2024

SSDLC Analysis and Design – TruckMotion

STEPS SPECIFICATION – VERSION 1.0

NUNO MARMELEIRO 1190922
ROGÉRIO SOUSA 1191017
RAFAEL FAÍSCA 1180658
ÓSCAR FOLHA 1181600
RAFAEL OLIVEIRA 1191611

Index

Index of Figures

Index of Tables

Document Approval

Name	Date	Signature
Rogério Sousa	21/04/2024	Rogério Sousa
Óscar Folha	21/04/2024	Óscar Folha
Rafael Faísca	21/04/2024	Rafael Faísca
Nuno Marmeleiro	21/04/2024	Nuno Marmeleiro
Rafael Oliveira	21/04/2024	Rafael Oliveira

Revision history

Version	Author	Description	Date
1.0	Óscar Folha	Initial version, divide the document and add analysis/requirements statements that already exist in our repository as a .md file.	17-04-2024
2.0	Everyone	Final version	21-04-2024

1 Introduction

1.1 Purpose

This document's intention is to describe and to document an application asked to be done during the curricular unit Desenvolvimento de Software Seguro, DESOFS.

Within the scope of the Secure Software Development (DESOFS) curriculum unit, a project was developed to address the critical challenges associated with efficient truck traffic management, while providing secure and reliable solutions for drivers and customers involved in this process.

This report aims to provide a comprehensive overview of the design and development of the application conceived to explore the security, vulnerability, threat analysis and its impact in the application.

1.2 Document conventions

For the development of this document some conventions were followed, such as:

- IEEE bibliographic citation style.
- For the development of domain models, logical data model and use case diagrams it was used the Unified Modeling Language (UML) notation. [1]

1.3 Project scope

Within the framework of the Secure Software Development (DESOFS) curriculum unit, the project aims to delve into security aspects surrounding the development of an application designed for truck traffic management.

The main topics described in this report are the following:

- Vulnerability Analysis: Conducting in-depth analysis to identify potential vulnerabilities within the application, including but not limited to code vulnerabilities, configuration weaknesses and design flaws.
- Threat Modeling: Developing threat models to systematically identify, prioritize, and mitigate potential threats to the security of the application and its associated data.
- Security Assessment: Performing comprehensive security assessments, including penetration testing, code reviews and security audits, to evaluate the effectiveness of security controls and identify areas for improvement.
- Risk Assessment: Assessing the potential risks posed by identified vulnerabilities and threats, considering their likelihood and potential impact on the confidentiality, integrity, and availability of the application and its data.

- Mitigation Strategies: Developing and implementing mitigation strategies to address identified vulnerabilities and mitigate potential threats, including the application of security best practices, patches and updates.
- Impact Analysis: Evaluating the potential impact of security incidents on the application, its users and associated stakeholders and developing contingency plans to mitigate adverse effects.
- Documentation and Reporting: Documenting findings, analysis and mitigation strategies in comprehensive reports to facilitate understanding, communication and decision-making among stakeholders.

1.4 Business Logic - What is TruckMotion?

TruckMotion is an application designed for package delivery management. Customers request deliveries through the app and managers assign these requests to the most suitable drivers to ensure packages are delivered to the correct destinations.

Customers enter personal information such as their name, email, and birthdate, and can connect multiple locations to their account. They also provide information about the companies they work for. Managers assign service requests to drivers and manage accounts. Drivers receive their assignments from managers, carry out the delivery and provide proof of delivery with a photo of the destination.

When customers request a delivery, they provide details like the service name, total weight of the items and expected delivery date. The status of a service can be active, finished, canceled, or pending, which customers can monitor through the app. Drivers update the status as they transport packages, marking a service as finished upon delivery.

TruckMotion ensures smooth package delivery operations, offering a system for customers to request services, managers to assign tasks and drivers to complete deliveries with proof of arrival.

2 Analysis/Requirements

2.1 Use Cases

There are thirteen use cases identified:

- As a Customer, I want to be able to register delivery services so that I can have the parcel delivered to my Location. I should only register delivery services for myself.
- As a Customer, I want to be able to see the status of my requested Service (Pending/Accepted/Rejected/In progress) so that I can have control of it. I should only check my requested services status.
- As a customer, I want to be able to add Locations associated to me so that I can choose one of them in the Service. I should only be able to add locations to myself.
- As a Customer, I want to be able to see the progress of my accepted Service so that I have full control of it. I should only have control of my accepted services.

- As the Driver, I want to visualize the Transports that have been assigned to me so that I can have control of my work. I should only visualize the transport assigned to me.
- As the Driver, I want to signal the start and end of a Transport that has been provided by me so that I can finish my Service. I should only signal the start of a transport if it is assigned to me, I should only signal the end of a transport that had been assigned to me and started by me and not yet concluded.
- As a driver, I want to be able to prove my delivery with a photo or a screenshot of the maps so that my job execution is accepted. I should only verify my execution if the job is assigned to me and started by me.
- As a manager, I want to be able to approve or reject a job so that I can control the Services. I should only approve and reject jobs if they are registered in the system by customers.
- As a manager, I want to be able to dispatch services to drivers so that the Services are executed. I should only dispatch services if the requests are approved.
- As a manager, I want to be able to register Drivers in the application so that the Transports can be executed by them. I should only be able to register them and not update their data.
- As a manager, I want to be able to register Customers in the application so that I have the possibility of new Services be asked. I should only be able to register them and not update their data.
- As a user of the system, I want to be able to login into the application so that I can access the application features. I should not be able to access the system without valid login credentials.
- As a user of the system, I want to be able to change my password following the correct rules. I should only be able to change my password.
    - This password should be at least 12 characters long, after multiple spaces are combined.
    - This password should not be more than 128 characters.
    - This password should not be truncated, only multiple concatenated spaces should be replaced by a single space.
    - This password should allow any printable Unicode character.
    - It requires me to use my current password and new password to change it.
    - There should be a strength meter.
    - There should be no requirement for upper or lower case or numbers or special characters.
    - There should not be any password history requirement.
    - "Paste" functionality should be allowed.
    - I should be able to see the full length of the password I just typed or only the last character.

All the mentioned Use Cases are illustrated in the Figure 1.

