Technical documentation

“Delivery of the vehicle”

Created by:

Peter Duška

Dmytro Moroz

Martin Krivosudský

Viera Michaela Blažíčková

Content of requirements document

1. Introduction

1.1 Purpose of requirements document

The purpose of this document is to define the requirements for the development of a feedback and complaint system for CEVA Logistics. This document is intended for stakeholders, ensuring a clear understanding of the functionality and purpose of the system. It's a binding agreement about specifications for the project.

1.2 Scope of the product

The system will allow customers to provide feedback and submit complaints upon the delivery of vehicles. The system will primarily be web-based, accessible via a link or QR code. The customer will be able to fill out a form (that consists of the VIN code [see 1.3.2] or order number, a description of the damage and a photo). Once a complaint is submitted, it will be routed to the appropriate CEVA *agent* [see 2.3] for using the data from the database for analysis.

1.3 Definitions, acronyms and abbreviations

1.3.1 All system user roles (Administrator, Agent, Customer) defined in 2.3
1.3.2 VIN = Vehicle identification number
1.3.3 PDI = Pre-delivery inspection
1.3.4 Audatex system = computer programs for insurance companies, car repair shops and experts to calculate the cost of repairing damage to vehicles.
1.3.5 C pillar = the support between a car's rear window and rear side window.

1.4 References

- Project on GitHub : https://github.com/TIS2024-FMFI/vehicle-delivery.git

1.5 Overview of the remainder of the document

The next paragraphs describe the system in more detail, including its overall structure and user requirements. Chapter 2 covers the general description of the system, providing an overview of its functionality, user roles, constraints, and

dependencies. Chapter 3 outlines the specific functional and non-functional requirements for the system. The appendices and index provide additional reference materials and terminologies.

2. General description

2.1. Product perspective

The developed system will serve as a platform for recording customer feedback, submitting complaints, and facilitating the exchange of relevant information between customers and CEVA Logistics.

The system will be designed to accelerate and simplify the process of submitting, tracking, and resolving customer issues for both customers and internal agents. By automating key steps in the workflow - such as notifying the responsible personnel, gathering necessary information, and generating statistical reports - the system will reduce the time and effort required from both the customers and CEVA *agents*.

2.2 Product functions

Feedback and Complaint Submission

Customers will be able to submit feedback or report an issue through a web-based form. The form will allow them to describe the problem and include critical details such as a *description* of the issue, *photos* of the damage or problem, and the *VIN* [see 1.3.2] or *order number*.

The system will ensure that all required fields are filled out, helping CEVA gather all necessary information to resolve the complaint quickly and efficiently.

Pre-filled Form for Customer Convenience

Customers will access the web form via a QR code or a link. CEVA will mediate this link for their customers. [see 3.2]

Data Storage

Once the customer submits the feedback, all information (including the problem description, photos, *VIN* or *order number* and customer details such as *name,*

contact details) will be automatically saved in the system's database. Photos will be stored in folders on disk. [see 3.3] Path to these photos will be saved in database with the rest of feedback.

The database will serve for all complaints, making it easy for CEVA *agents* to track and manage records. [see 3.11] Data will be stored for 2 years [see 3.20]

Automated Notifications

Upon the successful submission of a complaint, the system will automatically notify the appropriate agent via email. [see 3.4] Notifications will be triggered whenever a new record is added to the database. Customer will receive confirmation about successful submission of their *complaint*. [see 3.5]

Complaint Tracking and Management

CEVA *agents* will have to log into the system using secure credentials to access the *complaint* database and manage records. [see 3.9]

Statistical Reporting and Analysis

Data in the database will help CEVA analyze trends, it will show how many complaints of which type were received each month. [see 3.16]

2.3 User characteristics

- **Customers**: Individuals who have received a vehicle delivery. Their access to the system will be limited to submitting *complaints*.
- **Agents**: CEVA employees who will work with the system to resolve *complaints*.
- **Administrators**: Users that have all the permissions of *Agents* but will also have additional access such as *creating* and *managing* accounts in the system.

2.4 Assumptions and dependencies

The system will be a web application, requiring an internet connection and web browser for access. This setup allows users to operate the system directly from any internet-enabled device, without needing to install additional software. System will be developed in one language but with the option to add additional languages.

Python version 3.10 or newer needed.

Python libraries to install:

- Django
- openpyxl
- functools
- crispy-bootstrap5
- python-dotenv
- Pillow

https://mlocati.github.io/articles/gettext-iconv-windows.html has to be installed for the translations to work

3. Specific requirements

(requirements with ** will be done if there will be enough manpower)

1. Customers can submit feedback and *complaints* via a web form.
2. Customers can access the form via a generic link.
3. Customers can upload photos to the *form*.
4. Upon submission of a *complaint*, the system will send an email notification to the assigned *agent*.
5. Upon submitting a *complaint*, the system will send a confirmation email to the customer.
6. The *Administrator* will be able to register the *agents* into the system using an username and password.
7. The *Administrator* will be able to assign departments to users. Departments will have assigned type of *complaint* they are working with. Departments can represent company departments.
8. The *Administrator* will be able to delete *agents* account, this will also delete their information from database.

9. Once registered, *agents* can log in with the username and password the *administrator* set during the registration.
10. An *administrator* will have the ability to reset passwords if an *agent* forgets theirs.
11. *Agents* will have access to all *complaints* in the database, they will be shown in the web application as a list that can be filtered (based on type, status, date), and each *complaint* can be opened to show all information about it.
12. The *complaints* will have *statuses* depending on what is happening with them:
    - 12.1. New
    - 12.2. Opened
    - 12.3. Archived
13. *Agents* will be able to change the status of *complaints*.
14. *Agents* will have the option to export *complaints* to an .xlsx (Excel) file. **
15. The exported .xlsx (Excel) file will include all information submitted by the customer regarding the specific *complaint*. **
16. *Analysis*: the *agents* will be able to go to an *Analytics* page, where the system will show how many *complaints* of each type were received / resolved / declined for which month. *Administrators* will be able to see how many complaints were resolved by which *agent*. **
17. *Complaint* data will be stored for 2 years.
18. The system will initially be developed in one language, with the option to add more languages through language configuration files at any time.
19. *Users* will be able to change the language at any time during use.
20. In the *form*, there will be 5 types of *complaints* available:
    - 20.1. Vehicle damage/ vehicle dirty [see 3.24]
    - 20.2. Transport (delivery delay, parking) [see 3.31]
    - 20.3. Vehicle preparation (incorrect PDI) [see 3.33]
    - 20.4. Communication (driver complaints, communication with dispatcher) [see 3.35]
    - 20.5. Other [see 3.35]
21. Every type of *complaint* will start with text fields with information about the customer (fields with * are mandatory):
    - 21.1. Dealer / company name *
    - 21.2. Name *
    - 21.3. Surname *
    - 21.4. Email *
    - 21.5. Telephone number *
22. Every *complaint* type will have a checkbox at the end (fields with * are mandatory):
    - 22.1. Consent to data management (description: I agree to processing of my data in CRM system IOS Hungary Kft. THE) *

23. Every type of *complaint* will have a button at the end:
    - 23.1. Submit
24. Vehicle damage/ vehicle dirty *complaint* will be split into parts:
    - 24.1. Delivery information
    - 24.2. Damaged vehicle information
    - 24.3. Additional information / Observation
    - 24.4. Files
25. Vehicle damage/ vehicle dirty *complaint* will have text fields in the *Delivery information* part (fields with * are mandatory):
    - 25.1. Registration number of the towing vehicle
    - 25.2. Registration number of trailer
    - 25.3. Date of unloading *
    - 25.4. Country of unloading *
    - 25.5. Place of unloading (ZIP code) *
    - 25.6. City of unloading *
26. Vehicle damage/ vehicle dirty *complaint* will have text fields in the *Damaged vehicle information* part (fields with * are mandatory):
    - 26.1. VIN number of vehicle *
    - 26.2. Nature of the damage 01 *
    - 26.3. Place of the damage 01 *
    - 26.4. Nature of the damage 02
    - 26.5. Place of the damage 02
    - 26.6. Nature of the damage 03
    - 26.7. Place of the damage 03
    - 26.8. Nature of the damage 04
    - 26.9. Place of the damage 04
27. In the Vehicle damage/ vehicle dirty *complaint* each *Nature of the damage* will be a roll list that will show to the customer options to choose, and they will be in format (Code; Description):
    - 27.1. BR; Burned
    - 27.2. CA; Broken, teared off, cracked
    - 27.3. CD; Cut, torn
    - 27.4. CE; Pierced, cut
    - 27.5. EC; paint peeled
    - 27.6. EN; Dented
    - 27.7. MA; Missing
    - 27.8. PR; Damaged
    - 27.9. RF; Scratched, scuffed
    - 27.10. TA; Stained
    - 27.11. XX; Not specified

28. In the Vehicle damage/ vehicle dirty *complaint* each *Place of the damage* will be a roll list that will show to the customer options to choose, and they will be in format (Code; Description):
    - 28.1. 01; Front wing
    - 28.2. 02; Front bumper
    - 28.3. 03; Hood
    - 28.4. 04; Front bumper protection strip
    - 28.5. 05; A pillar
    - 28.6. 06; Roof
    - 28.7. 07; Roof window
    - 28.8. 08; Roof rack
    - 28.9. 09; Hinged roof
    - 28.10. 10; Windscreen
    - 28.11. 11; Trunk door
    - 28.12. 12; Rear bumper
    - 28.13. 13; Tail board
    - 28.14. 14; Rear bumper protection strip
    - 28.15. 15; Driver door – lining
    - 28.16. 16; Driver door – door sill
    - 28.17. 17; Door handle – front
    - 28.18. 18; Door handle – rear
    - 28.19. 19; Puncture repair kit
    - 28.20. 20; Immobilizer code card
    - 28.21. 21; Radio antenna
    - 28.22. 22; Dirt traps
    - 28.23. 23; Windshield washers / Headlight wipers
    - 28.24. 24; Front wipers
    - 28.25. 25; Rear wiper
    - 28.26. 26; Rear spoiler
    - 28.27. 27; Decorative wheel covers
    - 28.28. 28; Additional floor mats – set
    - 28.29. 29; User manual
    - 28.30. 30; Right rearview mirror
    - 28.31. 31; Right front fender
    - 28.32. 32; Right front wheel: rim
    - 28.33. 33; Right front wheel: tire
    - 28.34. 34; Right wheel arch
    - 28.35. 35; Right rear fender and C pillar
    - 28.36. 36; Right front door
    - 28.37. 37; Right rear door
    - 28.38. 38; Right side panel (utility vehicle)

28.39. 39; Protective moldings – right side
28.40. 40; Front turn signals
28.41. 41; Fixed front glass left/right (for 36/56)
28.42. 42; Exhaust
28.43. 43; Tow eye cover front/rear
28.44. 44; Tools + bag
28.45. 45; Miscellaneous
28.46. 46; Keys
28.47. 47; Radiator grille
28.48. 48; Rear fog lights
28.49. 49; Third brake light
28.50. 50; Left rearview mirror
28.51. 51; Left front fender
28.52. 52; Left front wheel: rim
28.53. 53; Left front wheel: tire
28.54. 54; Left wheel arch
28.55. 55; left rear fender and C pillar
28.56. 56; Left front door
28.57. 57; Left rear door
28.58. 58; Left side panel (utility vehicle)
28.59. 59; Protective moldings – left side
28.60. 60; Fuel filter / cable cap
28.61. 61; Logo / emblem
28.62. 62; Carpet / mat in battery compartment
28.63. 63; License plate lighting
28.64. 64; Charging cable (electric vehicle)
28.65. 65; Rear lights
28.66. 66; Fixed rear glass left / right (for 37 / 57) + C pillar
28.67. 67; Rear side glass (for 48 / 58)
28.68. 68; Remote control key
28.69. 69; Door locks
28.70. 70; Center pillar
28.71. 71; Speakers
28.72. 72; Front / rear seats
28.73. 73; Right rear wheel: rim
28.74. 74; Right rear wheel: tire
28.75. 75; Side turn signals
28.76. 76; Headlights
28.77. 77; Front fog lights
28.78. 78; Rear shelf
28.79. 79; Vehicle kit

