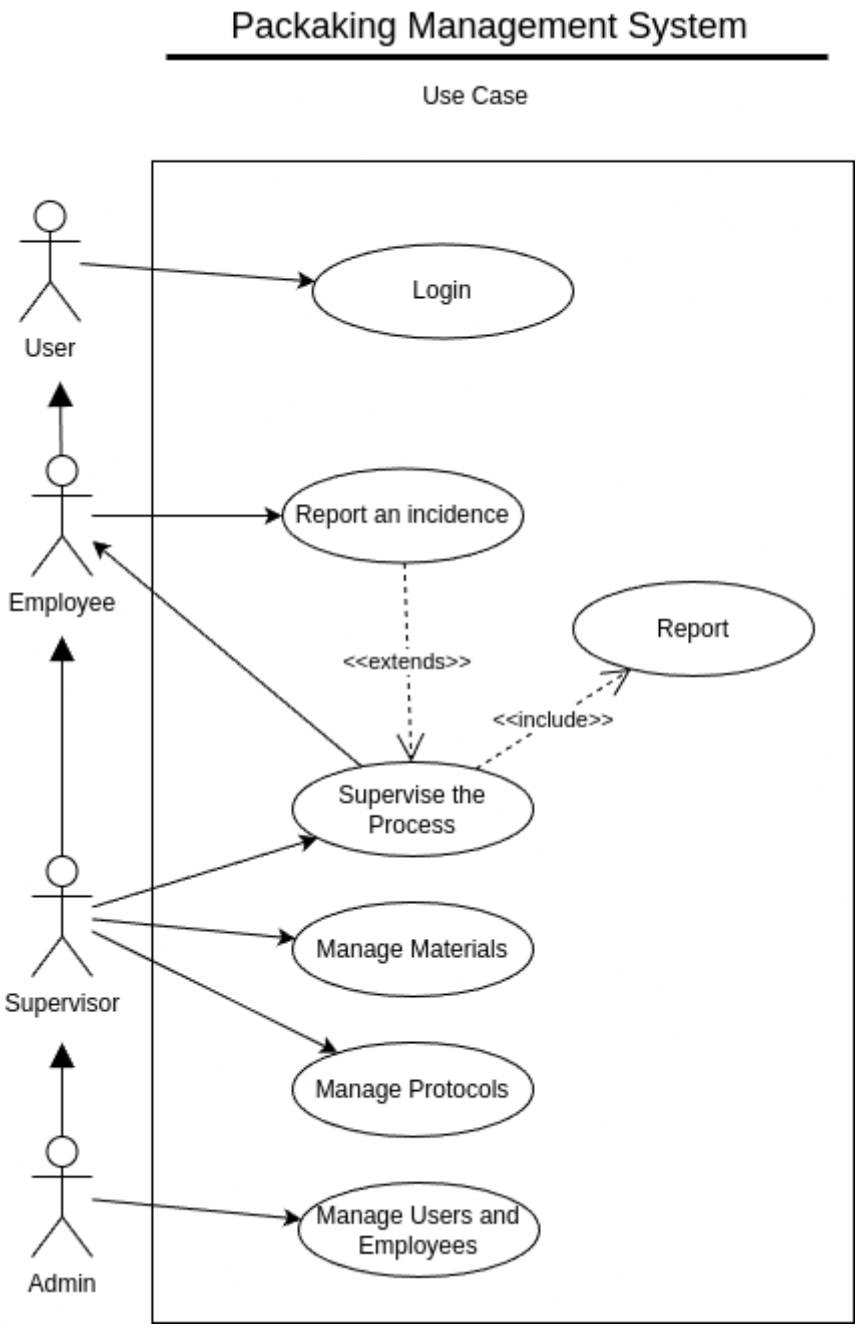
2 General Description

2.1 Product Perspective

The (PMS) is not a stand-alone product; It is part of a larger system, specifically as a module within the secondary and tertiary packaging system used by the company to manage its supply chain and generate reports

2.2 Product Functions

;



2.3 User Characteristics

User Type:	Employee
Training:	Basic packaging knowledge
Skills:	
Activities:	Make reports, monitor the process

User Type:	Supervisor
Training:	Advanced knowledge of packaging and basic processes for the system.
Skills:	
Activities:	Management the protocols, material, management traceability

User Type:	Administrator
Training:	Advanced knowledge of packaging processes and the company
Skills:	Excel, Database, Computer
Activities:	Management Employees

2.4 Constraints

- **Language and technologies in use:** HTML, CSS, JavaScript y PHP.
- **Database:** MySQL.
- **Operative System Server:** Linux.

2.5 Assumptions and Dependencies

The computers on which the system will be run must meet the requirements before indicated to guarantee its correct execution.

2.6 Evolution of the System

● Improve graphical interfaces.

● Develop the desktop application to cover the most influential sectors today (pc, mobile).

3 Specific Requirements

Requirement number	RF01
Requirement name	Packaging Type Management
Type	☐ Requirement ☐ Constraint
Source requirement	The platform allows you to register and manage different types of packaging, from boxes of different sizes to specific protective materials for each type of product.
Priority requirement	☐ High/Issentiall ☐ Medium/Desired ☐ Low/ Optional

Requirement number	RF02
Requirement name	Manage the Packaging Process
Type	☐ Requirement ☐ Constraint
Source requirement	Based on the type of product, the system automatically assigns the necessary protocols and materials to ensure adequate protection during transportation.
Priority requirement	☐ High/Issentiall ☐ Medium/Desired ☐ Low/ Optional

Requirement number	RF03
Requirement name	Package Registration and Traceability
Type	☐ Requirement ☐ Constraint
Source requirement	Each package is tagged with a unique code that allows you to track its status, verify who packed it, what materials were used, and when it was packaged.
Priority requirement	☐ High/Issentiall ☐ Medium/Desired ☐ Low/ Optional

Requirement number	RF04
Requirement name	Storage
Type	☐ Requirement ☐ Constraint
Source requirement	There will be control of the warehouse of all the packaging produced, taking care of its measurements for subsequent transportation.
Priority requirement	☐ High/Issentiall ☐ Medium/Desired ☐ Low/ Optional

Requirement number	RF05
Requirement name	Report Generation
Type	☐ Requirement ☐ Constraint
Source requirement	The platform generates detailed reports on the efficiency of the packaging process, showing data such as packaging time per product, material use and any recorded incidents.
Priority requirement	☐ High/Issentiall ☐ Medium/Desired ☐ Low/ Optional

3.1 Common Interface Requirements

3.1.1 User Interfaces

In this section we will see how the employee interacts with the software in a way that must be intuitive and allow users to perform their tasks without difficulty, they must be able to perform their tasks quickly and efficiently and it must be easy to remember after a period of time. use.

3.1.2 Hardware Interfaces

General Linux Server Configuration

- Input/Output Configuration:
  - Use /dev/input to access input devices.
  - Use udev to manage device configuration when connecting them.

- User Interface:
  - If using a desktop environment, configure keyboard and mouse preferences through system configuration options.
- Security and Permissions:
  - Ensure that the user has the appropriate permissions to access and configure input devices.

3.2 Functional Requirements

3.2.1 Functional requirement 1

Packaging Type Management

It allows you to manage different types of packaging, including important details, such as:

- **Length units**
  - Cm
- **Units of mass**
  - Kg
- **Primary Packaging Information**
  - Packaging Description
  - Packaging shape (rectangular)
  - Packaging measurements (length, width and height)
  - Total product weight (kg)
- **Box External Dimension Information**
  - Box properties
  - How many packages fit in a box?
  - Box restrictions (length, width and height)
  - Corrugated thickness (length, width and height)
- **Box standard**
  - Clearance in the box (length, width and height): Space between the products inside the box
- **Pallet Dimensions**
  - Length, width and height
  - Total pallet weight (kg)

3.2.2 Functional requirement 2

Manage the Packaging Process

The secondary and tertiary packaging process is essential
to guarantee protection and efficiency in the distribution of products.

Secondary Packaging Process

i. **Material Selection:** Boxes are chosen that are robust enough to contain several individual products (primary packaging).

ii. **Product Grouping:** Individually packaged devices (for example, a cell phone in its box) are grouped together. This may include multiple units of the same model or a mix of different models, depending on demand.

iii. **Packing:** Products are placed in secondary boxes, ensuring there is enough cushioning material (such as bubble wrap or kraft paper) to prevent movement during transportation.

iv. **Labeled:** Each secondary box is labeled with relevant information, such as the model number, the number of products inside, handling instructions and barcodes.

Information Required During the Process

- **Product Details:** Model names, serial number and specifications.
- **Handling Instructions:** Precautions and conditions for transportation.
- **Logistics Information:** Shipping addresses, tracking numbers and expiration dates if applicable.

Tertiary Packaging Process

1. **Container Selection:** Pallets, large boxes or containers that can hold several secondary boxes are chosen.
2. **Grouping of secondary boxes:** Secondary boxes are organized on pallets or in containers to maximize space and ease of handling.
3. **Assurance:** Strapping or plastic wrap is used to secure boxes on pallets and prevent them from sliding or falling.
4. **Labeled:** Each grouping is labeled with important logistics information, including destination, contents and quantities.

Storage Process in Tertiary Packaging

- **Space Optimization:** Pallets are organized efficiently in the warehouse, using stacking techniques and taking advantage of vertical space.
- **Inventory Management:** Records are kept of the contents of each pallet or container to facilitate inventory tracking and management.
- **Storage Conditions:** Adequate conditions are ensured to preserve the quality of electronic products.

3.2.3 Functional requirement 3

Package Registration and Traceability

Each package must receive a unique code that allows it to be tracked, including information about who packed it, the materials used, and when it was packaged.

Information Required During the Process

- **Product Details:** Model names, serial number and specifications.
- **Handling Instructions:** Precautions and conditions for transportation.
- **Logistics Efficiency:** It helps optimize the supply chain by facilitating transportation and distribution in large volumes.

3.2.4 Functional requirement 4

Storage

Control the inventory of all types of packaging and pallets in the warehouse, ensuring that measurements and storage conditions are recorded. In this case, the **Packaging Type Management** would enter into relationship with the warehouse.

The packaging will be stored in a specific location, information such as ID, name, area, total capacity, available capacity and date will be obtained.

It should be noted that before reaching the warehouse it goes through a certain process, where it already comes with certain characteristics and information, which is why not as much information is required at this storage point.

3.2.5 Functional requirement 5

Report Generation

The system will generate detailed reports showing:

- Packaging time by product type.
- Use of materials in the packaging process.
- Any incident recorded during packaging.

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

Number of Simultaneous Users

- **Supervisors and Administrators:** The system must allow a certain number of users to access the system.

Transactions

- **Packaging record:** The system must manage a record of packages and updating their status, ensuring that each product movement is reflected.
- **Report generation:** You should be able to generate reports every time you finish a packaging process.

Data Storage Capacity

- **Product database:** The system database must be able to store a large amount of information on different products, including their packaging specifications.
- **Packaging history:** The system must be able to maintain a packaging history, ensuring that this information is available for traceability.

Security and Access

- **Authentication and permissions:** The system must manage user authentication with a verification time and allow role management to control access to different modules.

3.3.2 Security

Database Security Improvements

The use of views in the database will be made, in order to protect and improve security against data vulnerability.

3.3.3 Reliability

Incidents Allowed per Year

Specification: The system must allow a maximum of 5 significant incidents per year. A significant incident is defined as a failure that affects the operation of the system for more than 30 minutes.

Objective: Reduce the number of significant incidents to 3 per year through proactive maintenance measures and continuous system monitoring.