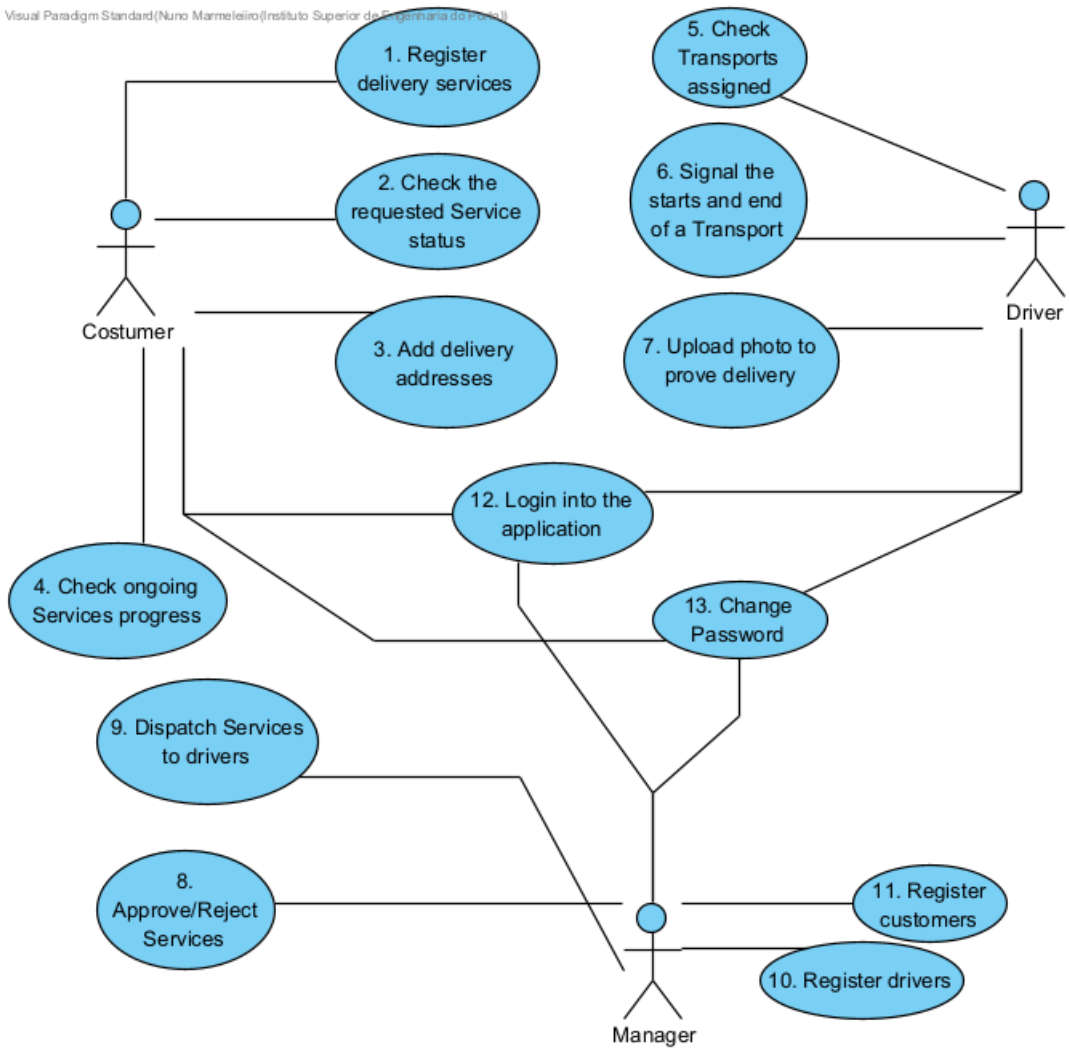

Figure 1 - Use Case Diagram

2.2 Domain-Driven Design

The Domain-Driven Design Diagram is represented in Figure 2.

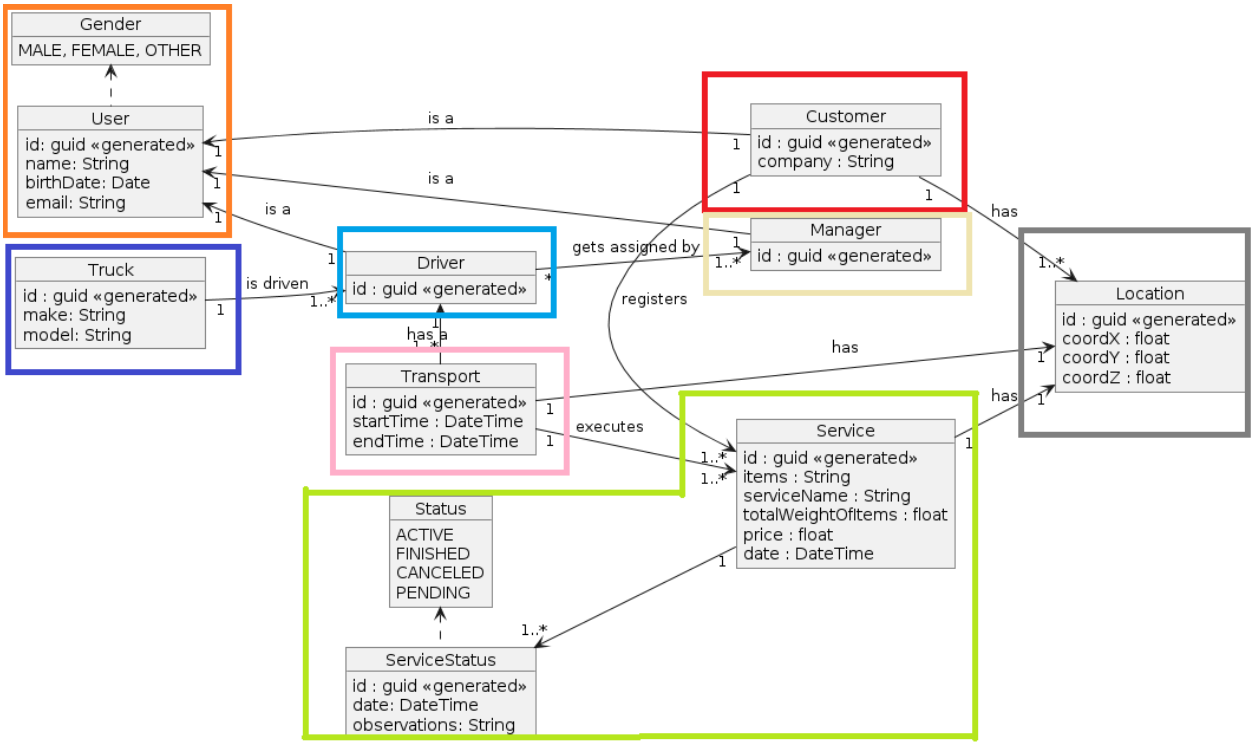

Figure 2 - Domain-Driven Design Diagram

The Manager is responsible for creating Drivers and Customers accounts and generally manages the application. The Customers asks for Services to be delivered at the Location desired and the Manager assigns a Driver with a Truck already assigned to execute the Transport of the Customer's Service. Finalizing the Transport, the Driver updates the Service Status to Finished.