28.80. 80; Rear glass
28.81. 81; Wheel wrench
28.82. 82; Front side windows (for 36 / 56)
28.83. 83; Rear side windows (for 37 / 57)
28.84. 84; Lighter
28.85. 85; Navigation, CD-ROM, remote control
28.86. 86; Radio, radio accessories
28.87. 87; CD changer
28.88. 88; Front headrests
28.89. 89; Rear headrests + middle
28.90. 90; Safety key extension
28.91. 91; FFA/DC41
28.92. 92; Jack (kit)
28.93. 93; Left rear wheel: rim
28.94. 94; Left rear wheel: tire
28.95. 95; Spare tire

29. Vehicle damage/ vehicle dirty *complaint* will have text fields in the *Additional information / Observation* part:
    29.1. Additional information / Observation
30. Vehicle damage/ vehicle dirty *complaint* will have fields for adding files in the *Files* part (fields with * are mandatory):
    30.1. Waybill *
    30.2. Damage report *
    30.3. Photos showing the whole vehicle *
    30.4. Photo of VIN behind the windshield *
    30.5. Photo of damaged part (use of measuring device mandatory) *
    30.6. Photo of damaged part from distance so it's possible to recognize the vehicle part
    30.7. Repair calculation (in AUDATEX system) [see 1.3.4]
    30.8. Other
31. Transport (delivery delay, parking) *complaint* will have text fields:
    31.1. Date *
    31.2. VIN number of vehicle *
    31.3. Vehicle model
    31.4. License number of towing vehicle *
    31.5. Short description *
32. Transport (delivery delay, parking) *complaint* will have field for adding files:
    32.1. Photo (incorrect parking of towing vehicle)
33. Vehicle preparation (incorrect PDI) [see 1.3.3] *complaint* will have text fields:
    33.1. Date *
    33.2. VIN number of vehicle *

33.3. Vehicle model
33.4. Issue description *

34. Vehicle preparation (incorrect PDI) *complaint* will have field for adding files:
    34.1. Photo (non-functional system, removable part not installed)
35. Communication (driver complaints, communication with dispatcher) and other *complaints* will have text fields:
    35.1. Date *
    35.2. Issue description *
36. Communication (driver complaints, communication with dispatcher) and other *complaints* will have fields for adding files:
    36.1. Photo

System Design Document

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed system design for the feedback and complaint management system for CEVA Logistics. It captures the architectural components, workflows, and requirements necessary to meet the goals outlined in the requirements.

1.2 Scope

The system is a web-based application enabling customers to submit feedback and complaints, allowing agents to view and export complaints, and providing administrators with user management and statistics. The application will feature a multilingual interface.

2. System Overview

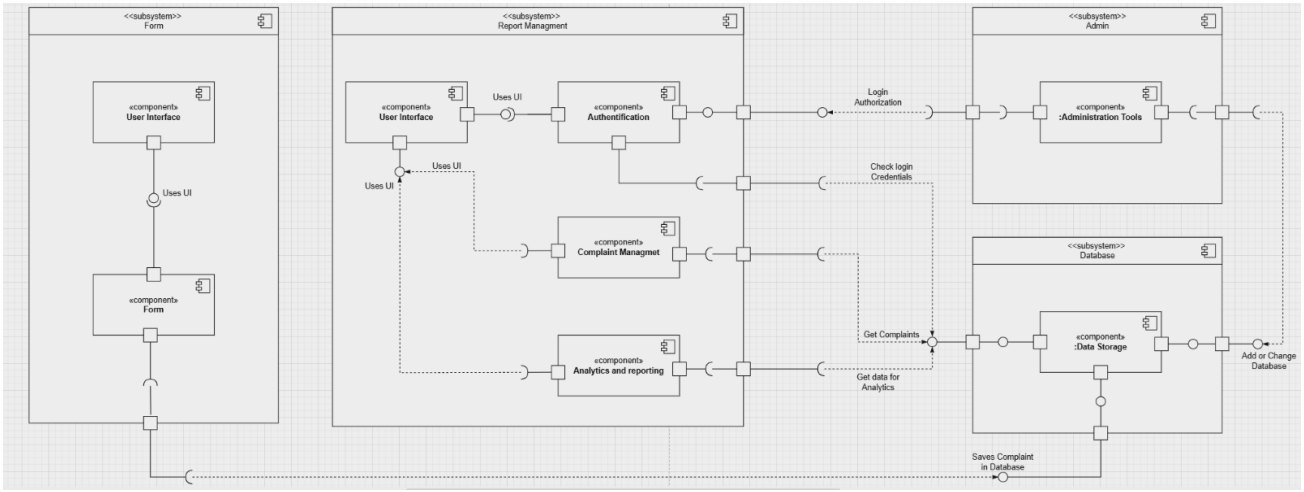
2.1 User Roles and Access Levels

1. **Customers:** Submit feedback and complaints through web forms.
2. **Agents:** Manage complaints, change complaint statuses, and generate reports.
3. **Administrators:** Perform all agent tasks and manage user accounts, permissions, and system configurations.

2.2 System Components

1. **User Interface (UI) Component:**
    a. Web forms for complaint submission, filtering, and viewing.
    b. Analytics dashboard for administrators and agents.
    c. Multilingual support for UI elements.
2. **Authentication and Authorization:**
    a. Role-based access control (customer, agent, administrator).
    b. Login/logout and password reset functionality.

3. **Complaint Management:**
    a. Complaint submission with support for file attachments.
    b. Complaint status management
4. **Data Storage and Retrieval:**
    a. Central database for users, complaints, logs.
    b. Structured file storage for uploaded photos.
5. **Notification System:**
    a. Email notifications for complaint submissions and assignments.
6. **Analytics and Reporting:**
    a. Statistical insights on complaint trends.
    b. Complaint export to Excel files.
7. **Administration Tools:**
    a. User and group management.
    b. Configuration of system settings and complaint types.



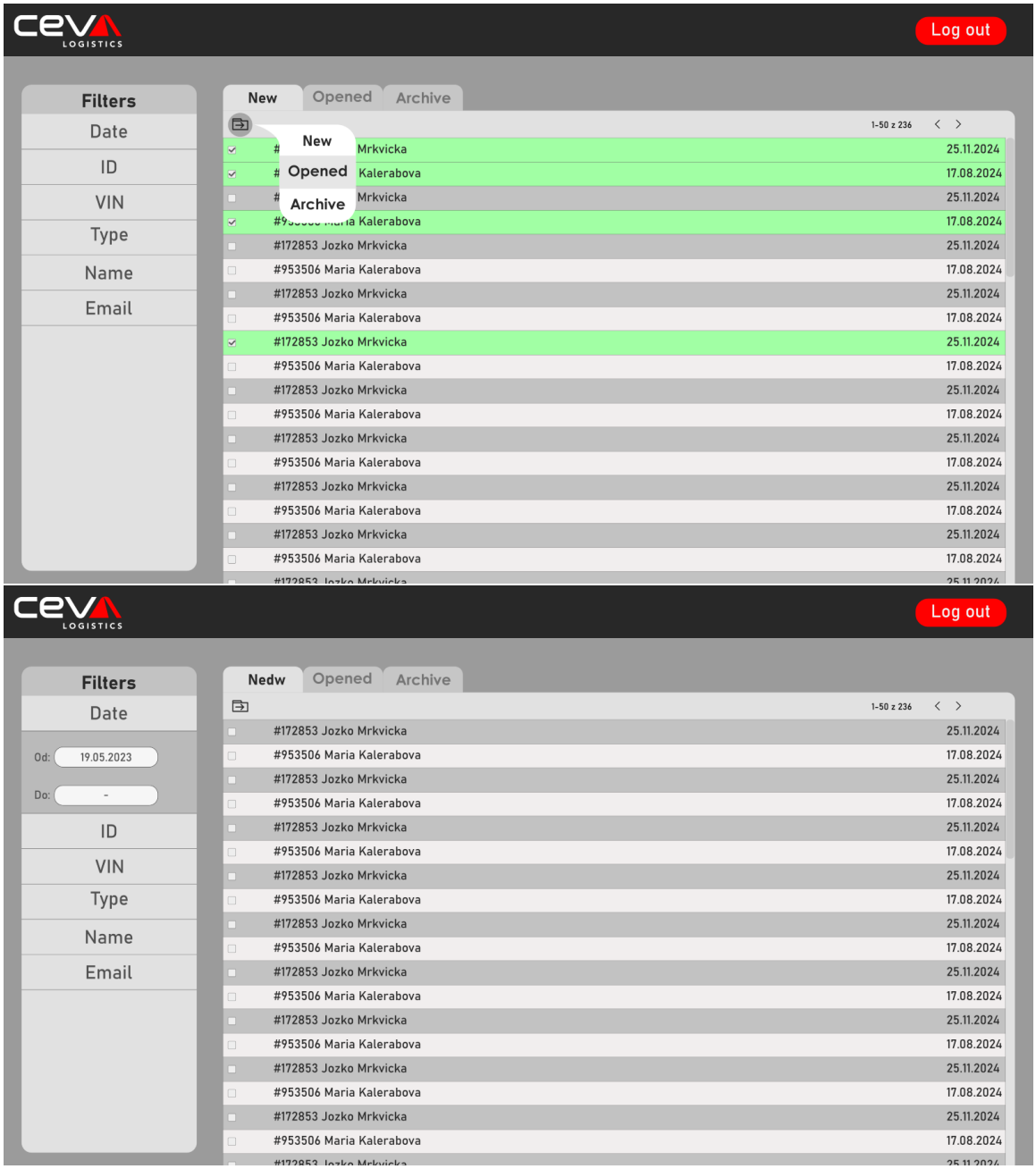
3. Design

3.1 User Interface Design

The UI will consist of:

- **Customer Complaint Form:**

- **Agent Dashboard:**



- **Admin Dashboard:**
  - Access to all agent functionalities.
  - Tools for managing users, groups, and system configurations.