3.3.4 Availability

Availability During Business Hours

The system must offer maximum availability to employees during working hours. This ensures that they can access the system at all times during their shifts.

3.3.5 Maintainability

Preventive Maintenance

This type of maintenance seeks to anticipate and prevent system failures. It includes regular software updates, database cleaning and process optimization.

3.3.6 Portability

Using a Compiler or Development Platform

- **Specification:** The system must use a compiler or development environment that supports multiple platforms.
- **Aim:** Facilitate development and deployment on different operating systems, allowing developers to work in the environment they prefer.

Use of an Operating System

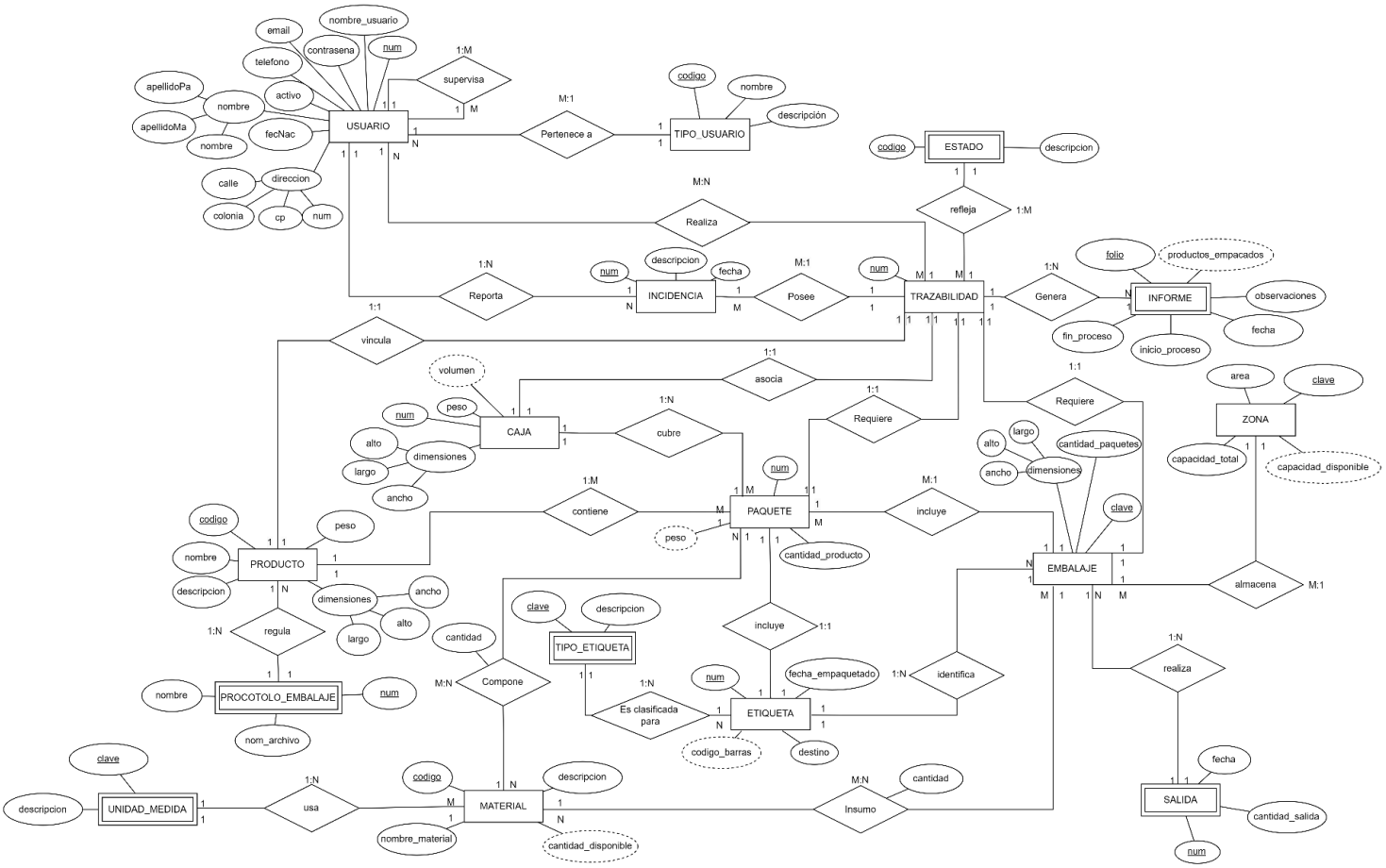
- **Specification:** The software must be compatible with at least two major operating systems, such as Windows and Linux. It must be ensured that software versions can run without problems in these environments.
- **Aim:** Expand system accessibility, allowing users to choose the environment that best suits their operational needs.

Portability Documentation

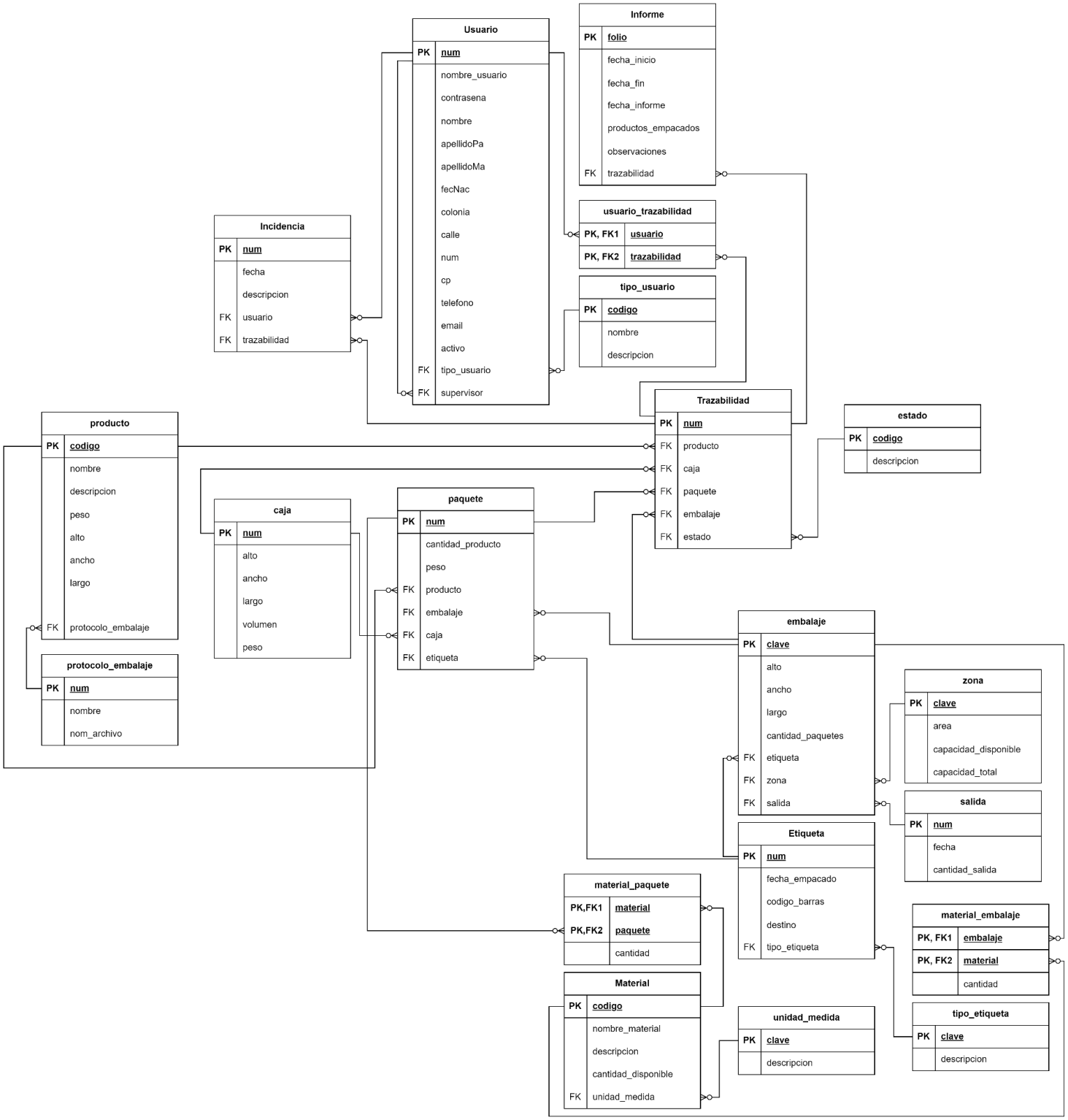
- **Specification:** Clear documentation on requirements must be provided.
- **Aim:** Facilitate system portability through detailed instructions.