It is visible eight Aggregates:

- User – composed with User entity and Gender Enum.
- Truck – composed with Truck entity.
- Driver - composed with Driver entity.
- Transport - composed of the Transport entity.
- Service - composed of Service and ServiceStatus entities and the Status Enum.
- Location - composed with Location entity.
- Manager – composed with Manager entity.
- Customer - composed with Customer entity.

2.3 Abuse Cases

UC1. As a Customer, I want to be able to register delivery services so that I can have the parcel delivered to my company/address/place.

1. Attackers could manipulate the location data provided during the registration process to redirect deliveries to a different address, potentially their own. This could result in theft or tampering with packages.

i. **CVSS Risk Rating:**
   - 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N

ii. **Justification:**
   - Location spoofing can redirect deliveries to unauthorized addresses, leading to theft or tampering of packages.

iii. **Kind of abuse:**
   - Business

iv. **Countermeasures:**
   - Implement geolocation validation during registration to detect anomalies.
   - Use secure communication protocols to prevent location manipulation.
   - Provide customers with delivery tracking and verification mechanisms.

2. During the registration process, if the communication channel between the customer and the delivery service provider is compromised, attackers could intercept the registration data, modify it, and redirect deliveries to an unauthorized location.

i. **CVSS Risk Rating:**
   - 8.7 (High) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N

ii. **Justification:**
   - Man in the Middle attacks during registration can intercept and modify registration data, leading to unauthorized redirection of deliveries.

iii. **Kind of Abuse:**
   - Business

iv. **Countermeasures:**
   - Implement HTTPS and certificate pinning to secure communication channels.
   - Regularly update security protocols to mitigate known vulnerabilities.
   - Educate users about the risks of using unsecured networks.

3. Attackers may inject malicious scripts into the delivery service registration page, potentially compromising the security of other users' data, redirecting deliveries to unintended locations or trigger denial of service (DoS).

i. **CVSS Risk Rating:**
   - 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N

ii. **Justification:**
   - XSS attacks can compromise the security of delivery service registration pages, potentially redirecting deliveries or compromising user data.

iii. **Kind of Abuse:**
   - Technical and Business

iv. **Countermeasures:**

- Implement input sanitization to prevent XSS vulnerabilities.
- Employ web application firewalls (WAFs) to detect and block malicious scripts.
- Regularly update web application frameworks and libraries.

4. Attackers may flood the delivery service provider's registration system with a high volume of fake registration requests, causing service disruption or preventing legitimate customers from registering delivery services.

   i. **CVSS Risk Rating:**

   - 5.9 (Medium)CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H

   ii. **Justification:**

   - DoS attacks can disrupt delivery service registration systems, preventing legitimate customers from registering or modifying delivery services.

   iii. **Kind of Abuse:**

   - Technical

   iv. **Countermeasures:**

   - Implement rate limiting and traffic filtering to mitigate DoS attacks.
   - Use scalable infrastructure to handle increased traffic loads.
   - Employ DDoS mitigation services for additional protection.

5. Attackers may attempt to gain unauthorized access to the delivery service provider's database or systems to retrieve sensitive information about past deliveries, including customer addresses and delivery contents, for malicious purposes such as identity theft or targeted theft of valuable goods.

   i. **CVSS Risk Rating:**

   - 8.7 (High) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N

   ii. **Justification:**

   - Unauthorized access to delivery records can lead to identity theft or targeted theft of valuable goods by revealing sensitive information about past deliveries.

   iii. **Kind of Abuse:**

   - Business

   iv. **Countermeasures:**

   - Implement strict access controls and encryption for delivery records.
   - Regularly monitor access logs for suspicious activity.
   - Comply with data protection regulations to safeguard customer information.

UC2. As a Customer, I want to be able to see the status of my requested Transport (Pending/Accepted/Rejected/In progress) so that I can have control of it.

1. As an attacker, I bypass access control checks by modifying the URL, internal application state, or the HTML page, or simply using a custom API attack tool so that I can access the privilege information about the transport status.
    i. **CVSS V3 risk rating:**
        - 5.7 CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:N/A:N
    ii. **Kind of Abuse:**
        - Business
    iii. **Countermeasures:**
        - Add verification for authorization inside the data access.
2. As an attacker and/or malicious user, I can manipulate the primary key so that I can access the transport status of other users.
    i. **CVSS V3 risk rating:**
        - 5.7 CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:N/A:N
    ii. **Kind of Abuse:**
        - Business
    iii. **Countermeasures:**
        - Add verification for authorization so that only verified users access the data defined for them.
3. As an attacker, I manipulate sessions, access tokens, or other access controls in the application to act as a user without being logged in, so that I can get access to the information about the transport status.
    i. **CVSS V3 risk rating:**
        - 5.3 CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N;
    ii. **Kind of Abuse:**
        - Business
    iii. **Countermeasures:**
        - Make session and access tokens secret and hardly inaccessible for outside users. Always check users' tokens for data, not only their role.
4. As an attacker, I force browsing to transport status page without being authenticated, gaining access to privilege the information about the transport status.
    i. **CVSS V3 risk rating:**
        - 5.3 CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N
    ii. **Kind of Abuse:**
        - Business
    iii. **Countermeasures:**

- Always verify authentication inside the transport status page.

UC3. As a customer, I want to be able to add Locations associated to me so that I can choose one of them in the Service.

1. As an attacker, I modify the details of existing locations associated to customers, such as changing addresses or altering coordinates, leading to incorrect information for the drivers who will conduct the Transport.
   i. **CVSS Risk Rating:**
      - 8.1 (High) CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N
   ii. **Justification:**
      - Data Manipulation can result in customer services being stolen or not being successfully delivered, that will cause constraints in the Driver's work and Customer's life.
   iii. **Kind of Abuse:**
      - Business
   iv. **Countermeasures:**
      - If any information about Customer's locations is changed, a notification is sent. This way, the Customer will know and will have time to report the situation.
      - When delivering the order, a code must be shown to the driver to validate if it is the correct person to deliver.
2. As an attacker, I insert malicious sentences causing Injection or SQL queries into input areas, potentially leading to data breaches or system compromise.
   i. **CVSS Risk Rating:**
      - 9.0 (Critical) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H
   ii. **Justification:**
      - Injection can result in many things, such as manipulation or visualization of anything in the Database by SQL Injection. It can make the application execute unintended commands compromising the entire system.
   iii. **Kind of Abuse:**
      - Technical and Business.
   iv. **Countermeasures:**
      - It must be validations to any input done by Users to check if it is a valid one.
3. As an attacker, I use automated scripts to create many locations to multiple customer accounts simultaneously, potentially overloading the system or causing performance issues.
   i. **CVSS Risk Rating:**
      - 5.9 (Medium)CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H

ii. **Justification:**

- Making the system unavailable will cause constraints to the Customers that might be expecting deliveries and won't get them because Driver will not have access to the destiny. Also, won't be able to ask for more Delivers.

iii. **Kind of Abuse:**

- Technical

iv. **Countermeasures:**

- Implement rate limiting and traffic filtering to mitigate DoS attacks.
- Use redundant systems to maintain service availability.
- Employ DDoS mitigation services as a proactive measure.