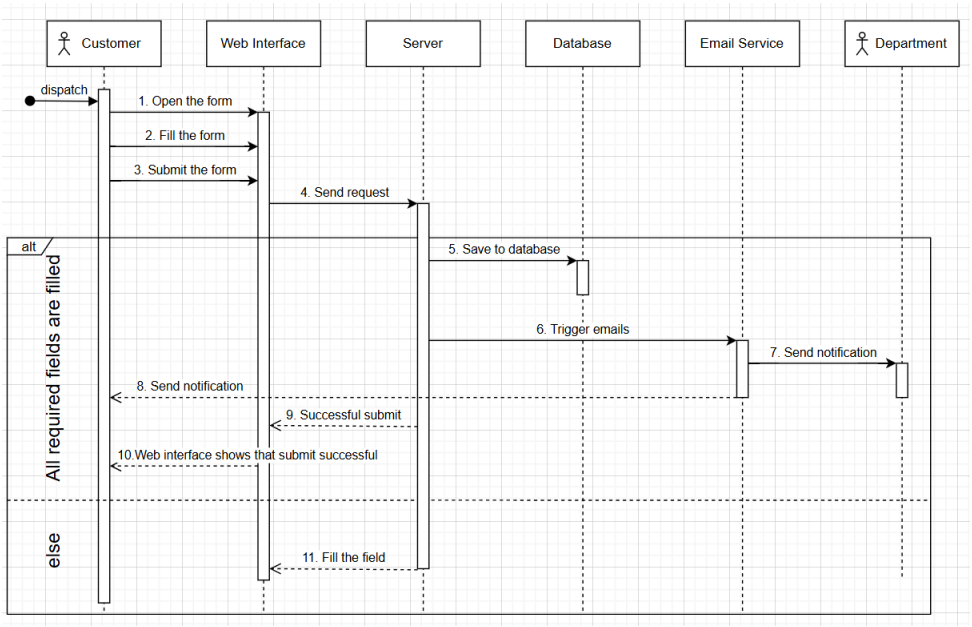
- **Analytics Dashboard:**

3.2 Workflow

Customer Workflow

1. Access the web form via QR code or link.
2. Fill in complaint details and upload necessary files.
3. Submit the form.
4. Receive confirmation via email.

Sequence diagram for filling the form by customer:



Agent Workflow

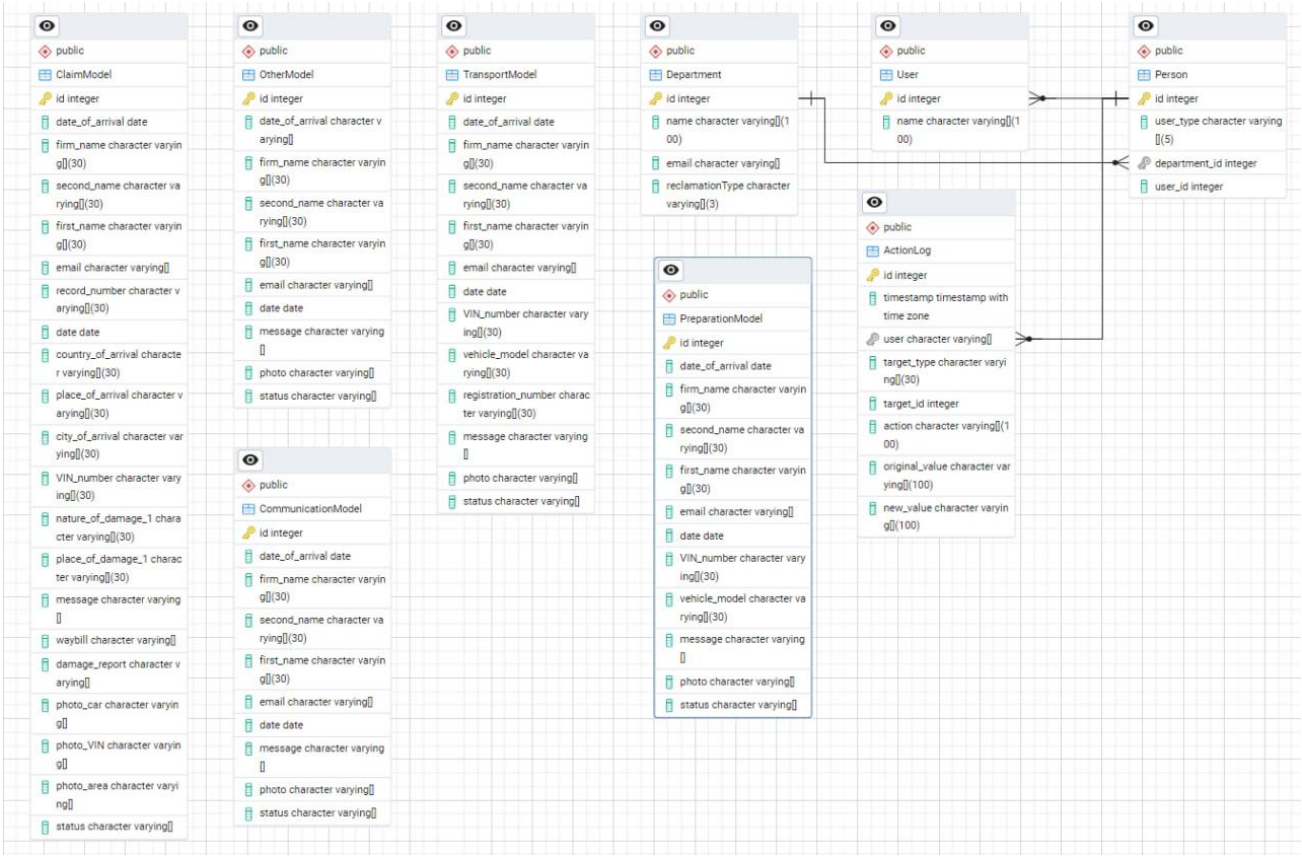
1. Log in to the dashboard.
2. View and filter complaints.
3. Update complaint status as necessary.
4. Export complaint data if required.

Administrator Workflow

1. Log in to the dashboard.
2. Add, delete and change users and departments.
3. Check logs data.
4. Access analytics and export data.

3.3 Database Design

Data model:



Tables:

ClaimModel

- id (integer, primary key)
- date_of_arrival (date)
- firm_name (varchar(30))
- second_name (varchar(30))
- first_name (varchar(30))
- email (varchar)
- record_number (varchar(30))
- date (date)
- country_of_arrival (varchar(30))
- place_of_arrival (varchar(30))
- city_of_arrival (varchar(30))
- VIN_number (varchar(30))

- nature_of_damage_1 (varchar(30))
- waybill_number (varchar(30))
- damage_report (varchar)
- photo_car (varchar)
- photo_VIN (varchar)
- photo_area_of_damage (varchar)
- status (varchar)

OtherModel

- Similar structure to ClaimModel, containing fields for arrival date, firm details, personal details, email, and damage-related information.

TransportModel

- id (integer, primary key)
- date_of_arrival (date)
- VIN_number (varchar(30))
- vehicle_model (varchar(30))
- registration_number (varchar(30))
- message (varchar)
- photo (varchar)
- status (varchar)

CommunicationModel

- Similar structure to ClaimModel, with additional fields for messages and photos.

PreparationModel

- Similar structure to TransportModel, with attributes related to preparation records.

Department

- id (integer, primary key)
- name (varchar(100))
- email (varchar)

- reclamationType (varchar(100))

User

- id (integer, primary key)
- name (varchar(100))

Person

- id (integer, primary key)
- user_type (varchar(5))
- department_id (integer, foreign key referencing Department.id)
- user_id (integer, foreign key referencing User.id)

ActionLog

- id (integer, primary key)
- timestamp (timestamp with timezone)
- user (varchar)
- target_type (varchar(30))
- target_id (integer)
- action (varchar(100))
- original_value (varchar(100))
- new_value (varchar(100))

Relationships

- Person references Department via department_id.
- Person references User via user_id.
- ActionLog tracks changes made by users to various entities.

File Storage:

- Structured directories for storing uploaded photos. Complaint ID will be the name of the folder which will have all the photos uploaded with the complaint.

3.4 Notifications

- **Trigger:** Complaint submission (customer and agent).

- **Mechanism:** Emails sent via a notification service.

3.5 Analytics and Reporting

- **Features:**
  - Can choose period of which statistics will be made.
- **Data Presentation:** Tables.

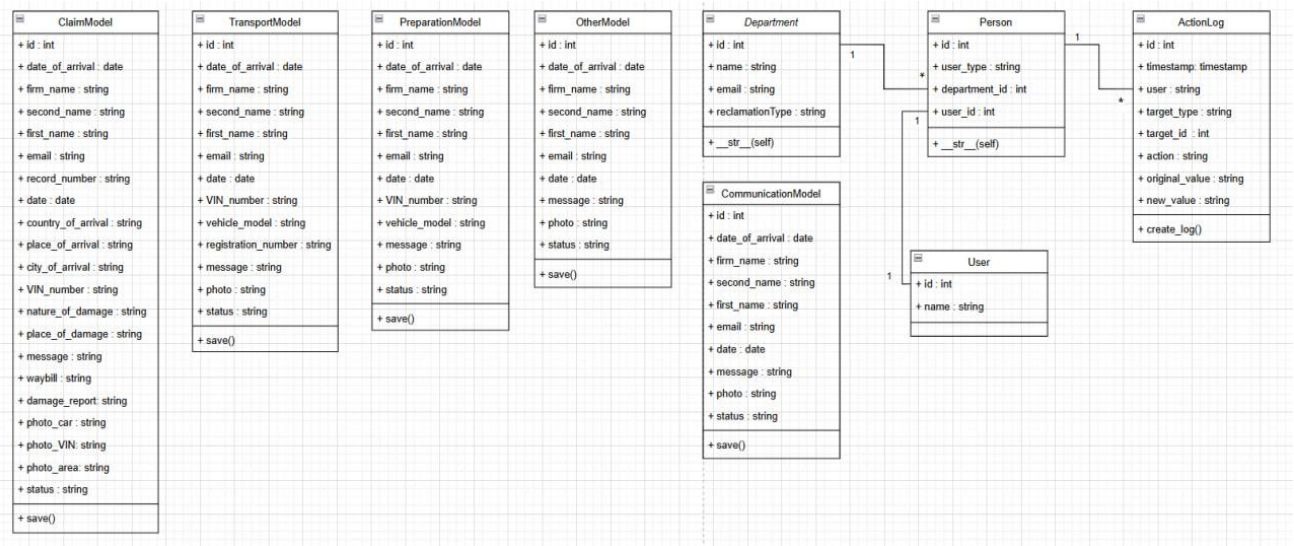
4. Non-Functional Requirements

1. **Security:** Secure data with encryption and access control.
2. **Localization:** Add support for multiple languages with configuration files.
3. **Data Retention:** Retain complaint data for two years.

5. Technologies

- **Frontend:** HTML, CSS, JavaScript.
- **Backend:** Python Django.
- **Database:** MySQL.
- **File Storage:** Local storage.

Class diagram



Tables:

- ClaimModel
- OtherModel
- TransportModel
- CommunicationModel

- PreparationModel
- Department
- User
- Person
- ActionLog

Relationships

- Person references Department via department_id.
- Person references User via user_id.
- ActionLog tracks changes made by users to various entities.

Back-end a Front-end

- The frontend makes requests to the backend (e.g., fetching user data, submitting a claim).
- The backend processes the request, interacts with the database, and returns a response.
- The frontend updates the UI based on the response.

6. Conclusion

This system design document provides the necessary details for the implementation of a feedback and complaint management system.

Implementation Plan

1. Initial Setup

- Install Django and initialize the project.
- Configure the development environment: virtual environment, database and required dependencies.

2. Components

User Interface (UI) Component -> Peter

1. **Web Forms**
    a. Design the feedback and complaint submission form (customers).
    b. Implement complaint list views for agents with filters (e.g., by status, type).
    c. Create an analytics dashboard using a library.
2. **Language Options**
    a. Implement a mechanism to add and switch between languages.

When completed: Fully functional multilingual UI, including forms, views.

Authentication and Authorization -> Vierka

1. **User Roles and Permissions**
    a. Define roles: customers, agents, and administrators.
    b. Implement role-based access for different views and actions.
2. **Authentication Features**
    a. Set up login/logout functionality.
    b. Add password reset functionality.

When completed: Secure role-based access and authentication system.