4 Appendix

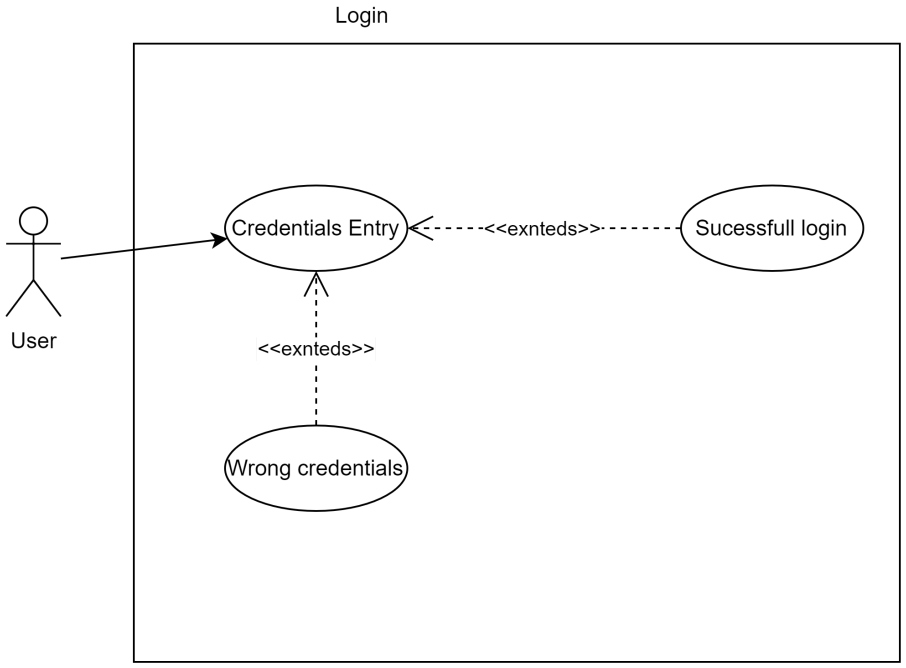
4.1 Entity-Relationship Diagram

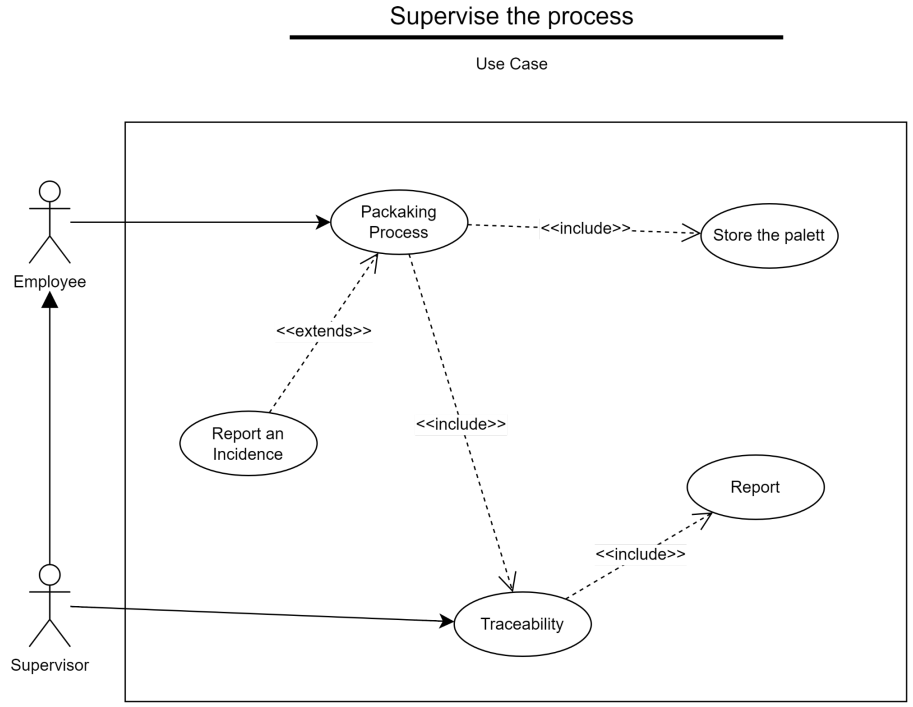


4.2 Relational Model

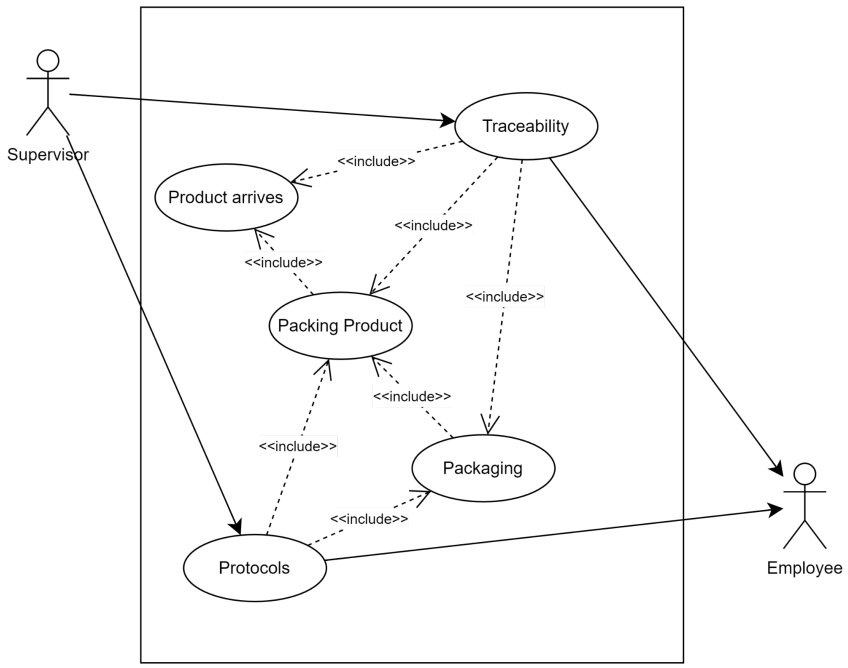


4.3 Use Case

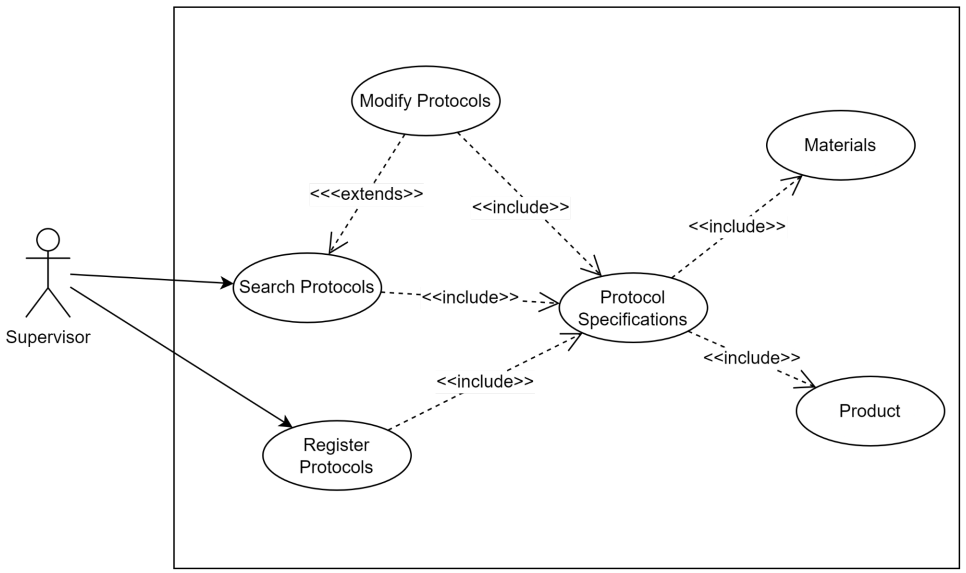

Login
Credentials Entry
<<exnteds>>
Sucessfull login
User
<<exnteds>>
Wrong credentials


Supervise the process
Use Case
Employee
Packaking Process
<<include>>
Store the palett
<<extends>>
Report an Incidence
<<include>>
Report
<<include>>
Traceability
Supervisor