4. As an attacker, I gain unauthorized access to the software and add locations to a customer's account without their consent, potentially causing confusion or inconvenience to the customer.

i. **CVSS Risk Rating:**

- 8.1 (High) CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N

ii. **Justification:**

- Gaining access to adding Locations to a Customer's account, besides causing confusion and inconveniences to him, might as well provide access to the Attacker to other confidential information or systems functionalities.

iii. **Kind of Abuse:**

- Business

iv. **Countermeasures:**

- Implement strict access controls and encryption for user's login information.
- Regularly monitor access logs for suspicious activity.
- Comply with data protection regulations to safeguard customer information.

UC4. As a Customer, I want to be able to see the progress of my accepted Service so that I have full control of it.

1. As an Attacker, I manipulate the system providing false or misleading live location updates for the transport associated with their service, causing confusion or leading to incorrect decision-making by the customer.

i. **CVSS Risk Rating:**

- 5.4 (Medium) CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:C/C:L/I:L/A:N

ii. **Justification:**

- Providing false information about live locations of Transports can mislead the customer into making wrong decisions.

iii. **Kind of Abuse:**

- Business

iv. **Countermeasures:**

- Automatically, notify the customer of an estimated time to get his delivery and when it is about 1 hour or 30 minutes away notify as well. This way, Customer will know that live location is incorrect.

2. As an attacker, I get unauthorized access to the system and intercept the live location of the transport, compromising the privacy and security of the service and endangering the transport.

i. **CVSS Risk Rating:**

- 8.5 (High) CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:L/A:N

ii. **Justification:**

- Having access to the live location of a Transport can result in stolen merchandise or breaking the confidentiality of the Customer's delivery. It can also lead to Ransomware or Extortion situations for the Customer or owners of the application.

iii. **Kind of Abuse:**

- Business

iv. **Countermeasures:**

- Notify the customer whenever his live locations deliveries are asked.
- Restrict the live location of User's deliveries that are logged by asking for a code that only the Customer knows, for example sending this code in SMS for his phone.

3. As an attacker, I flood the system with a high volume of requests for live location updates, overwhelming the system's resources and causing it to become unavailable for legitimate customers who want to track their services' progress.

i. **CVSS Risk Rating:**

- 5.9 (Medium)CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H

ii. **Justification:**

- Making the system unavailable will cause constraints to the Customers because they won't be able to see how much time their deliveries will take.

iii. **Kind of Abuse:**

- Technical

iv. **Countermeasures:**

- Implement rate limiting and traffic filtering to mitigate DoS attacks.

- Use redundant systems to maintain service availability.
- Employ DDoS mitigation services as a proactive measure.

4. As an attacker, I send fraudulent messages or emails to customers, with malicious websites or applications designed to steal their credentials or exploit their devices, under the guise of providing access to live location updates.

   i. **CVSS Risk Rating:**

      - 8.3 (High) CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N

   ii. **Justification:**

      - Sending malicious information to the Customer can influence him to access it and allow the Attacker to have his Login Credentials compromising his account. If this Customer have privilege accesses to other Components, all these components will be compromised as well. Besides the Login credentials, it can give access for the attacker to the application depending on the Malicious malware.

   iii. **Kind of Abuse:**

      - Business

   iv. **Countermeasures:**

      - Specify how the information will be given to the customer for him to know if something is wrong.
      - Previously, educated users about identifying phishing attempts.

UC5. As the Driver, I want to visualize the Transports that have been assigned to me so that I can have control of my work.

1. As an attacker, I bypass access control checks by modifying the URL, internal application state, or the HTML page, or simply using a custom API attack tool so that I can access the privilege information about the transports that have been assigned.

   i. **CVSS V3 risk rating:**

      - 5.7 CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:N/A:N

   ii. **Kind of Abuse:**

      - Business

   iii. **Countermeasures:**

      - Add verification for authorization inside the data access.

2. As an attacker and/or malicious user, I can manipulate the primary key so that I can access the transport services assigned for other drivers.

   i. **CVSS V3 risk rating:**

      - 5.7 CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:H/I:N/A:N

   ii. **Kind of Abuse:**

      - Business

   iii. **Countermeasures:**

- Add verification for authorization so that only verified users access the data defined for them.

3. As an attacker, I manipulate sessions, access tokens, or other access controls in the application to act as a user without being logged in, so that I can get access to the information about all the drivers transport jobs assigned.
   i. **CVSS V3 risk rating:**
      - 5.3 CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N
   ii. **Kind of Abuse:**
      - Business
   iii. **Countermeasures:**
      - Make session and access tokens secret and hardly inaccessible for outside users.
      - Always check users' tokens for data, not only their role.
4. As an attacker, I force browsing to transport status page without being authenticated, gaining access to privilege the information about the transport jobs assigned to the drivers.
   i. **CVSS V3 risk rating:**
      - 5.3 CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:N/A:N
   ii. **Kind of Abuse:**
      - Business
   iii. **Countermeasres:**
      - Always verify authentication inside the transport jobs page.

UC6. As the Driver, I want to signal the start and end of a Transport that has been provided to me so that I can finish my Service.

1. The driver signals the start of a transport service, but instead of starting the service, they deceive the system by activating the signal without intending to provide the service.
   i. **CVSS Risk Rating:**
      - 5.7 (Medium) CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:N/I:H/A:N
   ii. **Justification:**
      - This could be done to manipulate the system for personal gain, such as receiving compensation for services not rendered or to create a false record of activity.
   iii. **Kind of Abuse:**
      - Business
   iv. **Countermeasures:**

- This could be done to manipulate the system for personal gain, such as receiving compensation for services not rendered or to create a false record of activity.