Complaint Management & Notifications -> Dmytro

1. **Complaint Submission**
    a. Create models for complaints (fields: VIN, description, photos, etc.).
    b. Implement submission forms with validation for mandatory fields.
2. **Tracking and Status Management**
    a. Add functionality to update complaint statuses.
3. **Attachments Management**

    a. Implement file upload and association with complaints.
4. **Notification System**
    a. Automate emails for complaint submission confirmations (to customers) and new assignments (to agents).

When completed: Complete complaint submission and notification system.

Data Storage, Analytics, and Administration Tools -> Martin

1. **Database and File Storage**
    a. Create the database for users, complaints.
    b. Set up structured file storage for uploaded photos/documents.
2. **Analytics and Reporting**
    a. Implement statistics tracking: monthly trends, resolved vs unresolved complaints.
    b. Add export functionality.
3. **Administration Tools**
    a. Develop user management features (add, edit, disable accounts).
    b. Create system configuration tools for managing complaint types and settings.

When completed: Efficient data storage, analytics, and administration.

3. Integration

- Integrate components
- Conduct unit tests for individual modules and also integration testing.

4. Deployment

- Deploy the application to a server for testing.

5. Testing & Feedback

- Conduct user testing.
- Collect feedback and iterate on issues or feature enhancements.

Testing Scenarios

1. Customer submitting a new complaint [1, 2, 3, 4, 5, 19]

Steps:

1.1 Action: Customer open complaint via QR code or link
Result: The form opens in web.

1.2 Action: Customer clicks on icon that presents language.
Result: Language options are rolled down.

1.3 Action: Customer chooses option that he wants language to change it to.
Result: Language is changed and everything on website is changed to that language.

1.4 Action: Customer fills the form but lefts some required fields and submit it.
Result: In the form a notice is shown that all mandatory fields are required to be filled for submission.

1.5 Action: Customer fills all required fields but selects an image that is too big and tries to submit it.
Result: In the form a notice is shown that image is too big.

1.6 Action: Customer changes image to one with smaller size and submits the form.
Result: All required fields are completed, pictures aren't too big and it's all submitted. The complaint is saved in the database, and the employee gets a notification via mail. Customer got an email notification that a form was sent.

2. Admin creates new group and user and other stuff [6, 7, 8, 10]

Steps:

2.1 Action: Admin logs in.
Result: Admin is logged in. And sees all complaints and also everything else that he can do. Parts for 'adding groups/departments', 'adding users', 'changing the password of any user', 'deleting user' …

2.2 Action: Admin clicks on 'add group/department'
Result: Form for adding department is opened/popped out.

2.3 Action: Admin fills name of group, email of group, and click on 'add group'
Result: Group is created and added to the database and this form closes.

2.4 Action: Admin clicks on 'add user'
Result: Form for adding user is opened/popped out.

2.5 Action: Admin fills out the form with the user's email, new password, group of which he will be a member and if he is admin (no by default). And click on the button add member/save.
Result: User is added to database and can log in.

2.6 Action: Admin clicks on ‘change password’.
Result: Form for changing password is opened/popped out.

2.7 Action: Admin fills email of agent whose password he is going to change and clicks on ‘change password’ (tries to change to an empty password).
Result: Password is not changed, and a notice is shown to admin that password is too short.

2.8 Action: Admin fills a new password and clicks again on ‘change password’.
Result: Password is changed and saved in database, the form is closed.

2.9 Action: Admin clicks on ‘delete user’.
Result: Form for deleting user is opened/popped out.

2.10 Action: Admin fills an incorrect email of agent and clicks on ‘delete user’.
Result: In form a notice is shown that this user doesn’t exist. So, no user is deleted.

2.11 Action: Admin fills correct email and clicks on ‘delete user’.
Result: User is deleted and can’t log to this system anymore. And this form is closed.

2.12 Action: Admin clicks on ‘log out’.
Result: Admin is logged out.

3. Agent logs, sees complaints, ... [9, 11, 13, 14, 15, 19]

Steps:

3.1 Action: Agent opens the website.
Result: Log in formular opens to be filled in with email and password.

3.3 Action: Agent fills out login information, but makes a mistake and clicks the button ‘log in’.
Result: Red text appears that email or password are wrong.

3.4 Action: Agent fills out right login information this time and clicks the button ‘log in’.
Result: Agent is logged in and sees all complaints (and can manage data).

3.5 Action: Agent clicks on complaint.
Result: Agent sees content of complaint.

3.6 Action: Agent clicks on save/export button.
Result: Complaint is downloaded. Consisting of excel document and folder with images.

3.7 Action: Agent clicks on state icon.

Result: State roll down with possible options he could change it to.

3.8 Action: Agent chooses processing (or different word that will symbolize it).
Result: State of complaint is changed.

3.9 Action: Agent clicks on state icon again.
Result: State roll down with possible options he could change it to.

3.10 Action: Agent chooses processed (or different word that will symbolize it).
Result: State of complaint is changed.

3.11 Action: Agent clicks on icon that represents language.
Result: Language options are rolled down.

3.12 Action: Agent chooses option that he wants language to change it to.
Result: Language is changed and everything on website is changed to that language.

3.13 Action: Agent clicks on 'log out'.
Result: Agent is logged out.

4. Statistics [16] <- probably will be updated

Steps:

4.1 Action: Agent or Admin logs into the system and clicks on statistics.

Result: Statistics are showed.

4.2 Action: Admin/Agent clicks on period and chooses what he wants.

Result: Time period for Statistics.

How to work with the language system

1. ADDING A TRANSLATION
   - In HTML, wrap the words/phrases that need to be translated in the {% trans %} template tag:

     ```
     {% trans "your text" %}
     ```

   - In Python files, import the translation function and mark the strings for translation using _(""):

     ```
     from django.utils.translation import gettext as _

     translated_text = _("your text")
     ```

   - To generate the translation files, run the following command(in termila) to extract translatable strings and create/update the .po translation files:

     ```
     python manage.py makemessages -l your_language_code

     python manage.py makemessages -l sk //Example with sk
     ```

   - This will create or update the translation file in:

     ```
     /locale/your_language_code/LC_MESSAGES/django.po

     /locale/sk/LC_MESSAGES/django.po
     ```

   - Inside this file, you'll find entries like this:

     ```
     msgid "your text"

     msgstr ""  # <-- Write the translation here(int the double quotes)
     ```

   - After adding translations, compile them using:

     ```
     python manage.py compilemessages
     ```

2. ADDING A NEW LANGUAGE

- In setting.py and dropdown_options.py add the new language along with its corresponding language code:

  ("your_language_code", "Your Language")

  ("sk", "Slovak") <--- for Example

- Run the following command to create translation files for the new language:

  python manage.py makemessages -l your_language_code

- This will generate a folder for the new language inside the /locale/ directory

  /locale/your_language_code/LC_MESSAGES/django.po

- Then add translations like in 1.

How to work with system (User manual for admin and agent)

Home Page

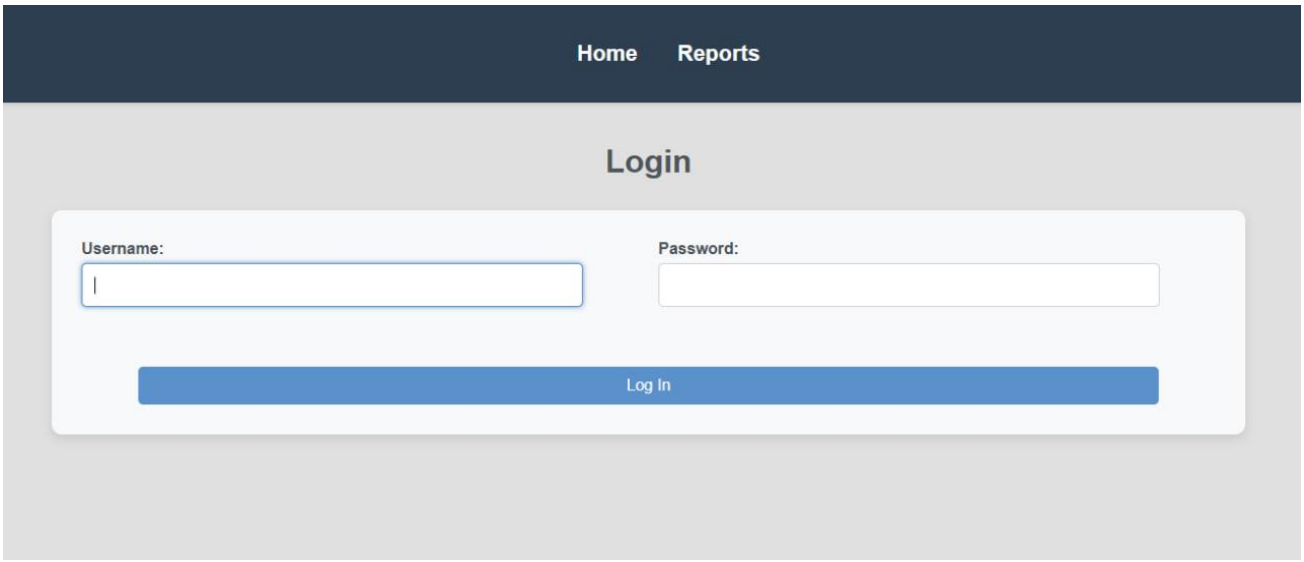
This is a home page that all actors (customer, agent and admin) can see.

All users can change the language to English or Slovak by clicking on the green button on the top right.

Now agent and admin can log in by clicking the red button on the top right.

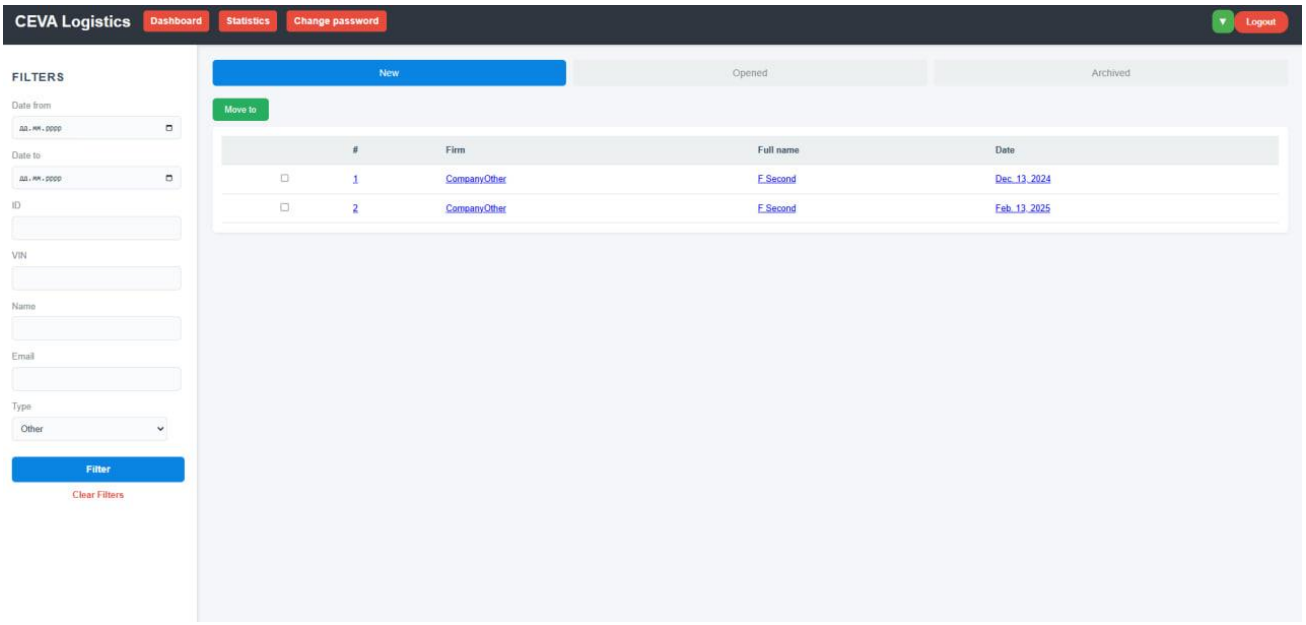
Log in

After clicking on the login button, an authorization with two fields will appear (username and password).