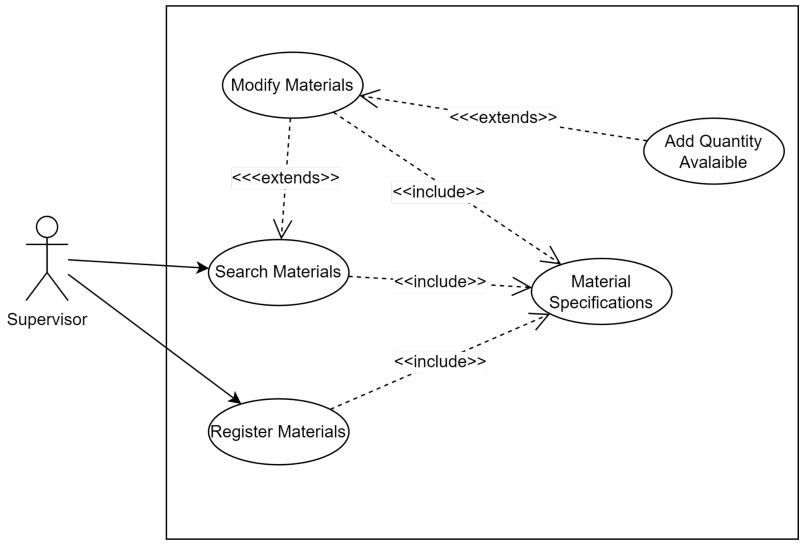
Packaking Process



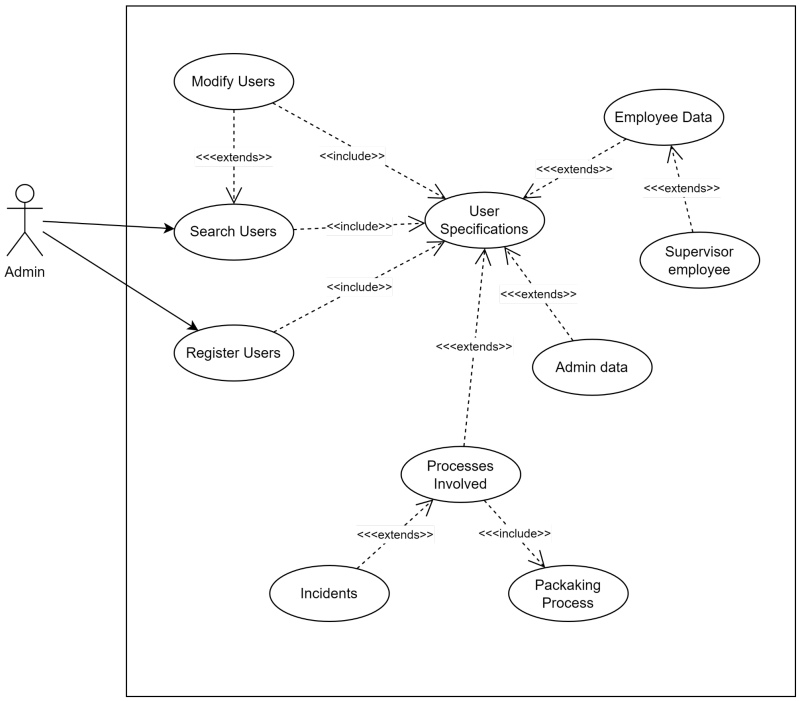
Manage Protocols



Manage Materials

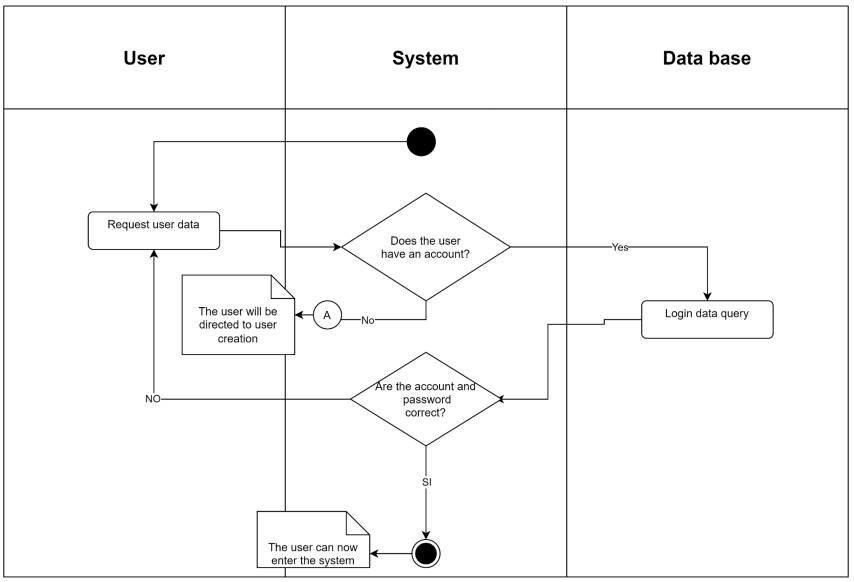


Manage Users

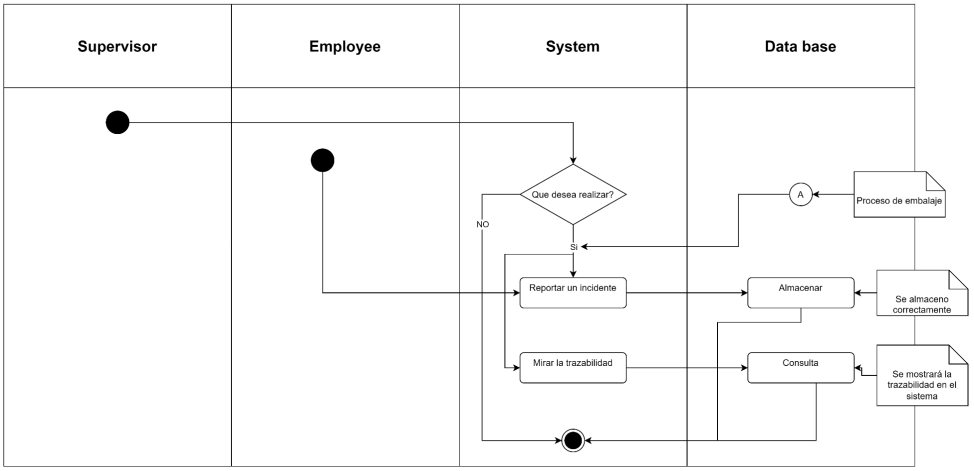


4.4 Activities

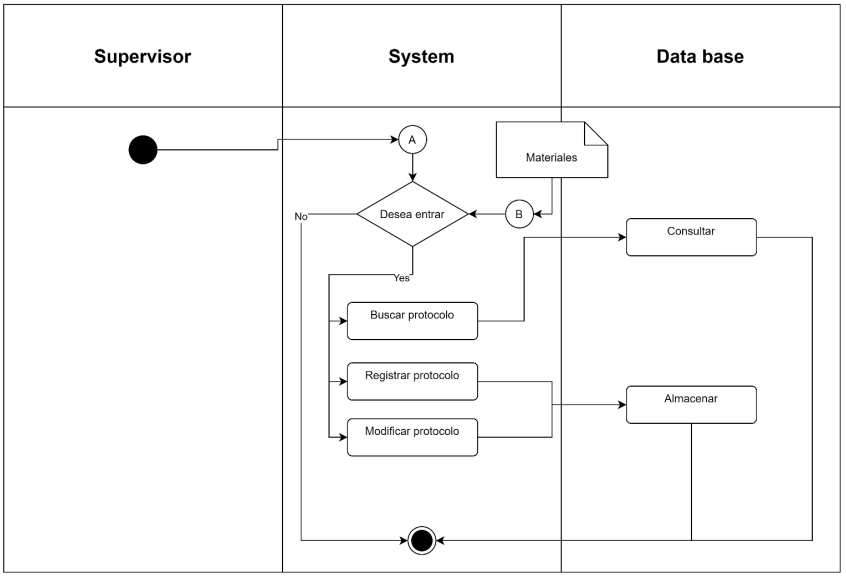
Login



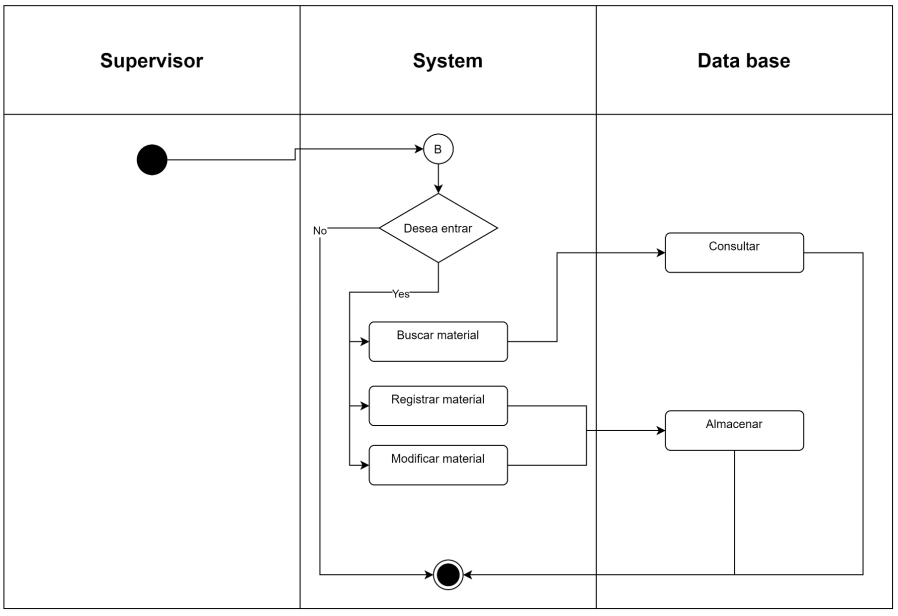
Supervise



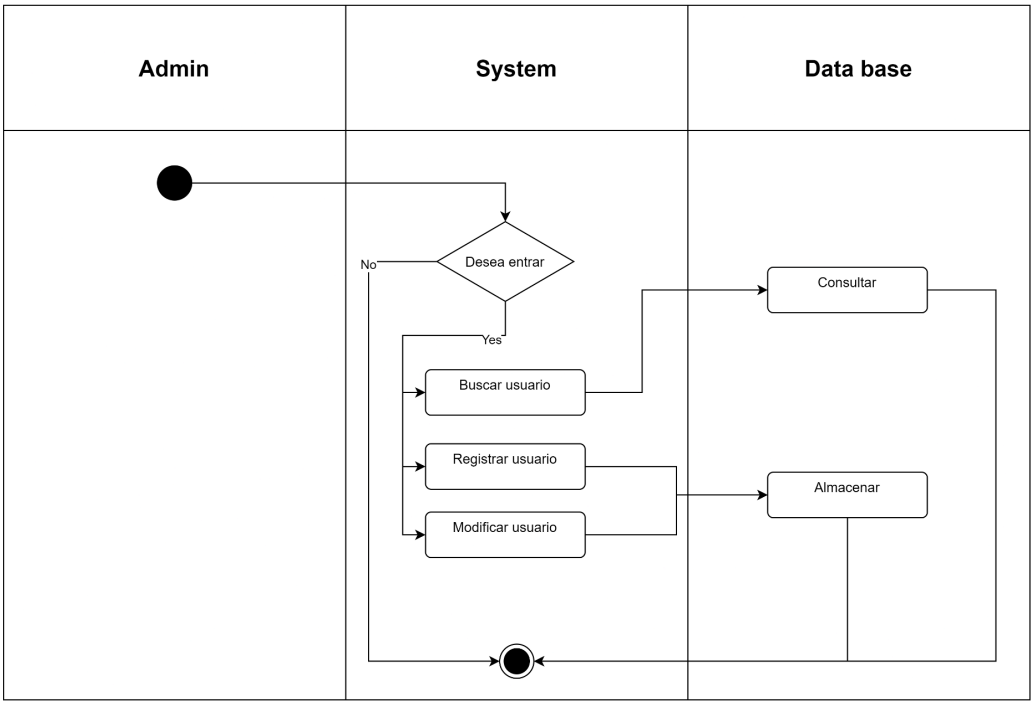
Protocols



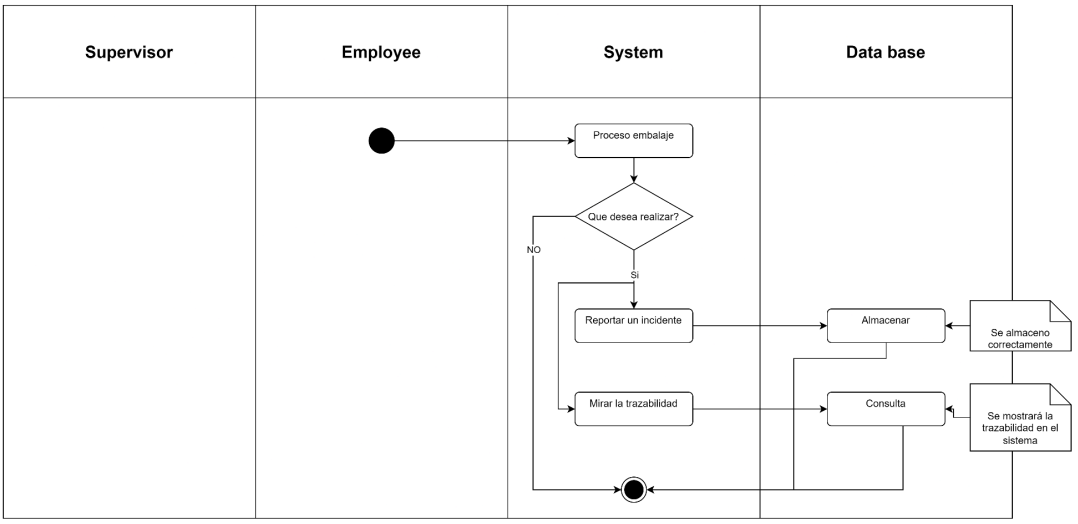
Materials



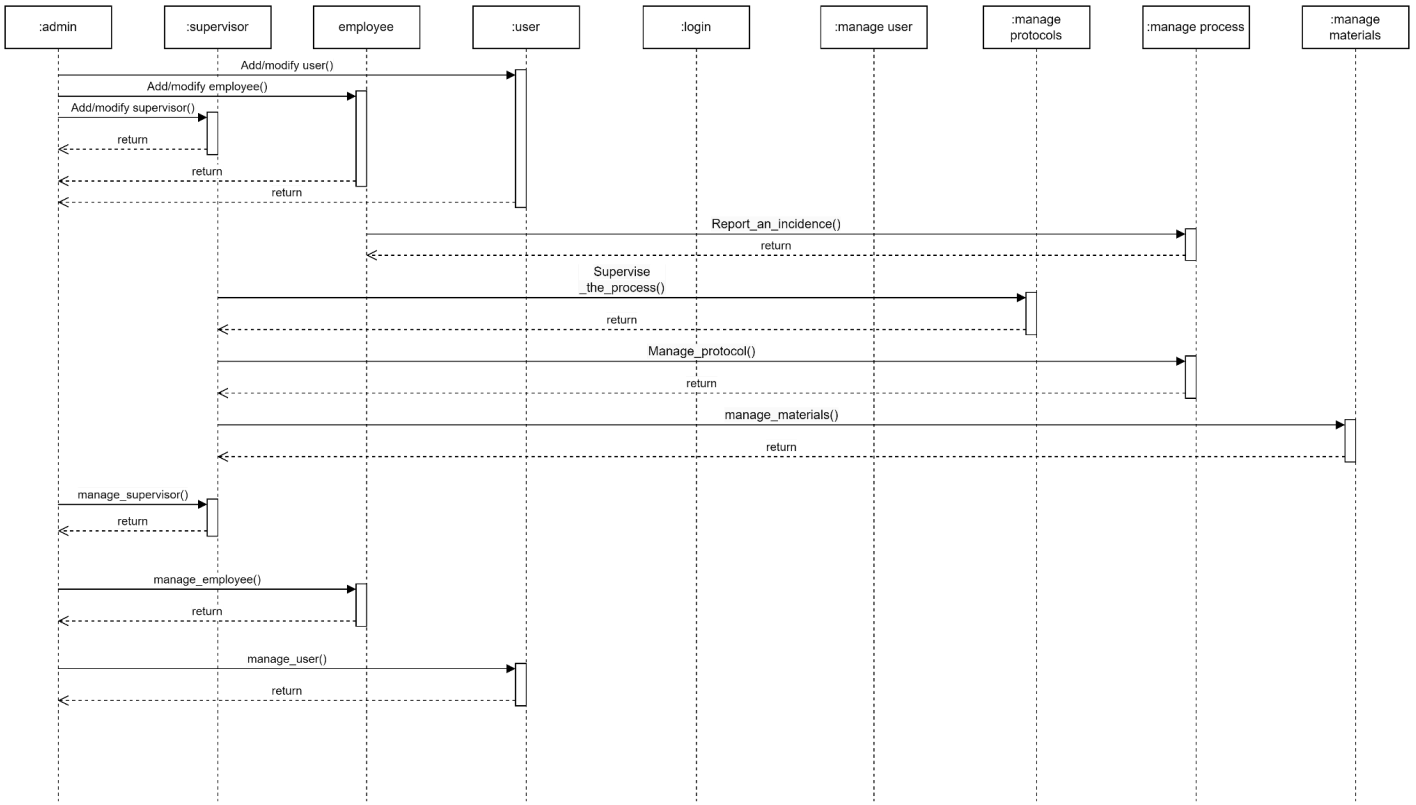
Users