2. An attacker impersonates a driver to fake signaling.
   i. **CVSS Risk Rating:**
      - 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N
   ii. **Justification:**
      - Impersonation can lead to unauthorized signaling, creating problems for deliveries.
   iii. **Kind of Abuse:**
      - Business
   iv. **Countermeasures:**
      - Implement strong authentication mechanisms for drivers.
      - Conduct regular verification checks for all drivers.
3. An attacker/driver spoofs the location of his truck, making it seem like they are in a different location than they are. This could lead to misallocation of resources or fraudulent activity.
   i. **CVSS Risk Rating:**
      - 6.2 (Medium) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:L/I:H/A:N
   ii. **Justification:**
      - Spoofed location data can result in incorrect resource allocation or fraudulent activities.
   iii. **Kind of Abuse:**
      - Business
   iv. **Countermeasures:**
      - Use secure geolocation services with validation checks.
      - Employ GPS tracking with tamper-proof mechanisms.
      - Educate drivers on the importance of verifying their assigned locations.
4. Attackers flood the signaling system with a high volume of fake signals requests, causing service disruption.
   i. **CVSS Risk Rating:**
      - 5.9 (Medium)CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H
   ii. **Justification:**
      - DoS attacks can disrupt the signaling system, preventing the manager from knowing when a transport started/ended.
   iii. **Kind of Abuse:**

- Technical

iv. **Countermeasures:**

- Implement rate limiting and traffic filtering to mitigate DoS attacks.
- Use redundant systems to maintain service availability.
- Employ DDoS mitigation services as a proactive measure.

UC7. As a driver, I want to be able to prove my delivery with a photo or a screenshot of the maps so that my job execution is accepted.

1. The driver could use pre-existing photos from their gallery or the internet and pass them off as proof of delivery or instead of genuinely delivering the package or completing the service, they may provide a photo taken at a different location.

   i. **CVSS Risk Rating:**

   - 5.7 (Medium) CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:N/I:H/A:N

   ii. **Justification:**

   - By fabricating proof of delivery, the driver can save time that would have been spent on completing the delivery. This time can then be used for personal activities or for taking on additional delivery jobs, thereby increasing potential earnings.

   iii. **Kind of Abuse:**

   - Business

   iv. **Countermeasures:**

   - Providing training and awareness programs for drivers about the importance of accurate reporting and the consequences of providing false proof of delivery can deter misconduct.
   - Implementing strict disciplinary measures and consequences for drivers caught providing false proof of delivery.
   - Implementing geofencing technology can help enforce delivery boundaries, ensuring that deliveries are made within designated areas.

2. An attacker impersonates a driver to fake photo sending.

   i. **CVSS Risk Rating:**

   - 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N

   ii. **Justification:**

   - Impersonation can lead to unauthorized signaling, creating problems for deliveries.

   iii. **Kind of abuse:**

   - Business

   iv. **Countermeasures:**

   - Implement strong authentication mechanisms for drivers.

- Conduct regular verification checks for all drivers.

3. Attackers flood the photo sending system with a high volume of photo requests, causing service disruption.
    i. **CVSS Risk Rating:**
        - 5.9 (Medium)CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H
    ii. **Justification:**
        - DoS attacks can disrupt the signaling system, preventing the manager from knowing when a transport started/ended.
    iii. **Kind of Abuse:**
        - Technical
    iv. **Countermeasures:**
        - Implement rate limiting and traffic filtering to mitigate DoS attacks.
        - Use redundant systems to maintain service availability.
        - Employ DDoS mitigation services as a proactive measure.

UC8. As a manager, I want to be able to approve or reject a job so that I can control the services.

1. An attacker gains access to the approval system and forges approvals for jobs that were not actually reviewed by the manager. This could result in unauthorized services being provided or resources being allocated improperly.
    i. **CVSS Risk Rating:**
        - 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N
    ii. **Justification:**
        - Forged approvals can result in unauthorized services being provided or resources being allocated improperly.
    iii. **Kind of Abuse:**
        - Business
    iv. **Countermeasures:**
        - Implement approval workflows with multiple layers of verification.
        - Provide training on recognizing fraudulent requests.
        - Implement auditing mechanisms to detect suspicious activity.
2. Attackers flood the approval system with a high volume of fake job approval requests, causing service disruption or preventing legitimate job approvals from being processed.
    i. **CVSS Risk Rating:**
        - 5.9 (Medium) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:H
    ii. **Justification:**

- DoS attacks can disrupt the job approval system, preventing legitimate approvals from being processed.

iii. **Kind of Abuse:**

- Technical

iv. **Countermeasures:**

- Implement rate limiting and traffic filtering to mitigate DoS attacks.
- Use redundant systems to maintain service availability.
- Employ DDoS mitigation services as a proactive measure.

3. Malware infects the manager's device, allowing attackers to manipulate job approvals or steal sensitive information related to job details or customers.

i. **CVSS Risk Rating:**

- 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N

ii. **Justification:**

- Malware infections can compromise the manager's device, allowing attackers to manipulate job approvals or steal sensitive information.

iii. **Kind of Abuse:**

- Business

iv. **Countermeasures:**

- Implement endpoint security solutions.
- Conduct regular malware scans and updates.
- Provide training on recognizing and avoiding malware threats.

4. Attackers intercept communication between the manager and the approval system, allowing them to manipulate job approval requests or intercept sensitive information.

i. **CVSS Risk Rating:**

- 8.7 (High) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N

ii. **Justification:**

- MitM attacks can intercept and manipulate job approval requests, leading to unauthorized changes in service delivery.

iii. **Kind of Abuse:**

- Business

iv. **Countermeasures:**

- Implement encryption protocols like TLS/SSL to secure communication channels.
- Use digital signatures to verify the integrity of approval requests.
- Educate users about the risks of unsecured networks.

5. Attackers exploit vulnerabilities in the approval system's database to manipulate job approval records or gain unauthorized access to sensitive information.
    i. **CVSS Risk Rating:**
        - 9.0 (Critical) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H
    ii. **Justification:**
        - SQL injection attacks can manipulate job approval records or gain unauthorized access to sensitive information.
    iii. **Kind of Abuse:**
        - Business
    iv. **Countermeasures:**
        - Implement input validation and parameterized queries to prevent SQL injection.
        - Conduct regular security assessments of the approval system.
        - Enforce strict database access controls.