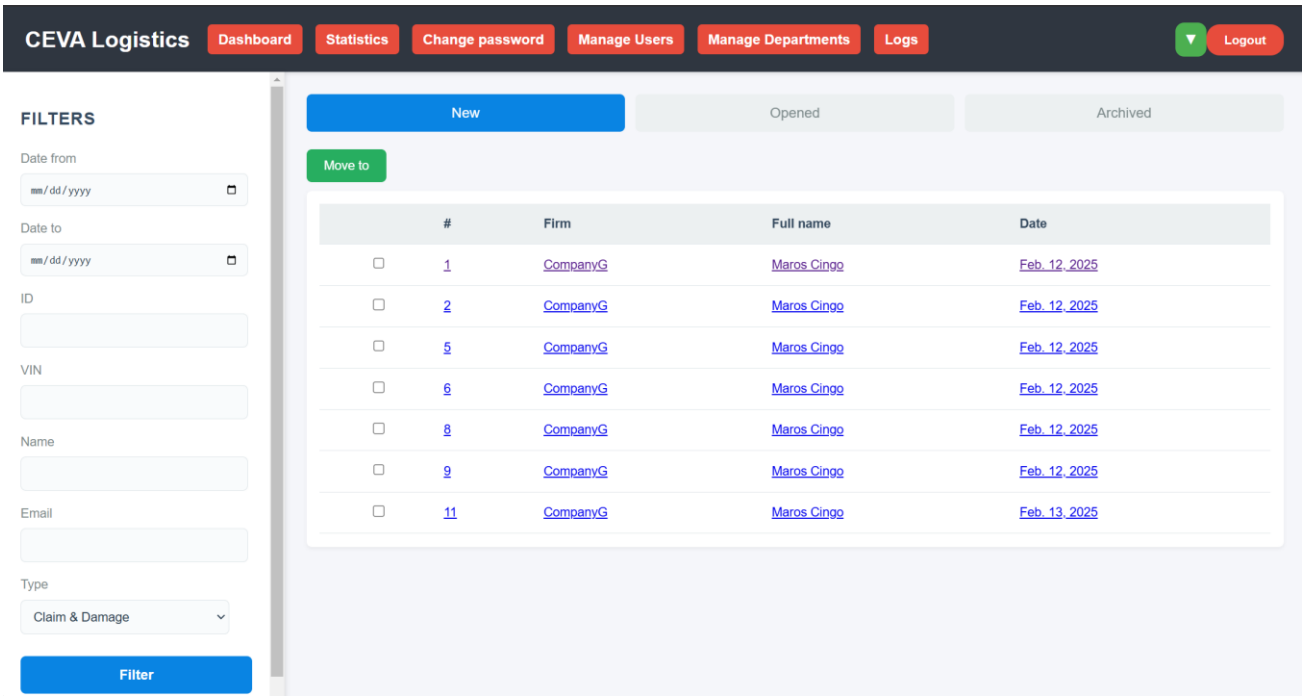


Dashboard

Depending on the user (agent or admin), after a successful login, a different dashboard will appear (1.1 agent, 1.2 admin). User can access dashboard also by clicking on button “dashboard”.



Pic. 1.1 Agent dashboard



Pic. 1.2 Admin dashboard

Features which are available to both agent and admin

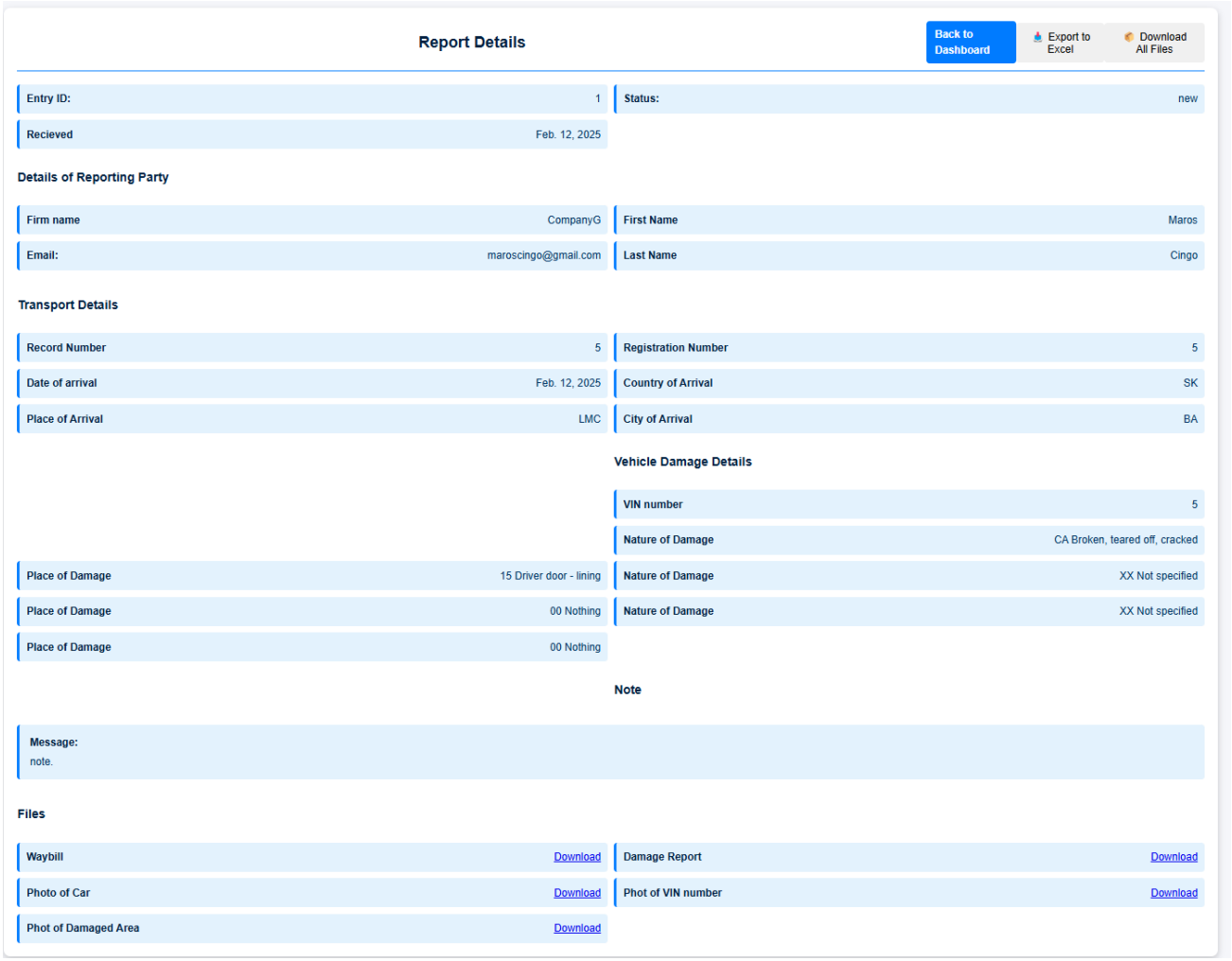
Report Details

If you click on the complaint, a window will open with detailed information as in the following picture.

By clicking on “Back to Dashboard” you get back to the dashboard.

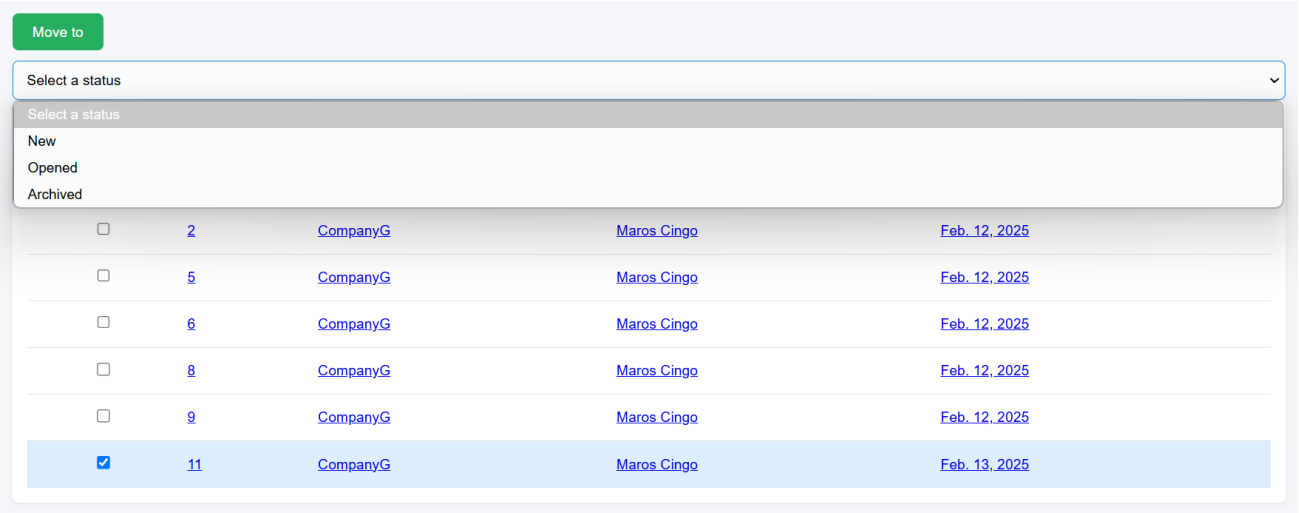
By clicking on “Export to Excel” you can export shown report to .xlsx (Excel).

By clicking on “Download All Files’ you can download all files (most of the time pictures) that have been send with report.



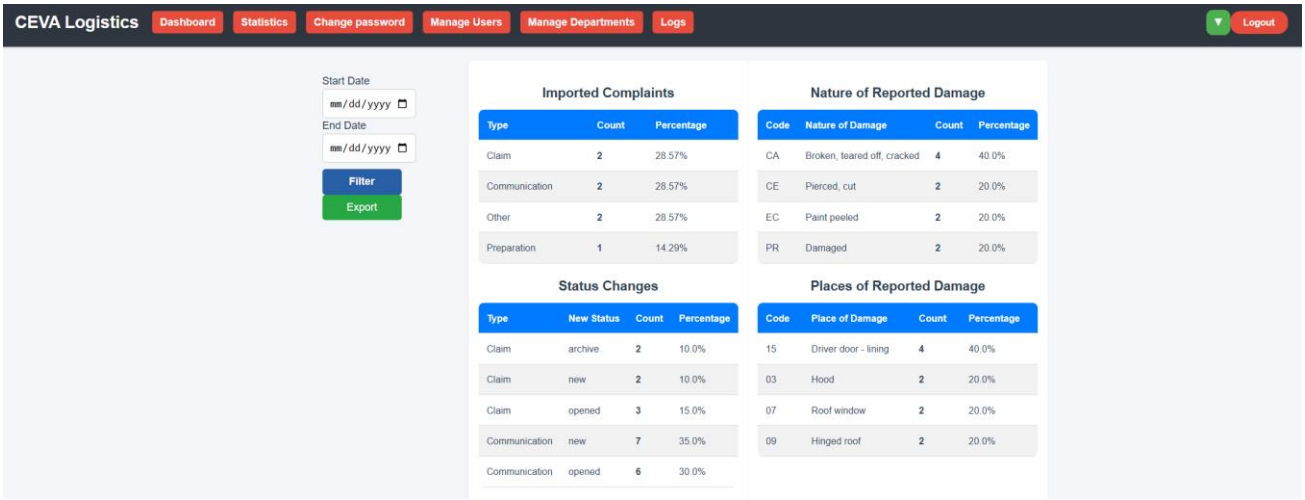
Change of complaint status

After the agent triggers the complaint, the agent can highlight the record with the checkbox and press the green "move it" button. After that, you need to choose where the agent wants to direct the recording. Available categories (New, Open, Archived). After the selected status, to successfully move the record, you need to press the button again.