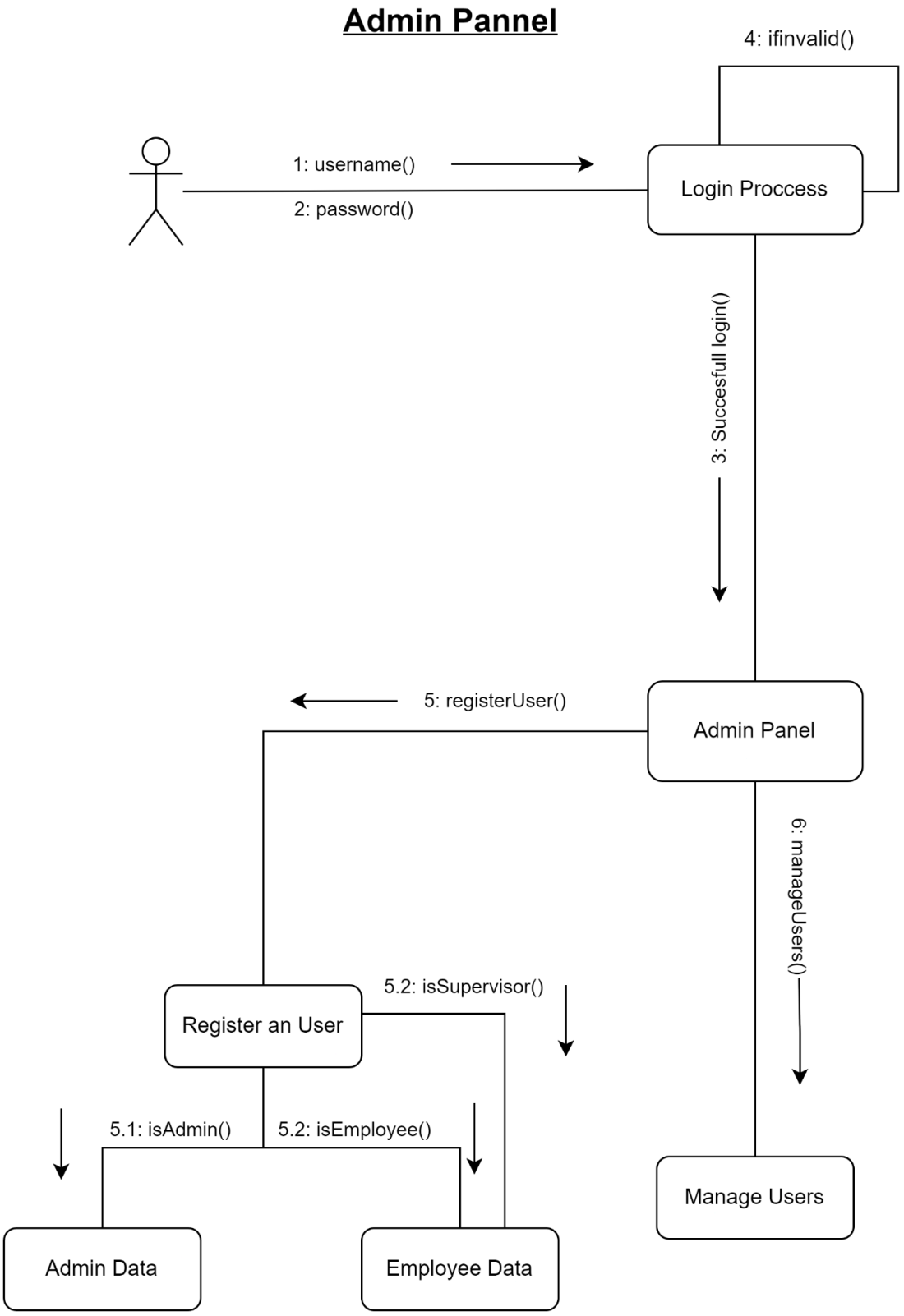
PROCESS

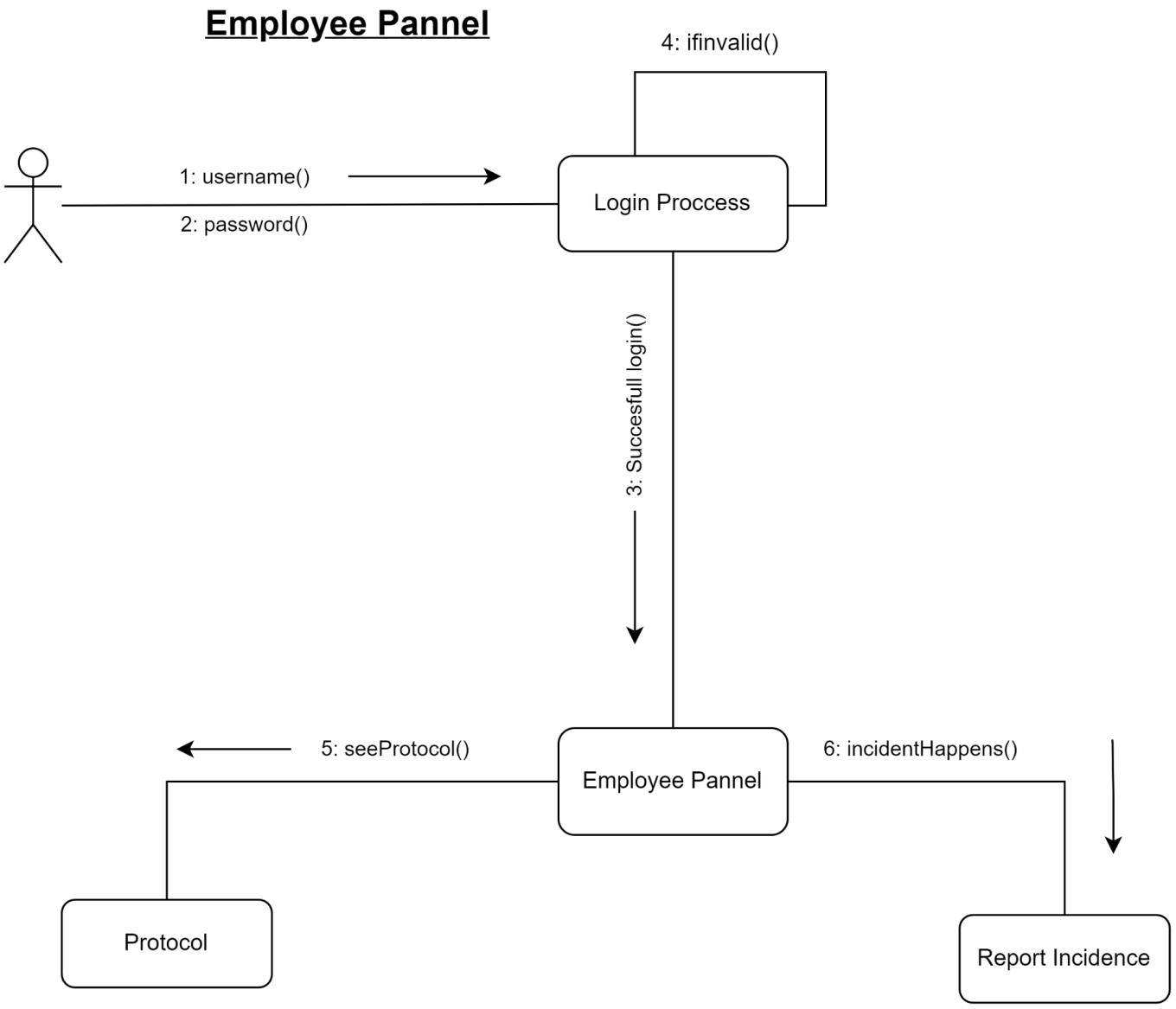


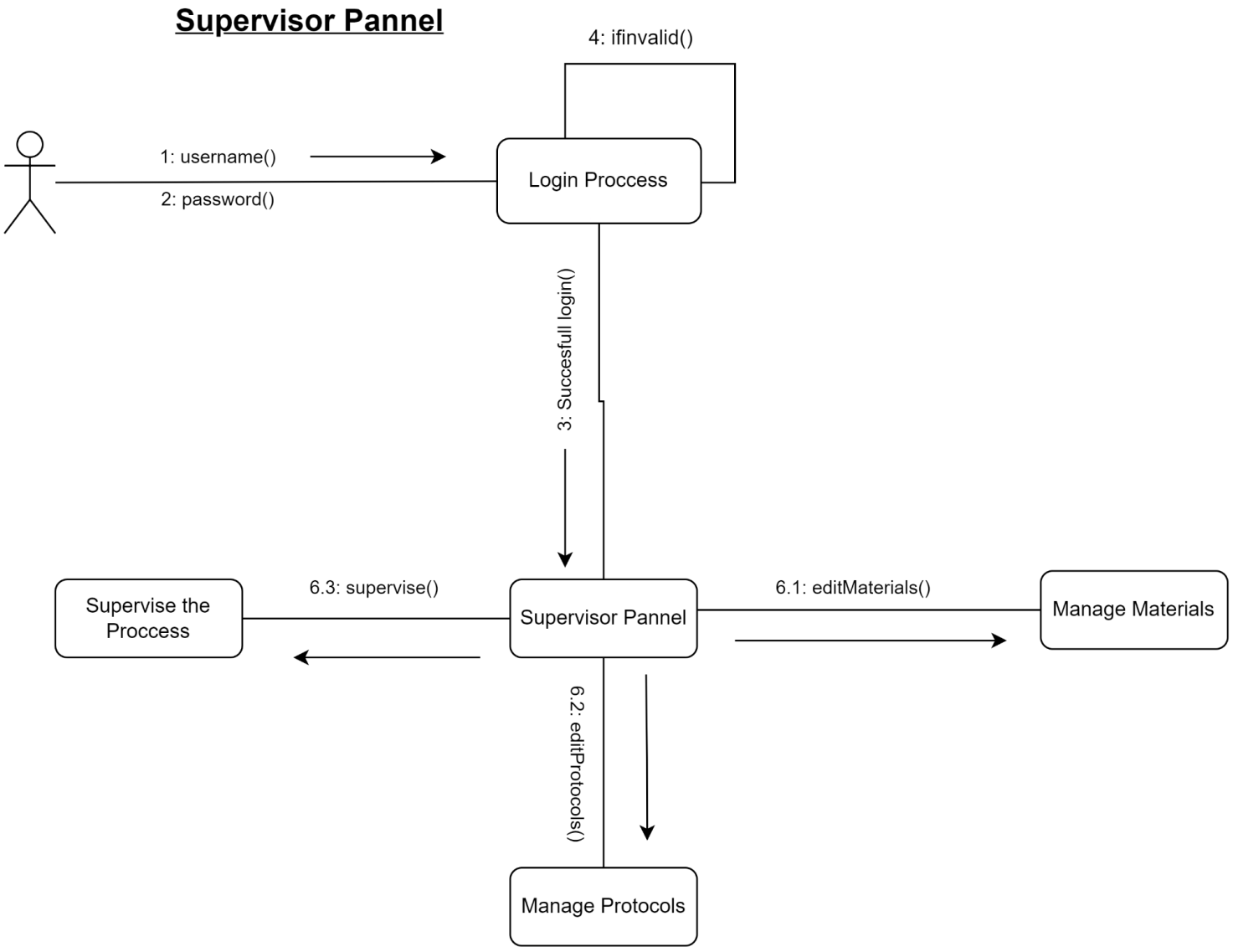
4.5 Sequence


:admin
:supervisor
employee
:user
:login
:manage user
:manage protocols
:manage process
:manage materials
Add/modify user()
Add/modify employee()
Add/modify supervisor()
return
return
return
Report_an_incidence()
return
Supervise _the_process()
return
Manage_protocol()
return
manage_materials()
return
manage_supervisor()
return
manage_employee()
return
manage_user()
return