UC9. As a manager, I want to be able to dispatch services to drivers so that the Services are executed. I should only dispatch services if the requests are approved

1. An attacker gains unauthorized access to the dispatch system and dispatches fake or malicious services to drivers.
    i. **CVSS Risk Rating:**
        - 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N
    ii. **Justification:**
        - Unauthorized access can lead to the dispatch of fake services, causing confusion and potentially leading to wasted resources.
    iii. **Kind of Abuse:**
        - Business
    iv. **Countermeasures:**
        - Implement strict access controls with multi-factor authentication.
        - Regularly monitor access logs for suspicious activity.
        - Conduct security training to educate employees on the risks of unauthorized access.
2. An attacker floods the dispatch system with bogus service requests, overwhelming it and preventing legitimate services from being dispatched.
    i. **CVSS Risk Rating:**
        - 8.7 (High) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:N/I:H/A:H
    ii. **Justification:**
        - DoS attacks can disrupt operations and prevent critical services from being dispatched.

iii. **Kind of Abuse:**

- Technical

iv. **Countermeasures:**

- Implement rate limiting and traffic filtering to mitigate DoS attacks.
- Use redundant systems to maintain service availability.
- Employ DDoS mitigation services as a proactive measure.

3. An attacker impersonates a manager or dispatcher to dispatch services, potentially causing confusion or miscommunication among drivers.

i. **CVSS Risk Rating:**

- 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N

ii. **Justification:**

- Impersonation can lead to unauthorized dispatches and miscommunication among drivers.

iii. **Kind of Abuse:**

- Business

iv. **Countermeasures:**

- Implement strong authentication mechanisms for dispatchers.
- Conduct regular verification checks for dispatched services.
- Educate drivers on verifying the authenticity of dispatch messages.

4. An attacker spoofs the location data of services to drivers, making it seem like they are in a different location than they are. This could lead to misallocation of resources or fraudulent activity.

i. **CVSS Risk Rating:**

- 6.2 (Medium) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:L/I:H/A:N

ii. **Justification:**

- Spoofed location data can result in incorrect resource allocation or fraudulent activities.

iii. **Kind of Abuse:**

- Business

iv. **Countermeasures:**

- Use secure geolocation services with validation checks.
- Employ GPS tracking with tamper-proof mechanisms.
- Educate drivers on the importance of verifying their assigned locations.

5. An attacker intercepts service dispatch messages and modifies them to change service details, such as pickup/delivery locations or cargo information, leading to confusion or potential theft.

i. **CVSS Risk Rating:**
   - 8.2 (High) CVSS:3.0/AV:N/AC:H/PR:L/UI:N/S:C/C:H/I:H/A:N

ii. **Justification:**
   - Modified service details can lead to confusion, delays, or potential theft of goods.

iii. **Kind of Abuse:**
   - Business

iv. **Countermeasures:**
   - Implement end-to-end encryption for dispatch messages.
   - Use digital signatures to ensure message integrity.
   - Conduct regular audits and verification checks for dispatched services.

6. A disgruntled employee with access to the dispatch system intentionally disrupts operations by canceling valid services or sending drivers on unnecessary routes.
   i. **CVSS Risk Rating:**
      - 8.1 (High) CVSS:3.0/AV:N/AC:L/PR:H/UI:R/S:C/C:H/I:H/A:N

   ii. **Justification:**
      - Insider threats can lead to operational disruptions and financial losses.

   iii. **Kind of Abuse:**
      - Business

   iv. **Countermeasures:**
      - Implement least privilege access controls.
      - Monitor employee activities and behavior.
      - Provide channels for reporting suspicious behavior.

7. An attacker intercepts communication between the manager and drivers, potentially gaining access to sensitive information or altering instructions.
   i. **CVSS Risk Rating:**
      - 7.7 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:N

   ii. **Justification:**
      - MitM attacks can lead to unauthorized access to sensitive information or alteration of critical instructions.

   iii. **Kind of Abuse:**
      - Business

   iv. **Countermeasures:**
      - Implement encryption protocols like TLS/SSL for communication.
      - Use digital signatures to verify message integrity.

- Educate users on detecting and reporting suspicious activities.

UC 10. As a manager, I want to be able to register Drivers in the application so that the Transports can be executed by them.

1. An attacker registers fake drivers in the application, possibly using stolen identities, to gain access to the system.
   i. **CVSS Risk Rating:**
      - 7.7 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:N
   ii. **Justification:**
      - Fake driver registrations can lead to unauthorized access and misuse of the system.
   iii. **Kind of Abuse:**
      - Business.
   iv. **Countermeasures:**
      - Implement identity verification checks for driver registration.
      - Use CAPTCHA or similar mechanisms to prevent automated registrations.
      - Conduct regular audits of registered drivers for anomalies.
2. An attacker injects malicious SQL queries into the registration form, gaining unauthorized access to the database or causing data loss.
   i. **CVSS Risk Rating:**
      - 7.7 (Critical) CVSS:3.0/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:N
   ii. **Justification:**
      - SQL injection can lead to unauthorized access to sensitive data and compromise the integrity of the system.
   iii. **Kind of Abuse:**
      - Technical and Business.
   iv. **Countermeasures:**
      - Implement input validation and parameterized queries to prevent SQL injection.
      - Conduct regular security audits of the registration system.
      - Use secure coding practices to mitigate injection vulnerabilities.
3. A malicious insider or hacker gains access to sensitive driver information stored in the system, such as personal details or driving history.
   i. **CVSS Risk Rating:**
      - 7.7 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:N
   ii. **Justification:**

- Unauthorized access to driver information can lead to privacy violations and misuse of personal data.

iii. **Kind of Abuse:**

- Business.

iv. **Countermeasures:**

- Implement strict access controls and least privilege principles.
- Encrypt sensitive driver information at rest and in transit.
- Monitor access logs for suspicious activity and unauthorized access attempts.

4. An attacker floods the registration system with automated requests, overwhelming it and preventing legitimate drivers from registering.

i. **CVSS Risk Rating:**

- 9.0 (Critical) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H

ii. **Justification:**

- DoS attacks can disrupt the registration process, preventing legitimate drivers from accessing the system.

iii. **Kind of Abuse:**

- Technical.

iv. **Countermeasures:**

- Implement rate limiting and captcha mechanisms to mitigate DoS attacks.
- Use scalable infrastructure to handle increased registration traffic.
- Employ intrusion detection systems to detect and block suspicious activities.

5. An attacker uses automated scripts to try known username and password combinations (obtained from previous data breaches) to gain unauthorized access to driver accounts.

i. **CVSS Risk Rating:**

- 6.1 (Medium) CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:N/A:N

ii. **Justification:**

- Credential stuffing exploits weak passwords and can lead to unauthorized access to driver accounts.

iii. **Kind of Abuse:**

- Business.

iv. **Countermeasures:**

- Enforce strong password policies for driver accounts.
- Implement multi-factor authentication to mitigate credential stuffing.
- Educate drivers on creating and using strong, unique passwords.