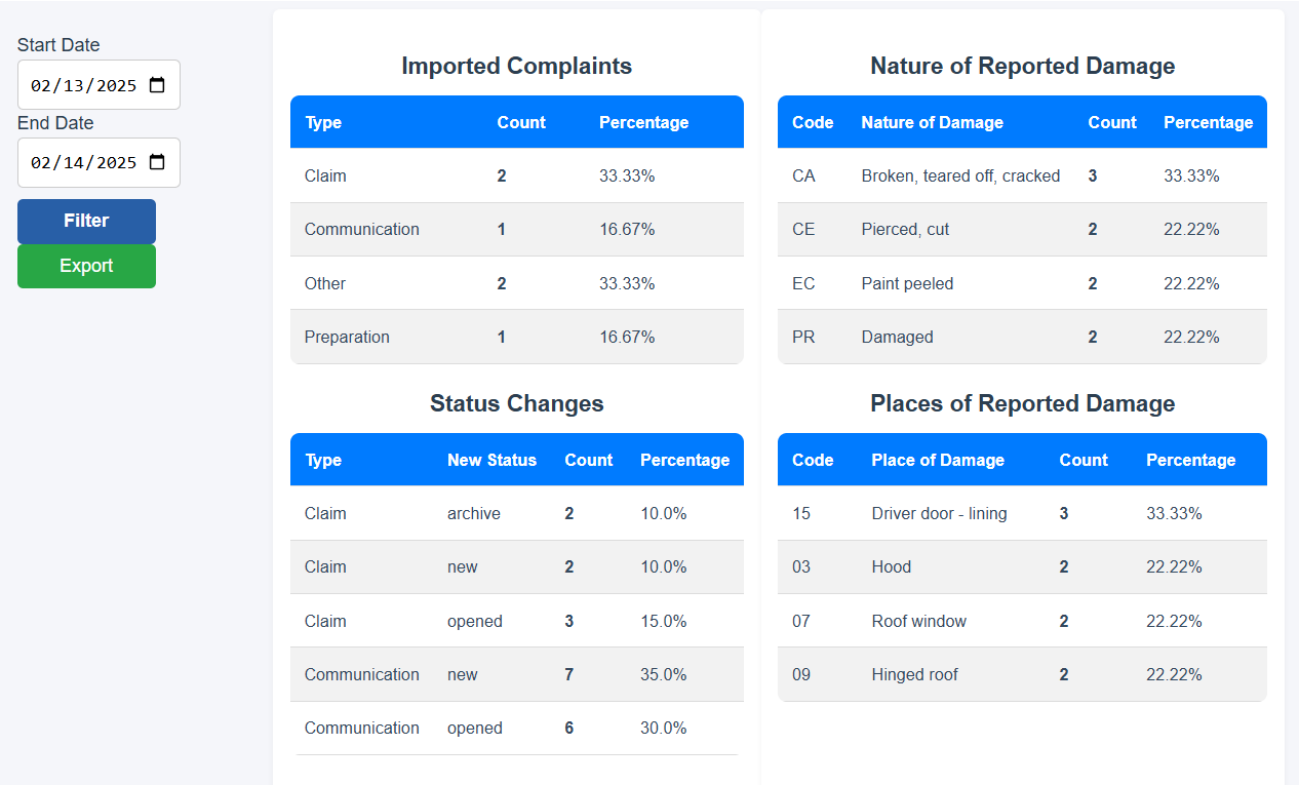


Statistics

The second red button on the top left is responsible for the statistics page. After clicking on the button, the agent will get to the page shown in picture 1.3, after that the agent must select Start Date and End Date (the period for which the agent wants to see statistics), then press the "Filter" button, which will immediately fill in all the available information (statistics) for the selected period, picture 1.4. “Export” button will download you Excel file with statistics.



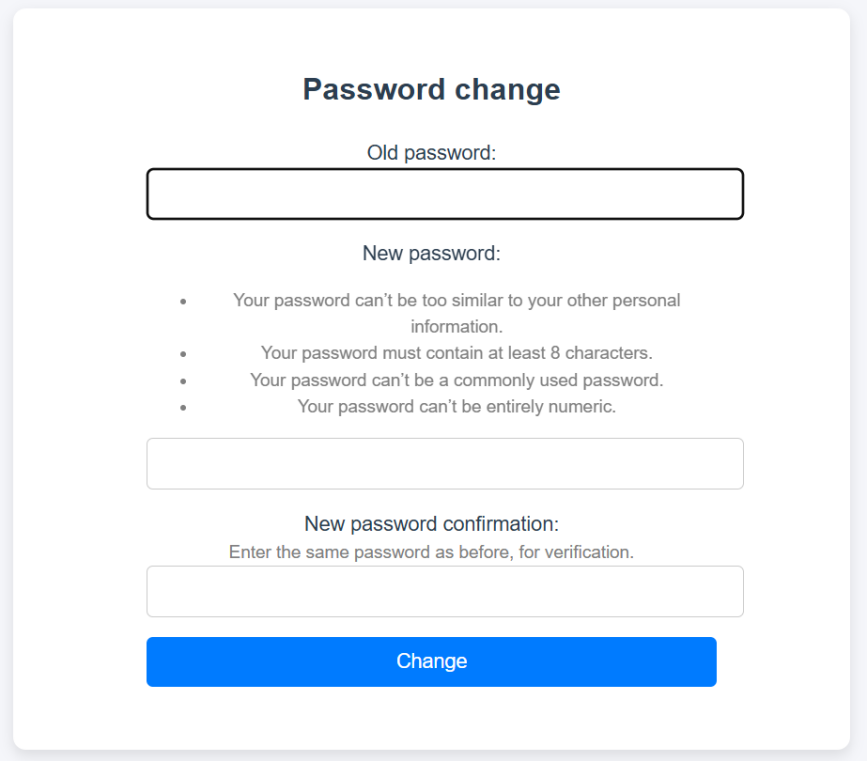
Pic. 1.3 Statistics dashboard



Pic. 1.4 Statistics after pressing the "filter" button

Password change

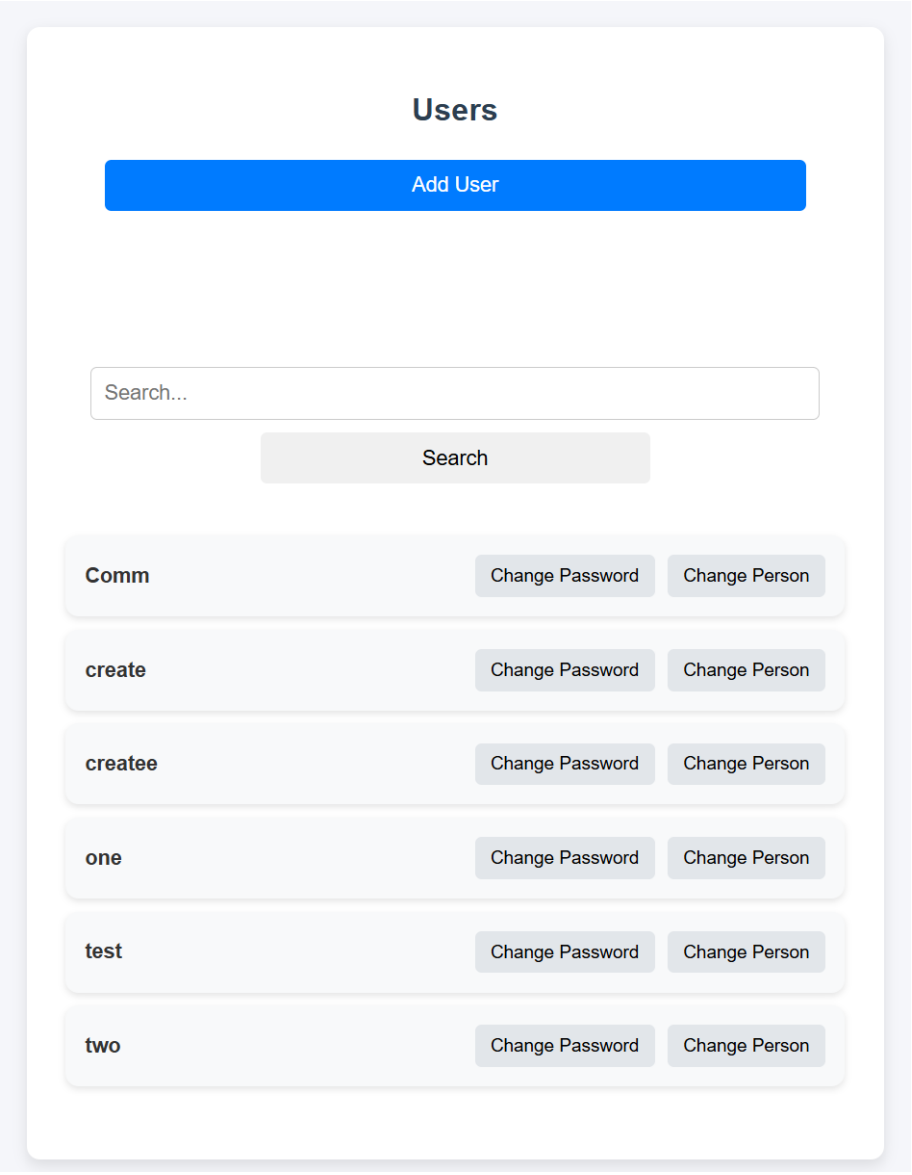
Third red button above "change password" will take you to the password change page shown in the image below. The user will need to enter the old password, create a new one according to the criteria listed and confirm the new password again. After clicking the "change" button, the agent's password will be successfully changed.