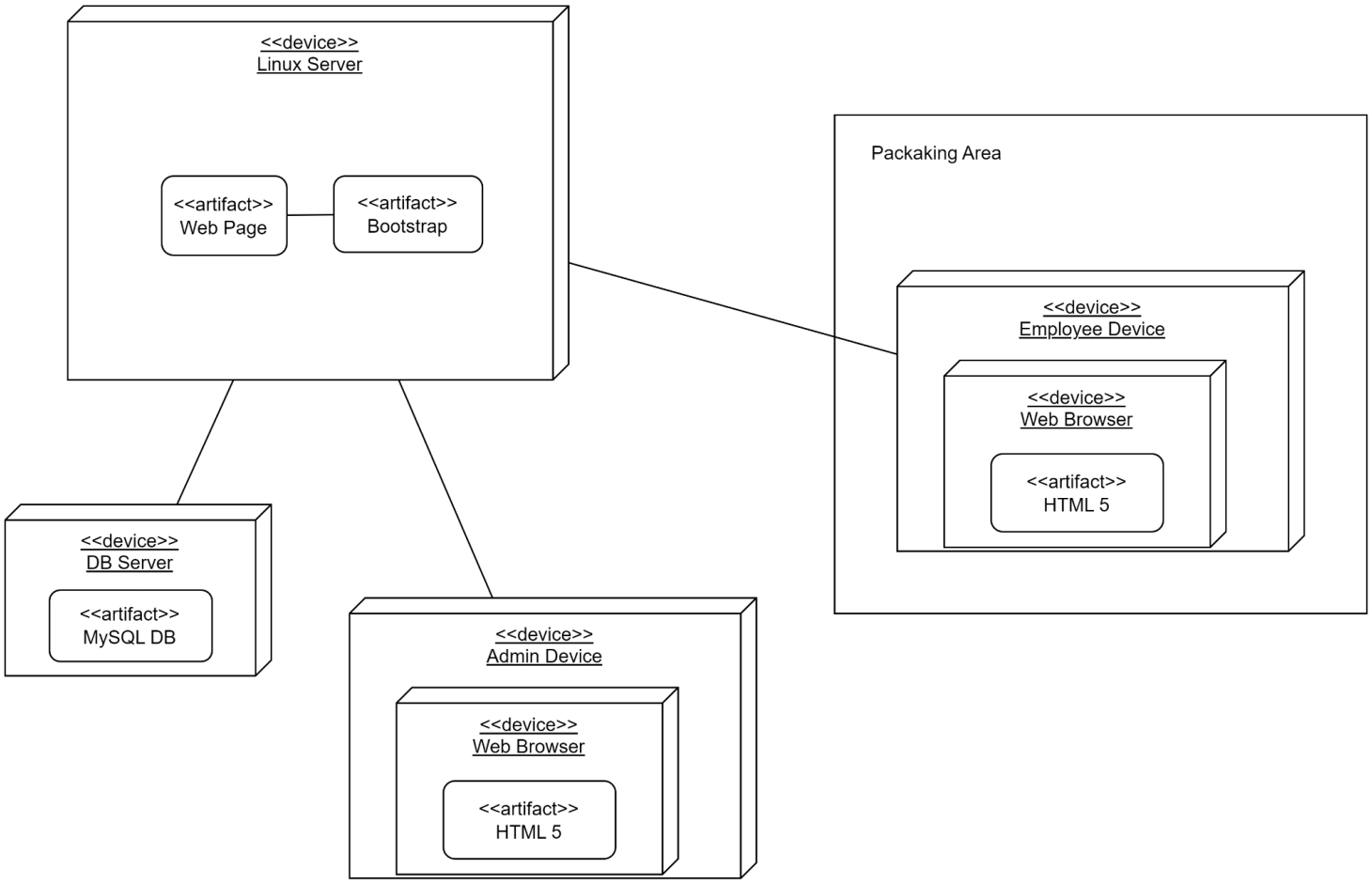
4.6 Communication




Employee Pannel
4: ifinvalid()
1: username()
2: password()
Login Proccess
3: Succesfull login()
5: seeProtocol()
Employee Pannel
6: incidentHappens()
Protocol
Report Incidence

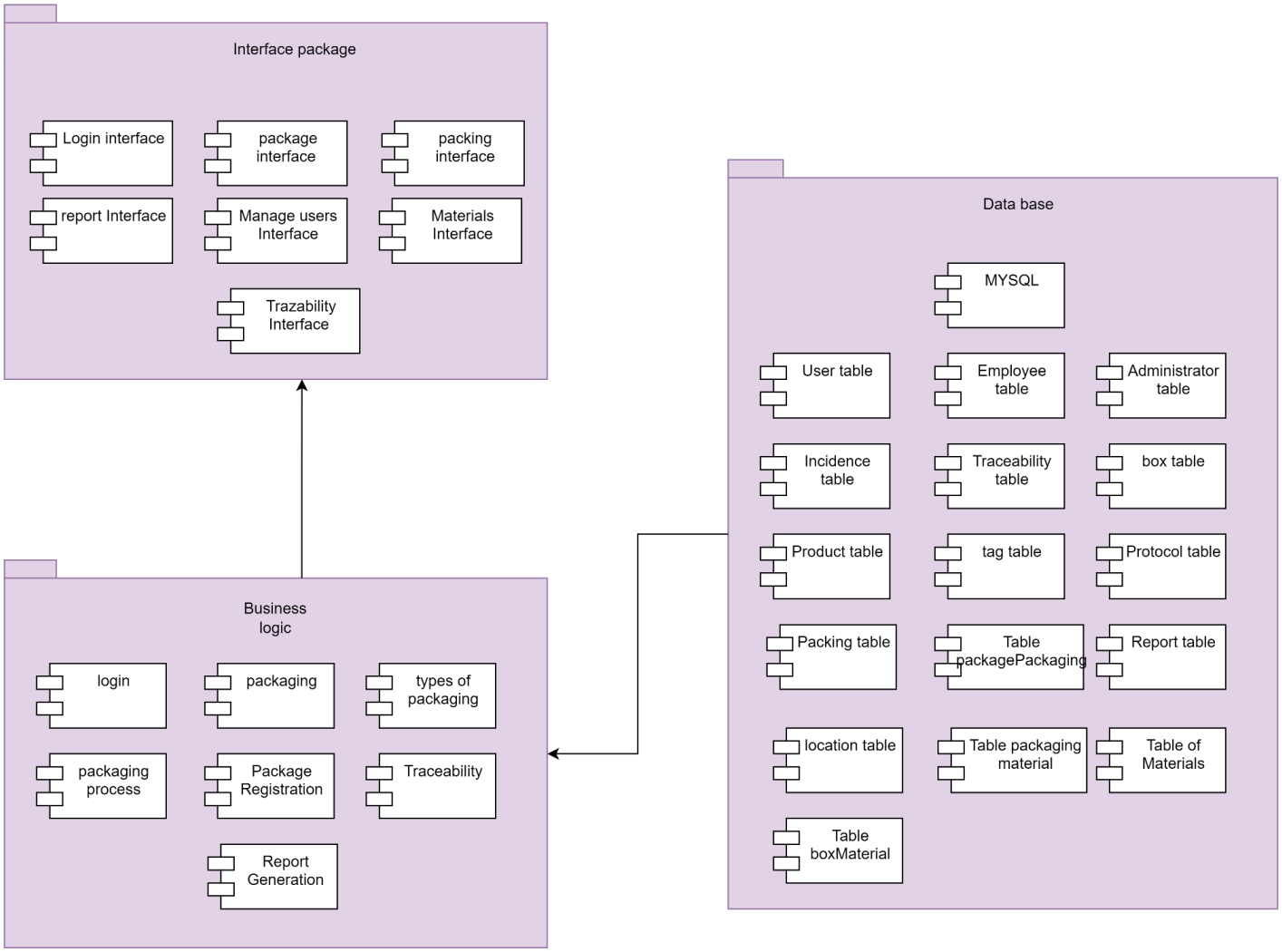

Supervisor Pannel
4: ifinvalid()
1: username()
2: password()
Login Proccess
3: Succesfull login()
Supervise the Proccess
6.3: supervise()
Supervisor Pannel
6.1: editMaterials()
Manage Materials
6.2: editProtocols()
Manage Protocols

4.7 Deployment



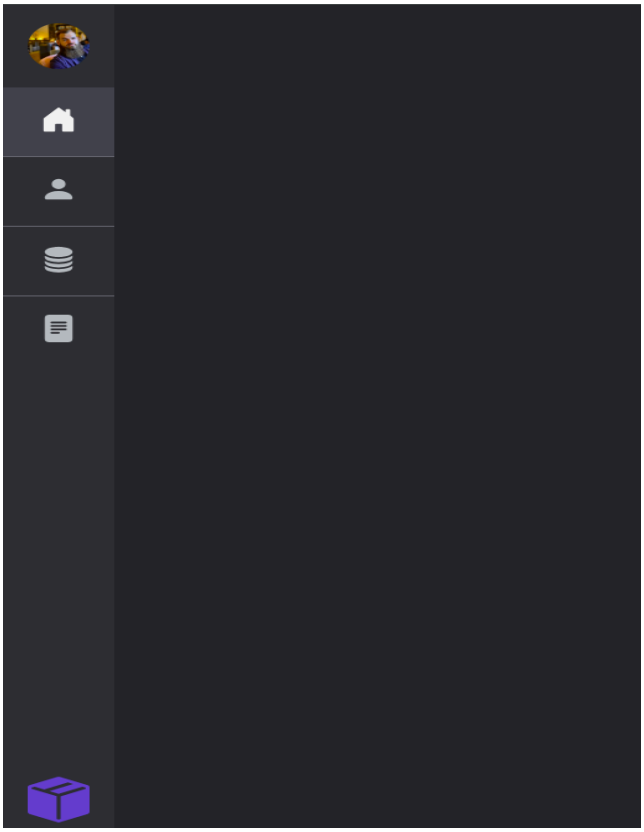
4.8 Components

Component diagram



4.9 Screens

Index

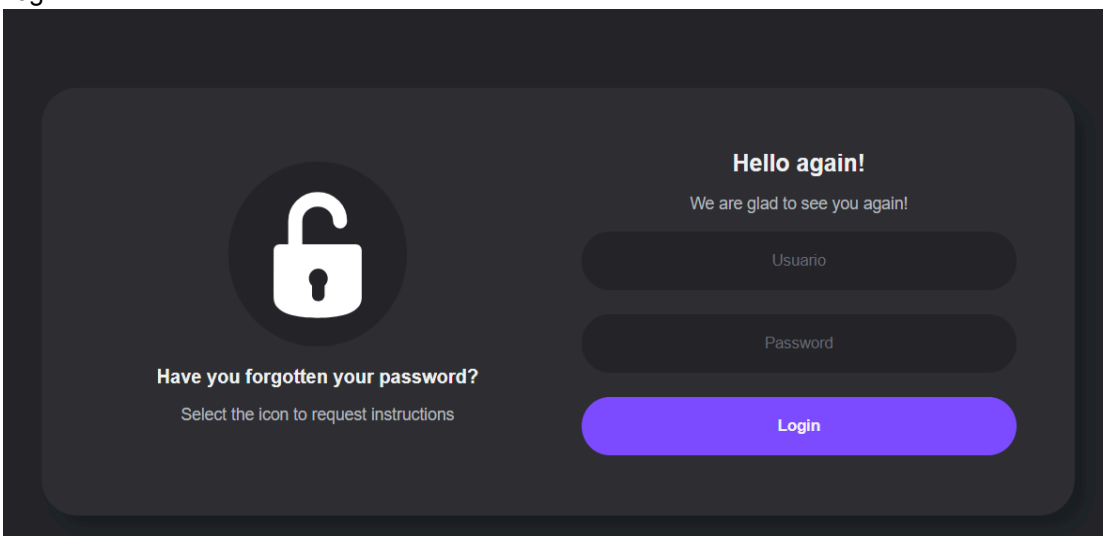


This screen is going to be our home and is responsible for encompassing and redirecting the others.

```php
<?php

function connectdb(){

    try {
        $db = mysqli_connect("localhost", "root", "", "packaging");
        //echo "<p>Conectado<p>";
        return $db;
    } catch (Exception $e) {
        echo "<p>Conection Error: {$e->getMessage()}<p>";
        return false;
    }
}
```

This function is responsible for making the connection to the database.