6. Weak authentication mechanisms allow an attacker to easily bypass the login process and gain access to driver accounts.
    i. **CVSS Risk Rating:**
        - 7.7 (High) CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:N
    ii. **Justification:**
        - Insecure authentication can lead to unauthorized access and compromise the security of driver accounts.
    iii. **Kind of Abuse:**
        - Business.
    iv. **Countermeasures:**
        - Implement strong authentication protocols such as OAuth 2.0 or OpenID Connect.
        - Use secure token-based authentication for API access.
        - Conduct regular security assessments to identify and address authentication vulnerabilities.
7. Hackers gain access to the database by storing driver information, compromising sensitive data such as driver licenses, insurance details, or personal addresses.
    i. **CVSS Risk Rating:**
        - 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N
    ii. **Justification:**
        - Data breaches can lead to severe privacy violations and financial losses for both drivers and the organization.
    iii. **Kind of Abuse:**
        - Business.
    iv. **Countermeasures:**
        - Encrypt sensitive data at rest and in transit.
        - Implement intrusion detection and prevention systems.
        - Comply with data protection regulations such as GDPR or CCPA.
8. An attacker gains access to a legitimate driver's account by stealing their credentials through phishing or other means, and then misuses this access for malicious purposes.
    i. **CVSS Risk Rating:**
        - 7.7 (High) CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:C/C:H/I:H/A:N
    ii. **Justification:**
        - Account takeovers can lead to fraudulent activities and misuse of driver accounts.
    iii. **Kind of Abuse:**

- Business.

iv. **Countermeasures:**

- Educate drivers on phishing awareness and safe accounting practices.
- Implement anomaly detection for unusual account activities.
- Provide a mechanism for drivers to report suspicious account access.

UC11. As a manager, I want to be able to register Customers in the application so that I have the possibility of new Services be asked.

1. An attacker registers fake customers in the application to place fraudulent service requests or obtain sensitive information.

   i. **CVSS Risk Rating:**

   - 7.7 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:N

   ii. **Justification:**

   - Fake customer registrations can lead to fraudulent transactions and misuse of the system.

   iii. **Kind of Abuse:**

   - Technical and Business.

   iv. **Countermeasures:**

   - Implement identity verification checks for customer registration.
   - Use CAPTCHA or similar mechanisms to prevent automated registrations.
   - Conduct regular audits of registered customers for anomalies.

2. A rogue employee or unauthorized user accesses and misuses customer data for personal gain or malicious purposes.

   i. **CVSS Risk Rating:**

   - 7.7 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:N

   ii. **Justification:**

   - Privacy violations can lead to legal repercussions and loss of trust from customers.

   iii. **Kind of Abuse:**

   - Business.

   iv. **Countermeasures:**

   - Implement strict access controls and data encryption for customer data.
   - Conduct background checks on employees with access to sensitive data.
   - Monitor access logs for unauthorized access attempts.

3. An attacker alters customer details in the system, such as contact information or billing details, leading to confusion or financial loss.

   i. **CVSS Risk Rating:**

- 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N

ii. **Justification:**

- Altered customer details can lead to service disruptions and financial harm.

iii. **Kind of Abuse:**

- Business.

iv. **Countermeasures:**

- Implement data integrity checks and validation rules.
- Encrypt sensitive customer information to prevent tampering.
- Provide customers with the ability to review and verify their information.

4. An attacker takes over a legitimate customer account, changes their contact information, and places orders under their name, leading to confusion and potential financial harm.

i. **CVSS Risk Rating:**

- 7.3 (High) CVSS:3.0/AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:N

ii. **Justification:**

- Account hijacking can result in fraudulent transactions and damage to the customer's reputation.

iii. **Kind of Abuse:**

- Business.

iv. **Countermeasures:**

- Implement multi-factor authentication for customer accounts.
- Notify customers of any changes to their account information.
- Provide a mechanism for customers to report suspicious account activities.

5. If the application has APIs for customer registration, an attacker could abuse these APIs to register fake customers in bulk, overwhelming the system and causing disruption.

i. **CVSS Risk Rating:**

- 9.0 (Critical) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H

ii. **Justification:**

- API abuse can lead to system overload and service disruption.

iii. **Kind of Abuse:**

- Technical.

iv. **Countermeasures:**

- Implement API rate limiting and authentication.
- Use API keys with proper access controls.

- Monitor API usage for unusual patterns and behavior.

UC12. As a user of the system, I want to be able to login into the application so that I can access the application features.

1. An attacker attempts to gain unauthorized access to user accounts by repeatedly guessing usernames and passwords. This could be done manually or using automated scripts or tools to systematically try different combinations until a valid login is found with the purpose of getting sensitive information.
    i. **CVSS Risk Rating:**
        - 7.5 (High) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:L/A:N
    ii. **Justification:**
        - Brute force attacks can lead to unauthorized access to user accounts, potentially compromising sensitive information.
    iii. **Kind of Abuse:**
        - Business.
    iv. **Countermeasures:**
        - Implement account lockout mechanisms after a certain number of failed login attempts.
        - Enforce strong password policies.
        - Employ CAPTCHA or multi-factor authentication to mitigate brute force attacks.
2. Attackers use previously leaked username/password pairs from other breaches and attempt to log in with those credentials on your application with the purpose of stealing sensitive information. Since users often reuse passwords across multiple services, this can lead to successful unauthorized access.
    i. **CVSS Risk Rating:**
        - 6.9 (Medium) CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:L/A:N
    ii. **Justification:**
        - Credential stuffing exploits users' tendencies to reuse passwords across multiple services, potentially granting unauthorized access to accounts.
    iii. **Kind of Abuse:**
        - Business.
    iv. **Countermeasures:**
        - Encourage users to use unique passwords for each service.
        - Implement multi-factor authentication.
        - Regularly monitor for suspicious login attempts or patterns.
3. Attackers send deceptive emails or messages pretending to be from the application, tricking users into providing their login credentials on a fake login page with the purpose

of getting sensitive information. Once the user submits their credentials, the attacker can use them to access the legitimate application.

i. **CVSS Risk Rating:**
   - 8.6 (High) CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N

ii. **Justification:**
   - Phishing attacks can deceive users into providing their login credentials, compromising their accounts and potentially leading to further exploitation.

iii. **Kind of Abuse:**
   - Business.

iv. **Countermeasures:**
   - Educate users about identifying phishing attempts.
   - Implement email filtering to detect and block phishing emails.
   - Use domain validation techniques to verify legitimate login pages.