Features which are available only for the admin

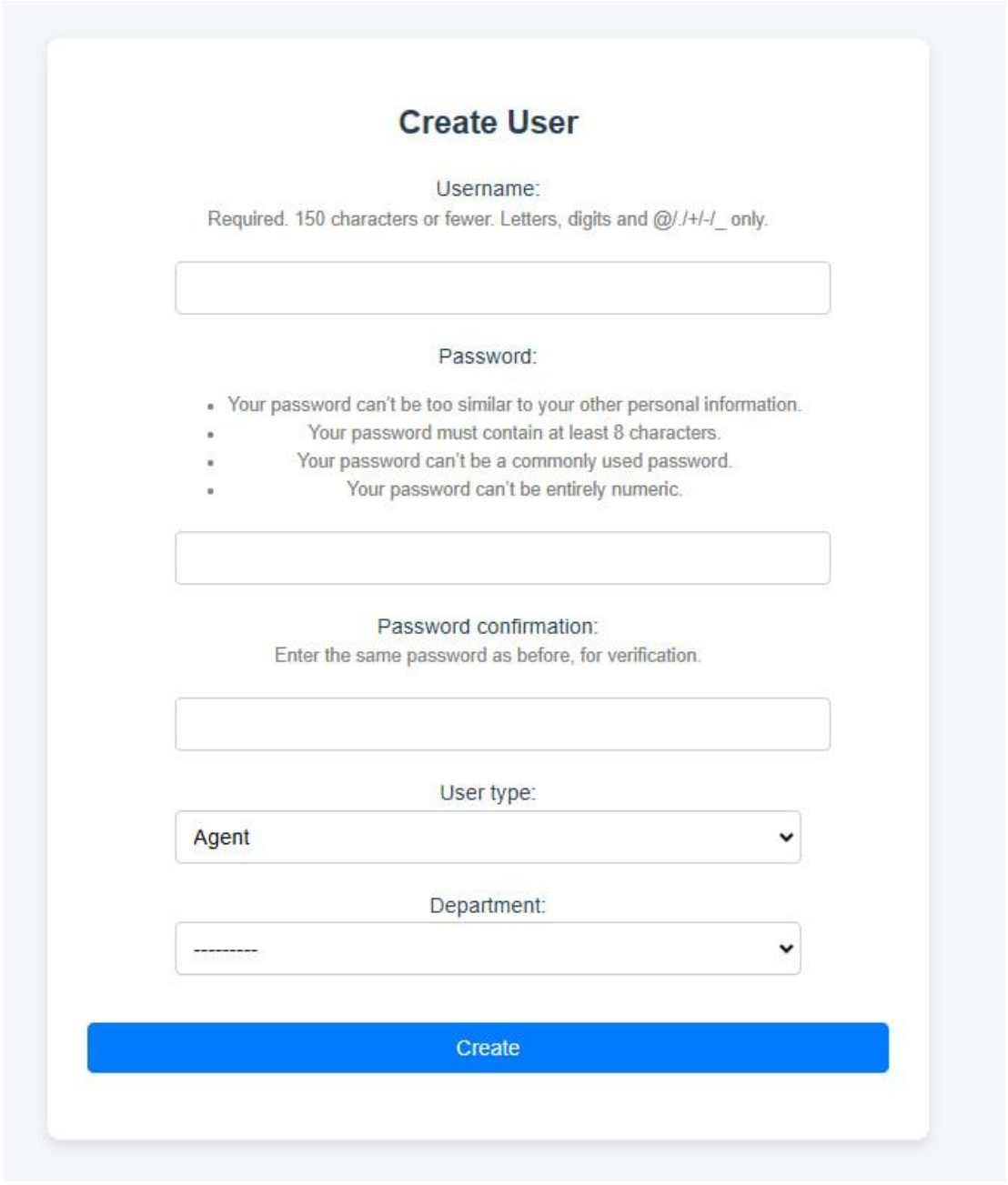
Manage Users

The administrator can click on the "Manage Users" button and it will open the page where the administrator can add a user, find a user, change the user's password, and change the person.



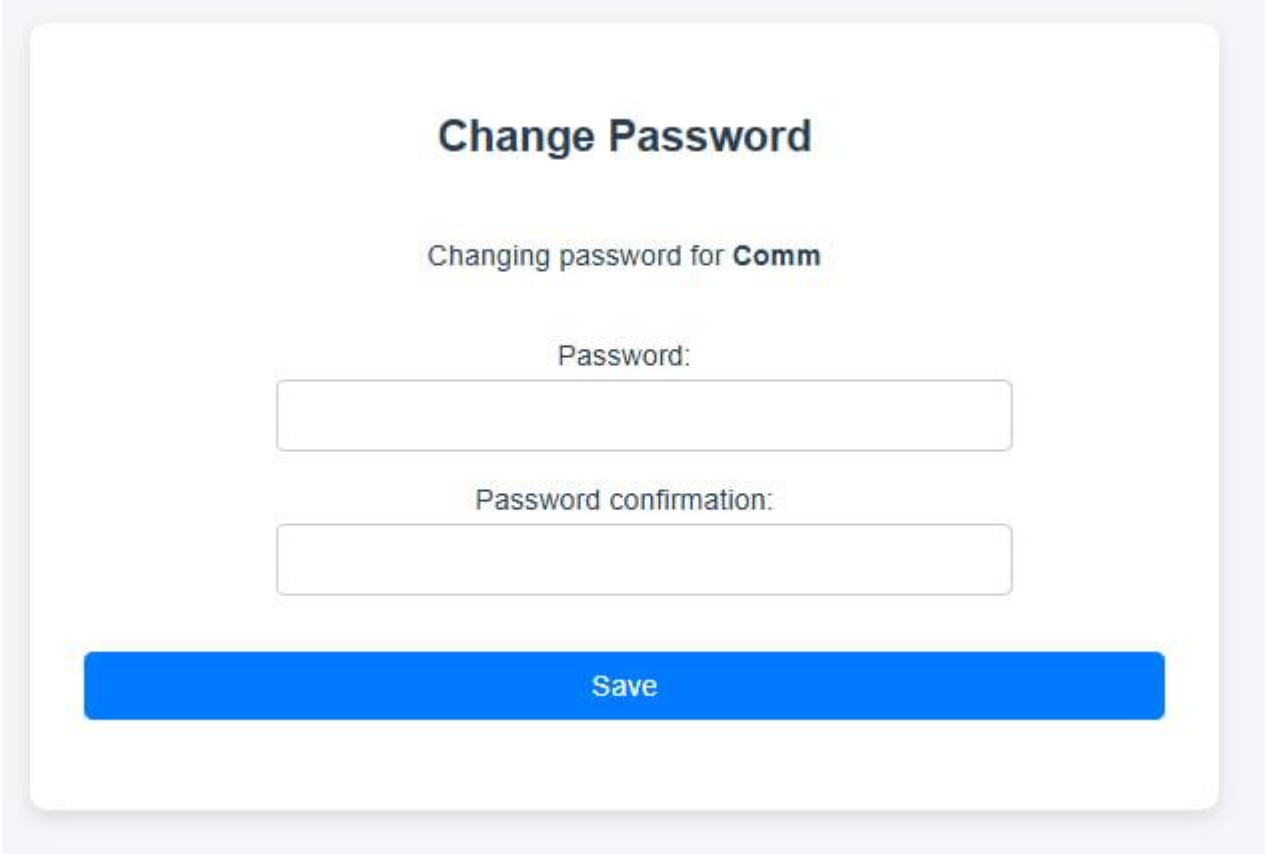
Add User

To register a new user, the administrator can click the "Add User" button. After that, in accordance with all the requirements, fill in the relevant fields: username, password, repeat password, choose the type of user and assign a department to the user. After fulfilling the requirements, the administrator can create a user by clicking the "create" button.



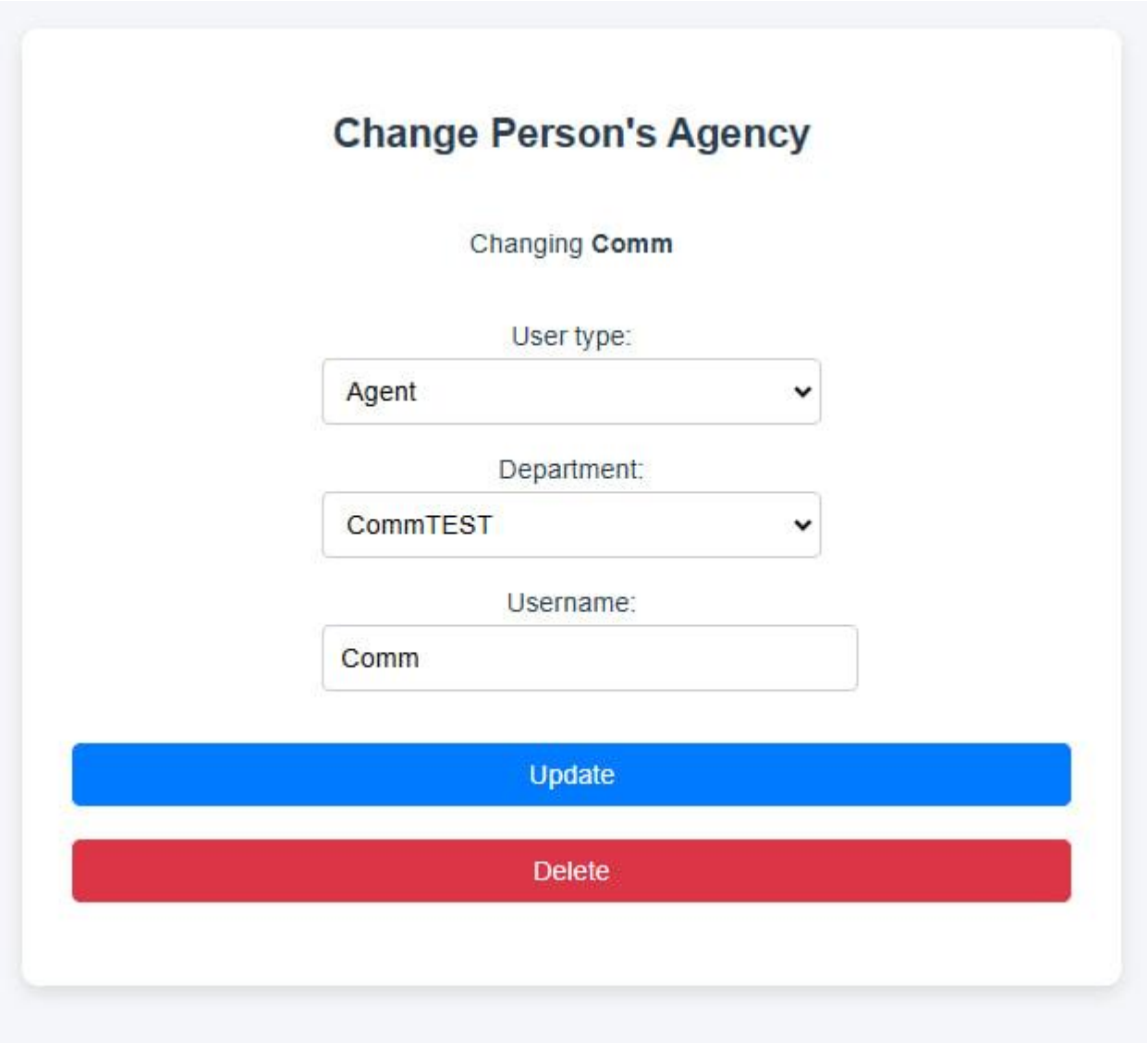
Change another user’s Password

The administrator can also change the password for any user. To do this, you need to find the user in the list whose password you want to change and click on "Change Password". After that, a page for changing the password will open, where you only need to enter a new password and confirm it.