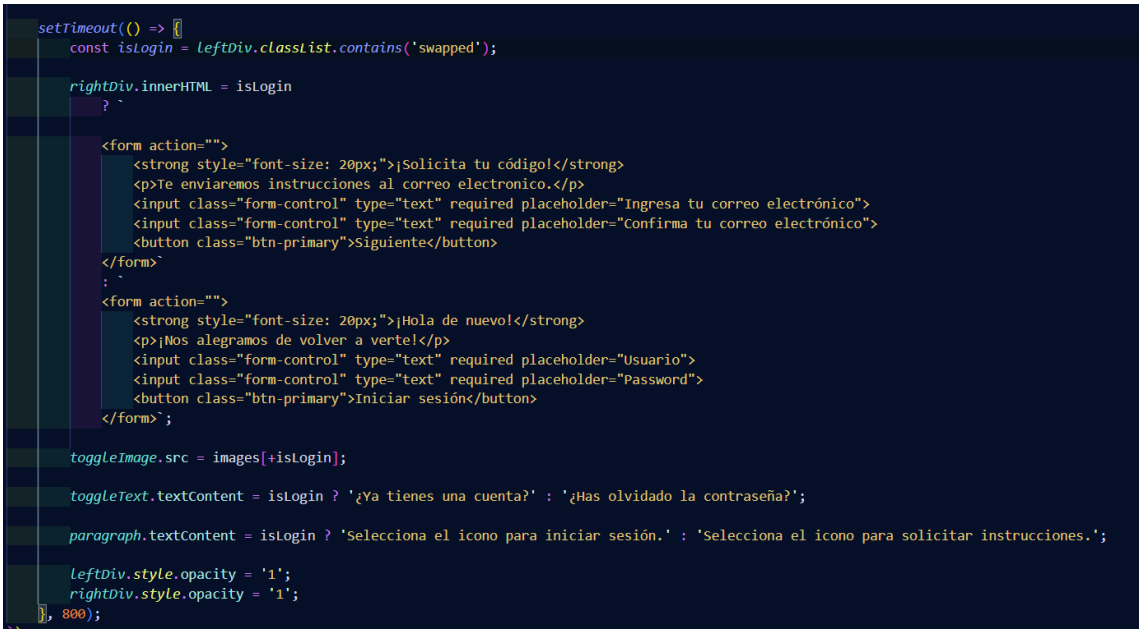
Log in



This screen is for the user to log in. It has the username and password fields, and the button to send the information and validate it.

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <link rel="stylesheet" href="/styles/!important.css">
    <link rel="stylesheet" href="/styles/!color-palette.css">
    <link rel="stylesheet" href="../styles/login.css">
</head>

<body class="d-flex">
    <main class="d-flex m-auto">
        <div id="leftDiv">
            <a class="d-contents" href="#" id="link">
                <img class="p mb rounded-circle" src="../structures/svg/padlock-unlocked.svg" alt="" id="toggleImage">
            </a>
            <strong id="toggleText">Have you forgotten your password?</strong>
            <p id="paragraph">Select the icon to request instructions</p>
        </div>
        <form action="../process/incidents/addIncident.html" id="rightDiv">
            <strong style="font-size: 20px;">Hello again!</strong>
            <p>We are glad to see you again!</p>
            <input class="form-control" type="text" required placeholder="Usuario">
            <input class="form-control" type="text" required placeholder="Password">
            <button class="btn-primary">Login</button>
        </form>
    </main>
    <script src="login.js"></script>
</body>

</html>
```

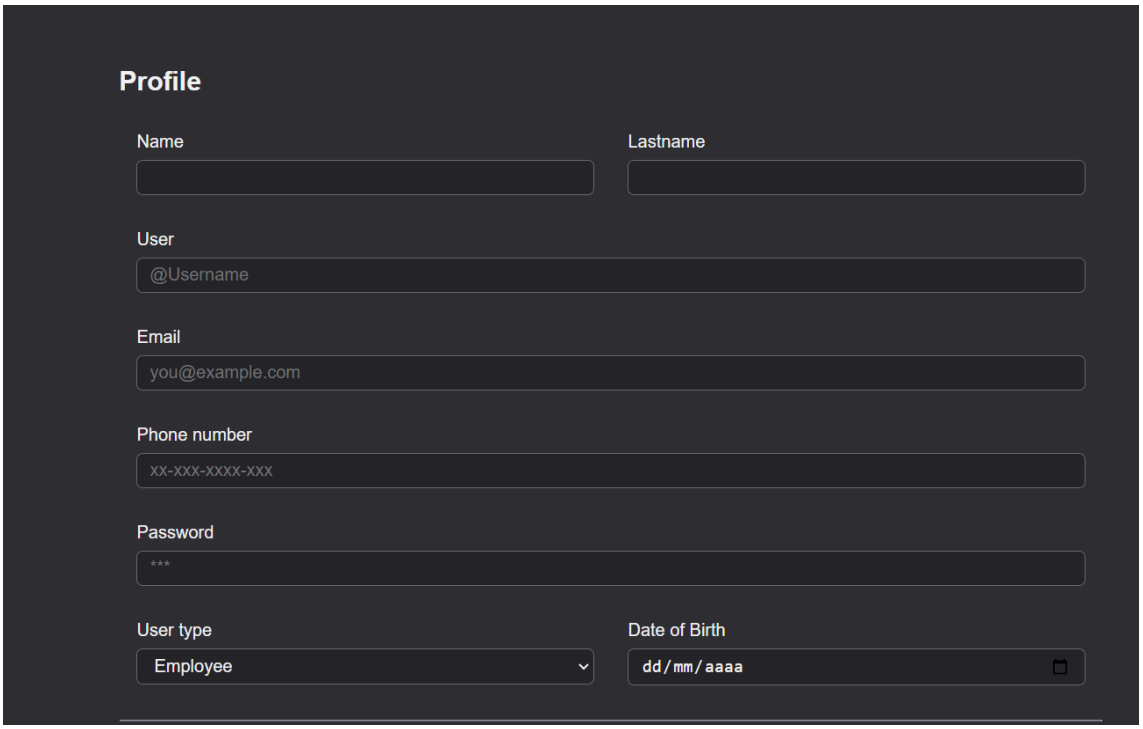
As you can see, the file is in a post-type form where user data is captured.

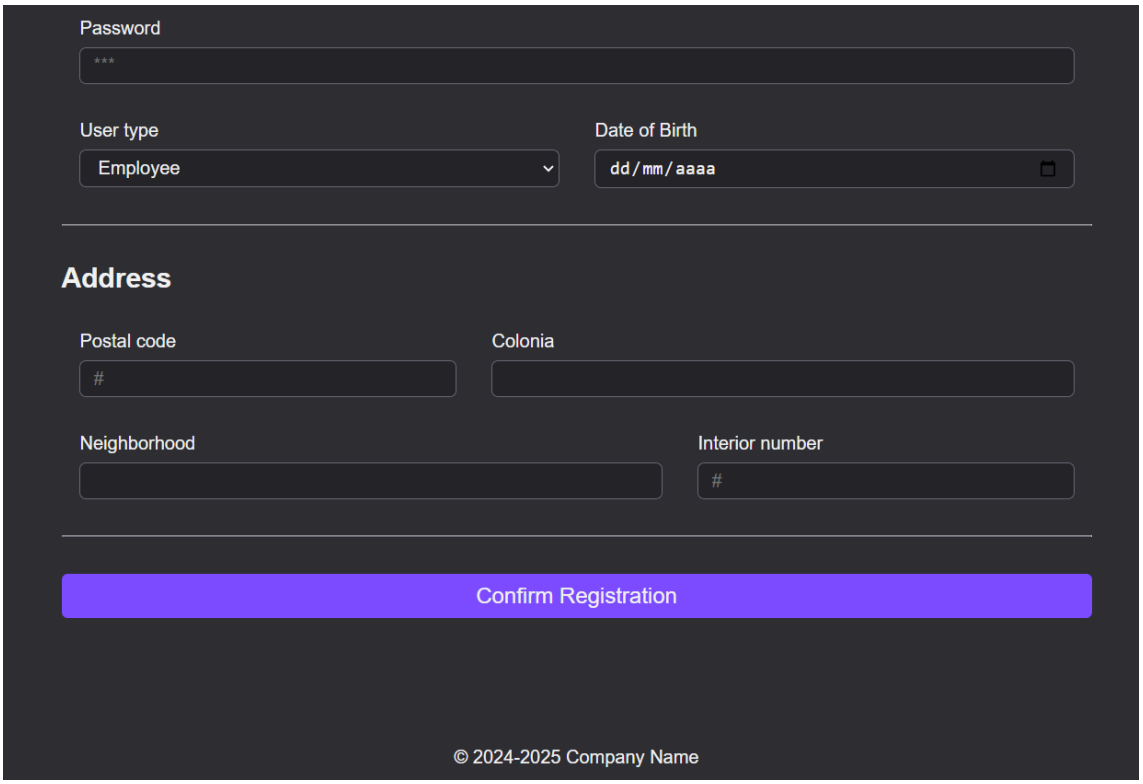


```
setTimeout(() => {
    const isLogin = leftDiv.classList.contains('swapped');

    rightDiv.innerHTML = isLogin
        ? `

        <form action="">
            <strong style="font-size: 20px;">¡Solicita tu código!</strong>
            <p>Te enviaremos instrucciones al correo electronico.</p>
            <input class="form-control" type="text" required placeholder="Ingresa tu correo electrónico">
            <input class="form-control" type="text" required placeholder="Confirma tu correo electrónico">
            <button class="btn-primary">Siguiente</button>
        </form>`
        : `
        <form action="">
            <strong style="font-size: 20px;">¡Hola de nuevo!</strong>
            <p>¡Nos alegramos de volver a verte!</p>
            <input class="form-control" type="text" required placeholder="Usuario">
            <input class="form-control" type="text" required placeholder="Password">
            <button class="btn-primary">Iniciar sesión</button>
        </form>`;

    toggleImage.src = images[+isLogin];

    toggleText.textContent = isLogin ? '¿Ya tienes una cuenta?' : '¿Has olvidado la contraseña?';

    paragraph.textContent = isLogin ? 'Selecciona el icono para iniciar sesión.' : 'Selecciona el icono para solicitar instrucciones.';

    leftDiv.style.opacity = '1';
    rightDiv.style.opacity = '1';
}, 800);
```

Manages the behavior of a user interface to switch between a login form and a code request form. Clicking a link hides both forms, changes the content of the right form based on the current state

Add user



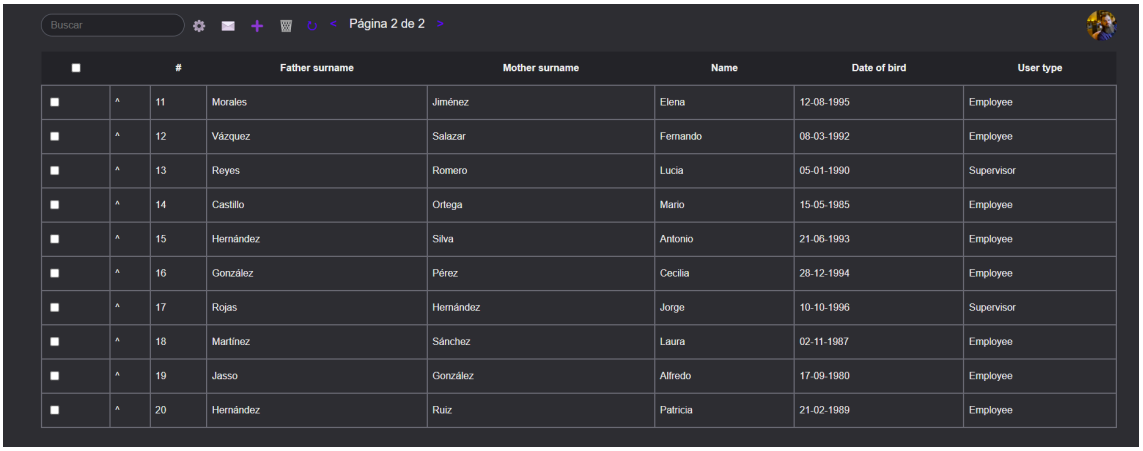


Allows you to create and register new users in the packaging management system. It provides fields to enter essential information such as name, email, role… This functionality is essential to manage user access and permissions, ensuring that only authorized people can interact with the system and contribute to the efficiency of the packaging process.