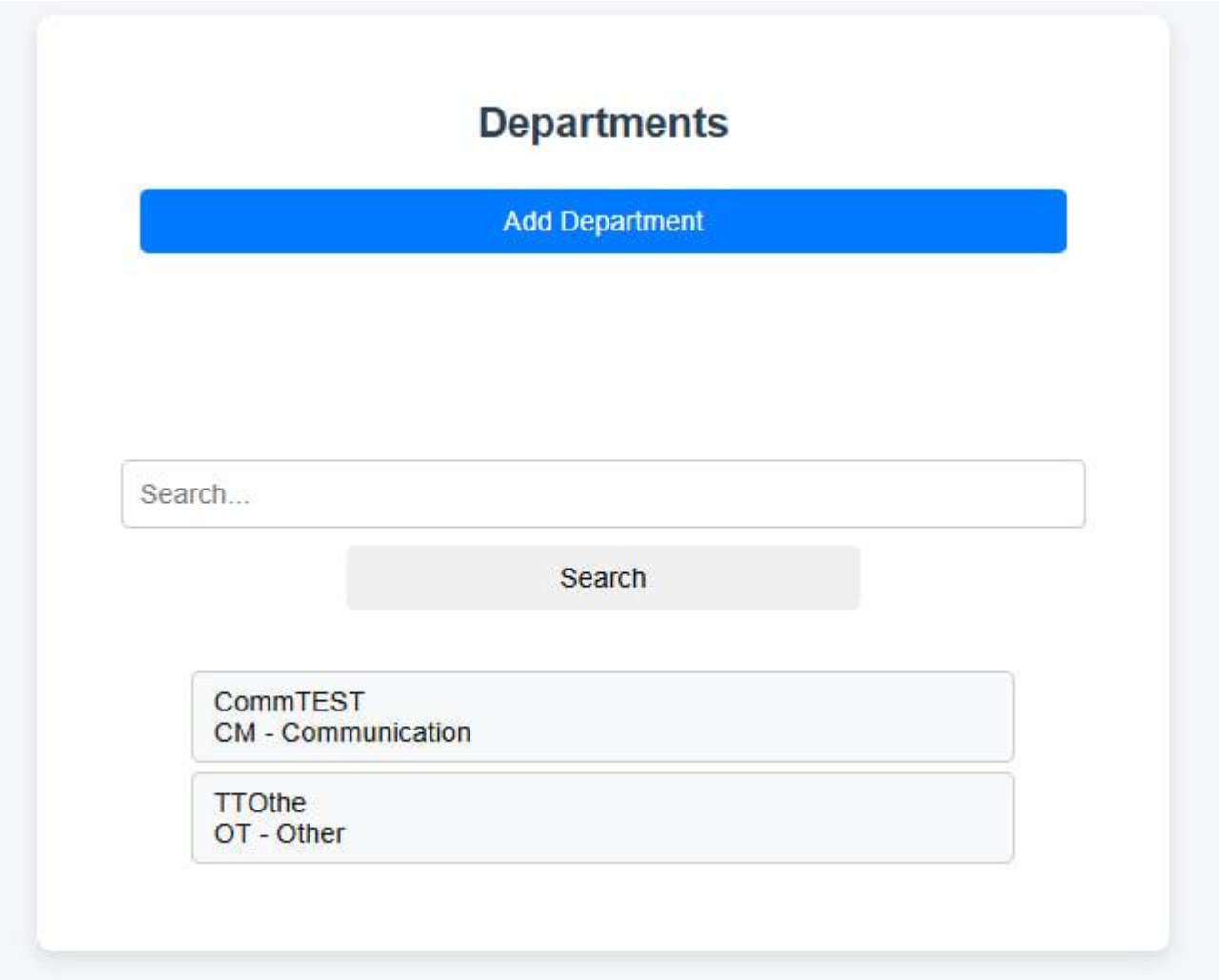
Change Person

The administrator can also change the type of user, his username and the department in which the user is located using the "Change Person" button. After going to the page, the administrator can choose the type and department, as well as change the username manually. After all changes, the "Update" button will save the changes. The administrator can also delete the user simply by using the "Delete" button.



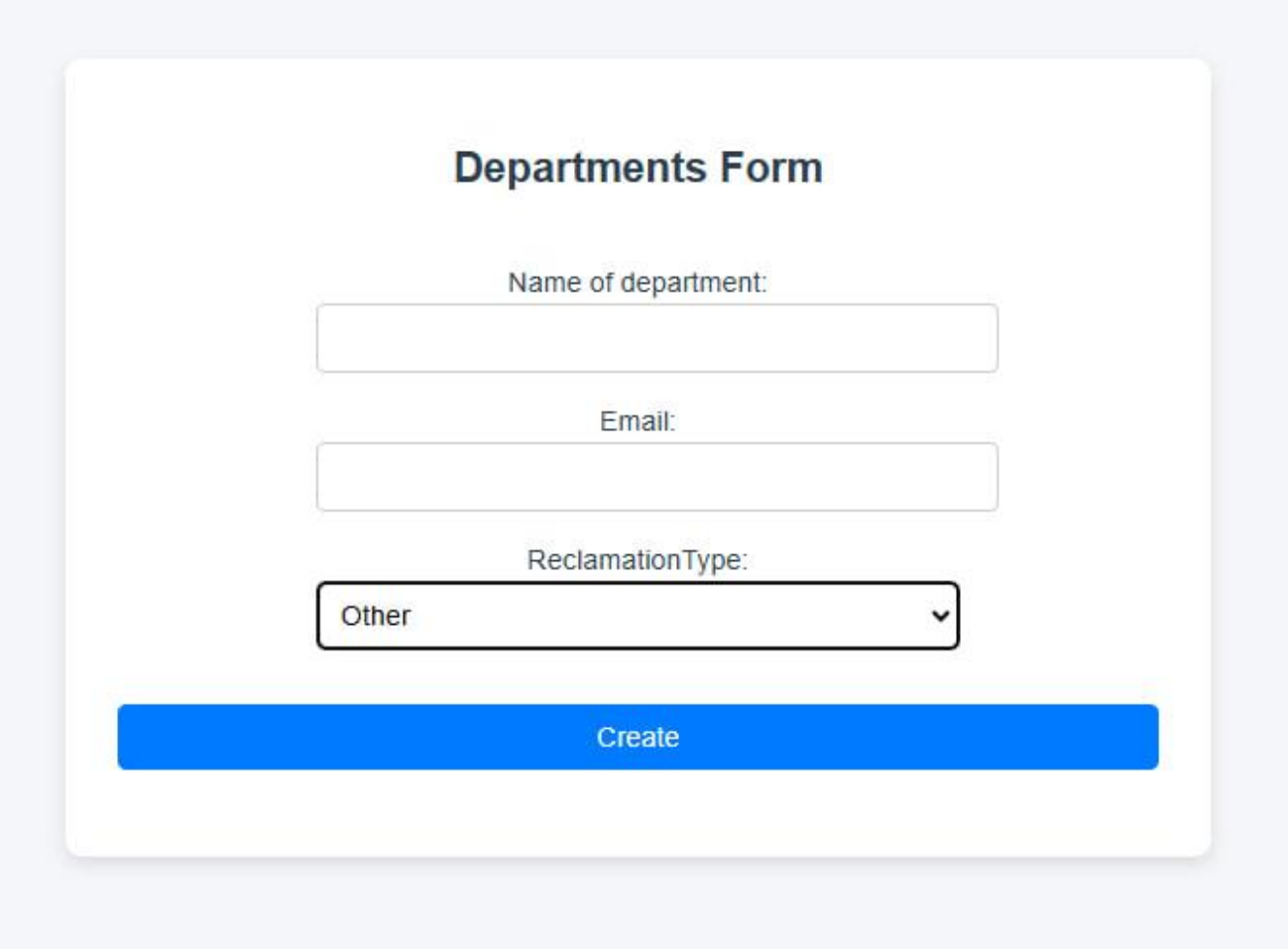
Manage Departments

Admin can also manage departments thanks to "Manage Departments" button. By clicking the button, the administrator will see all departments, will be able to find the desired one in the search and create a new one.



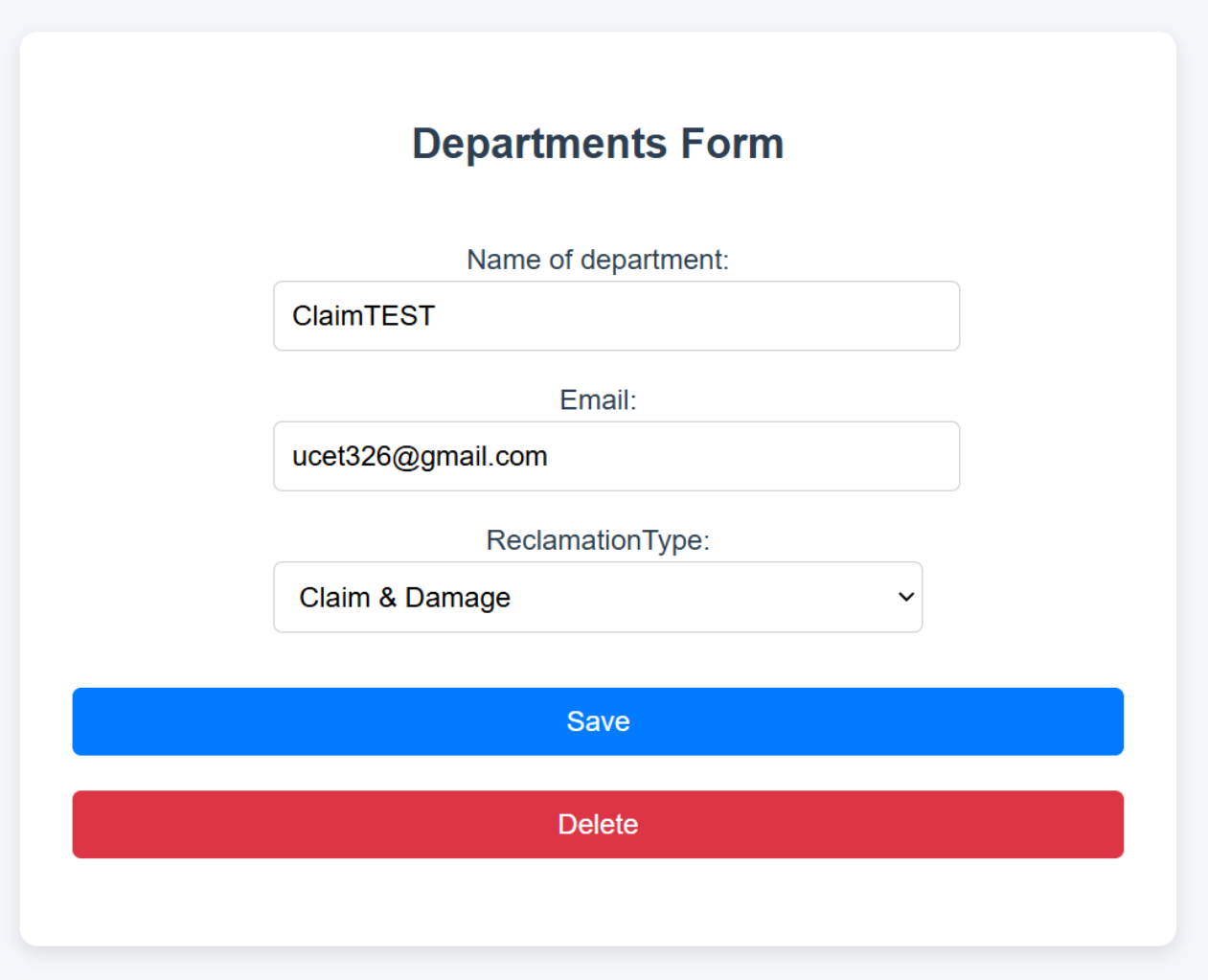
Create Department

To create a new admin department, you need to click the "Add department" button after going to "Manage Departments". After that, you will need to enter the name of the department, email and choose from the drop-down list which complaints the newly created department will be responsible for. All changes must be confirmed with the "Create" button.



Change department

You can change department information by clicking on desired department after going to "Manage Departments". By clicking on “Save” you change information about that department. By clicking on “Delete” you will delete that department.



Logs

You can also track all changes using Logs. By clicking on "Logs” button, you will go to the page with all the logs, which will have a number (id), timestamp, user who performed the action, complaint in relation to which the action was taken, its ID, name of the action, previous status and new one. You can also filter those logs.