```php
<?php
require "/config.php";
function addUser(
    $username, $password,
    $name, $firstSurname, $secondSurname,
    $dateOfBirth, $neighborhood, $street, $postalCode,
    $phone, $email, $active, $userType, $supervisor
){
    $db = connectdb();
    $query = "call sp_ser(".
        $username.",". $password.",".
        $name.",". $firstSurname.",". $secondSurname.",".
        $dateOfBirth.",". $neighborhood.",". $street.",". $postalCode.",".
        $phone.",". $email.",". $active.",". $userType.",". $supervisor
    .");";

    try {
        $response = mysqli_query($db, $query);
        return true;
    } catch (Exception $e) {
        return false;
    }
}
```

Here will be all the logic to make user insertions through stored procedures

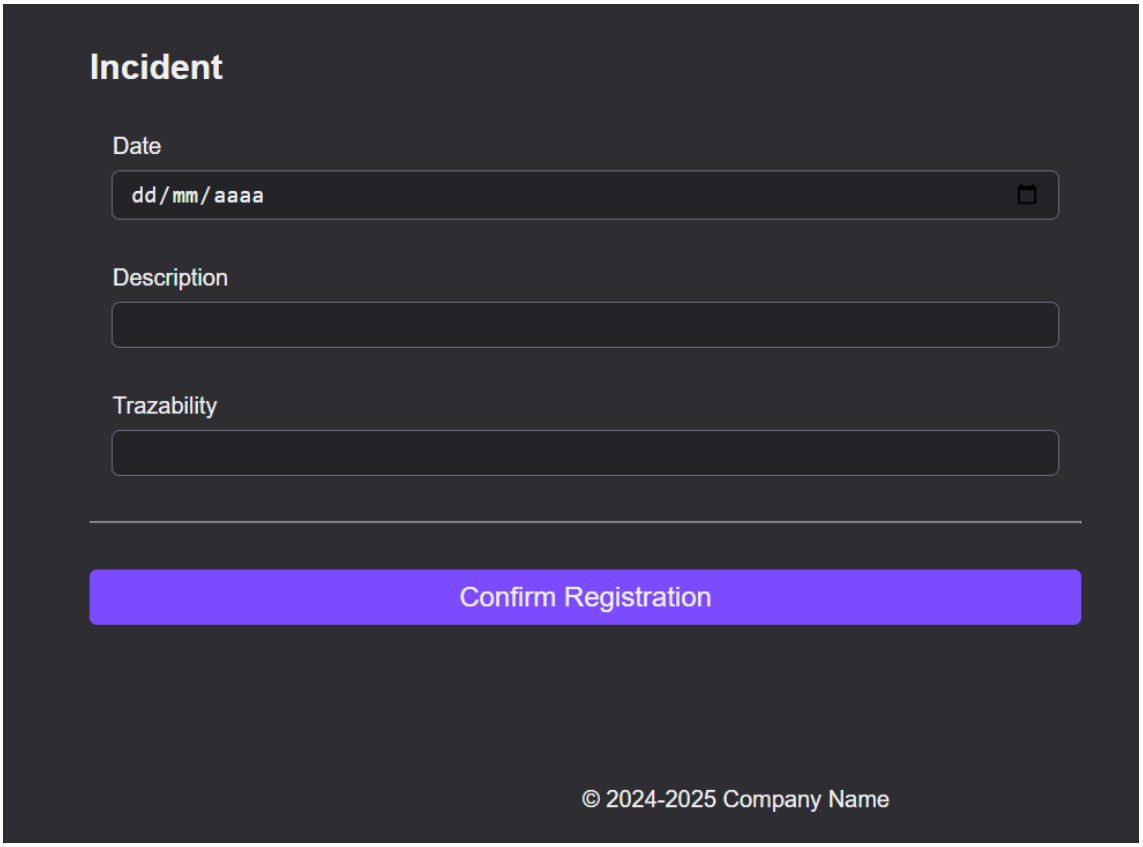
List user



	#	Father surname	Mother surname	Name	Date of bird	User type
	11	Morales	Jiménez	Elena	12-08-1995	Employee
	12	Vázquez	Salazar	Fernando	08-03-1992	Employee
	13	Reyes	Romero	Lucia	05-01-1990	Supervisor
	14	Castillo	Ortega	Mario	15-05-1985	Employee
	15	Hernández	Silva	Antonio	21-06-1993	Employee
	16	González	Pérez	Cecilia	28-12-1994	Employee
	17	Rojas	Hernández	Jorge	10-10-1996	Supervisor
	18	Martínez	Sánchez	Laura	02-11-1987	Employee
	19	Jasso	González	Alfredo	17-09-1980	Employee
	20	Hernández	Ruiz	Patricia	21-02-1989	Employee

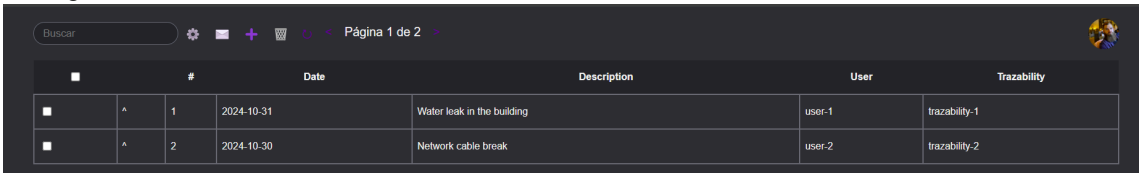
This screen will be used to view all users.

Add incident



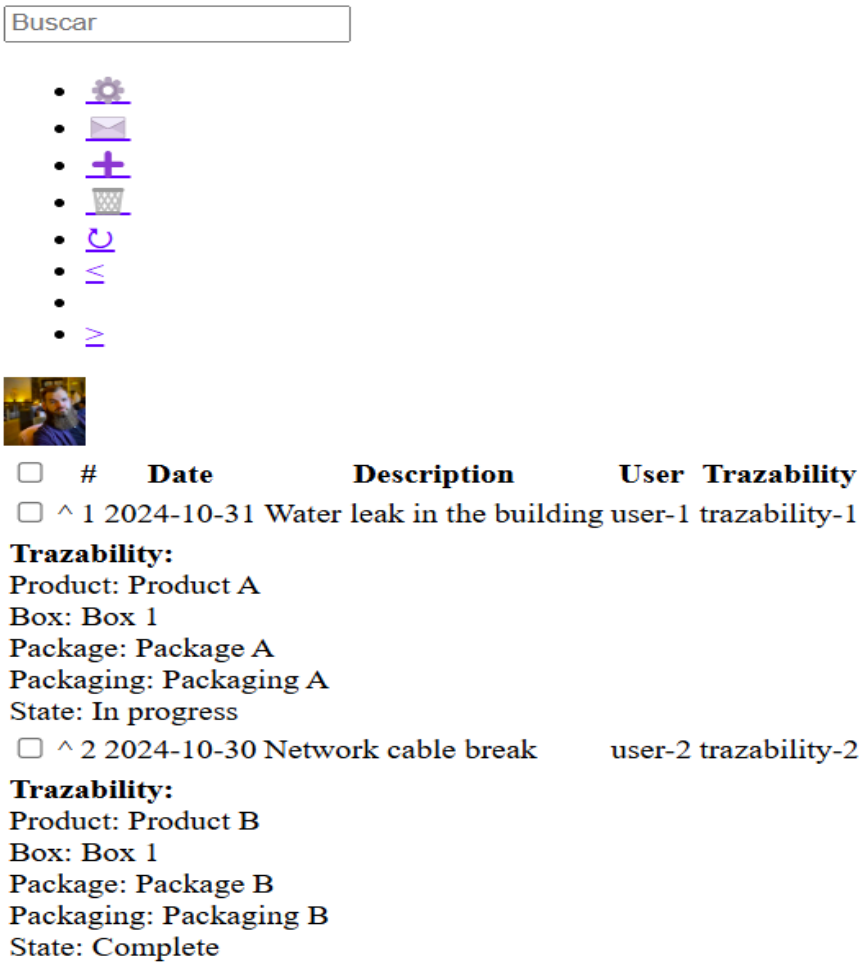
This screen allows you to quickly and accurately record any problems or unexpected events during the packaging process. It makes it easy to track each incident by capturing information such as date, description and traceability details.

Manage incident



Allows you to view, edit and manage all incidents recorded in the packaging system. Provides an overview of each incident with key information such as date, description and the user who recorded it.

List incident



Displays a complete listing of all incidents recorded in the packaging system, organized by date, type, or status. Facilitates quick review of historical incidents, with filters and search options to locate specific events.

Color palet

```css
:root {
    --background-light: #2c2f33;
    --background-dark: #23272a;
    --background-not-hover: #404249;
    --background-hover: #393B40;
    --text-light: #F2F2F2;
    --icon-dark: #D7D8D9;
    --text-gray: #B8BBBF;
    --text-dark: #686B73;
    --blue-medium: #5C73F2;
    --blue-dark: #495ABF;
    --green: #2A8C4C;
    --green-dark: #18592F;
    --red: #D93B48;
    --red-dark: #A62D2D;
}
```

In this CSS section you can see our color palette, which is diverse and used throughout the website.

Forms

```css
div,
form {
    width: 50%;
    padding: 20px;
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    text-align: center;
    transition: transform 0.5s ease, opacity 0.5s ease;
}
```

In general the forms that were created here use the same properties, which are these.

Body

```css
body {
    background-color: var(--background-light);
    color: var(--text-light);
    font-family: sans-serif;
    margin: 0;
}

h2 {
    margin: 0;
}

.container {
    width: 100%;
    max-width: 960px;
    margin-inline: auto;
}
```

The body and containers used, as well as the area of the forms, are the same class that is used in them.

Bar Hub

```css
.nav-link {
    fill: var(--text-gray);
    display: block;
    padding: 1rem;
    transition: background-color .15s ease-in-out, fill .15s ease-in-out;
}

.nav-link:hover {
    background-color: var(--background-hover);
    fill: var(--text-light);
}

.nav-link.active {
    background-color: var(--background-not-hover);
    fill: var(--text-light);
}

.nav-link.active:hover {
    background-color: var(--background-hover);
    fill: var(--icon-dark);
}
```

The menu uses a single navigation bar, which uses this style.