4. Attackers intercept or steal a valid session token or cookie after a user successfully logs in. With the stolen session token, the attacker can then impersonate the logged-in user without needing to know their credentials.

i. **CVSS Risk Rating:**
   - 8.5 (High) CVSS:3.0/AV:N/AC:H/PR:N/UI:R/S:C/C:H/I:N/A:N

ii. **Justification:**
   - Session hijacking can result in unauthorized access to user accounts and sensitive information without requiring knowledge of login credentials.

iii. **Kind of Abuse:**
   - Business.

iv. **Countermeasures:**
   - Implement secure session management techniques such as using HTTPS.
   - Use secure cookies with HttpOnly and Secure flags.
   - Regularly rotating session tokens.

5. Attackers intercept communication between the user and the application during the login process. They can eavesdrop on the login credentials being transmitted, potentially capturing sensitive information like usernames and passwords.

i. **CVSS Risk Rating:**
   - 8.7 (High) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:N

ii. **Justification:**
   - MitM attacks during login can intercept sensitive information like usernames and passwords, leading to unauthorized access to accounts.

iii. **Kind of Abuse:**

- Business.

iv. **Countermeasures:**

- Use encryption protocols like TLS/SSL to secure communication channels.
- Implement certificate pinning.
- Educate users about connecting to trusted networks.

6. If the application displays database content retrieved via SQL injection in user-accessible pages without proper sanitization, it can lead to XSS vulnerabilities Attackers can inject malicious scripts into the database, which are then executed in the context of other users' sessions when the data is displayed in the application.

i. **CVSS Risk Rating:**

- 9.0 (Critical) CVSS:3.0/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H

ii. **Justification:**

- XSS via SQL injection can lead to the execution of malicious scripts in the context of other users' sessions, compromising the security of the application and user data.

iii. **Kind of Abuse:**

- Business.

iv. **Countermeasures:**

- Implement input validation and parameterized queries to prevent SQL injection.
- Sanitize user input to mitigate XSS vulnerabilities.
- Regularly patch and update application frameworks and libraries.

2.4 Functional Security Requirements

2.4.1 Authentication and Authorization

- Users (customers, managers, drivers) must authenticate before accessing the system.
- Customers should have the authority to create Services.
- Managers should have the authority to assign Services to drivers.
- Drivers should only be able to access Services assigned to them.
- No default passwords in use for the application framework or any components used by the application.
- Usage of JWT Token for Authentication managed by Server-Side Code.
- Invalidate Token when user Logouts.
- Whenever the JWT Token expires, disable every transaction or data modification.
- The password policy should follow:

1. Must be at least 12 characters long, after multiple spaces are combined.

2. Must not be more than 128 characters.

3. Must not be truncated, only multiple concatenated spaces should be replaced by a single space.

4. Must allow any printable Unicode character such as emojis.

5. Must not be any requirement for upper or lower case or numbers or special characters.

6. "Paste" functionality should be allowed.

- The e-mail/SMS push-up notification should not contain any sensitive information.
- Any password generated (such as for drivers and customers) should be at least 6 characters long, may contain letters and numbers and expire after a short period of time. It should be securely randomly generated.
- No Secrets Questions used.
- Credential recovery must not reveal the current password.
- No default accounts should be present (admin, root or sa).
- Any authentication factor replaced or changed should be notified.
- Lookup secrets should not be predictable.

2.4.2 Data Encryption

- Users' passwords will be encrypted during transmission and storage.
- Encryption should handle errors and problems in a way that does not enable Padding Oracle attacks.
- AES 256 will be the encryption mechanism used since it is industry proven and even approved by the NSA.
- All configuration related to the encryption will use the latest advice.
- The system will allow easy configuration of any configuration related to the encryption (random number, encryption or hashing algorithms, key lengths, rounds, ciphers or modes).
- The system won't use repeated nonces per encryption key.
- The system will verify if the encrypted data is authenticated via signatures and other authenticated cipher modes, or HMAC to ensure that ciphertext is not altered by an unauthorized party.
- The system won't use "short-circuit" operations to avoid leaking information.
- All generated GUIDs will be created with the GUID v4 algorithm and with a Cryptographically secure Pseudo-random Number Generator (CSPRNG) to avoid predictability.

2.4.3 Secure Communication

- Secure communication channels (HTTPS) should be used between customers, managers, drivers, and the server to prevent data interception.

2.4.4 Access Control

- Role-based access control (RBAC) should be implemented to restrict access to specific features based on user roles.
- Managers should have access to administrative functions such as assigning requests, drivers should only have access to request details and navigation features and customers should access the create service page.
- Every performed request must be identified with the Logged in Users' Token.

2.4.5 Audit Trails

- Log and monitor user activities to create an audit trail.
- The system should log login attempts, access to sensitive data, and any actions performed by users.

2.5 Non-functional Security Requirements

2.5.1 Performance

- Security measures should not significantly impact system performance.
- Encryption and decryption processes should be optimized to minimize latency.

2.5.2 Scalability

- The system should be designed to handle an increasing number of users, requests, and data without compromising security.
- Scalability should be considered in terms of both hardware infrastructure and software architecture.

2.5.3 Availability

- The system should be highly available to ensure users can access it whenever needed.
- Redundancy and failover mechanisms should be in place to mitigate the impact of potential downtime due to security incidents or hardware failures.

2.5.4 Reliability

- Security mechanisms should be reliable and consistently enforce access controls, encryption, and other security measures.
- Regular security testing and updates should be performed to address vulnerabilities and maintain system reliability.

2.5.5 Compliance

- Ensure compliance with relevant security standards and regulations like GDPR.
- Implement features such as data anonymization and user consent management to comply with privacy regulations.

2.6 Secure Development Requirements

2.6.1 Secure Design

- Conduct threat modeling to identify potential security threats and vulnerabilities early in the design phase.
- Design the application with security in mind, following security best practices and principles such as the principle of least privilege and defense-in-depth.

2.6.2 Secure Coding Practices

- Follow secure coding guidelines and standards such as OWASP Top 10 and CWE/SANS Top 25.
- Use secure coding practices to prevent common vulnerabilities such as injection attacks (SQL injection, XSS), insecure direct object references, and buffer overflows.
- Do not use unsupported, insecure or deprecate client-side technologies.
- Avoid the usage of eval() or other similar dynamic coding.
- Code Integrity and malicious code search will be analyzed by tools such as Sonarqube and Dependency track. If possible, with integration of quality gates.
- ZAP to test the application security.

2.6.3 Input Validation

- Implement input validation to prevent injection attacks and ensure that user input is properly sanitized and validated before processing.
- Validate input data types, length, format, and range to mitigate the risk of malicious input.
- Data images should be served by their octet stream.
- Protection against OS Command Injection.
- Verify that sign, range, and input validation techniques are used to prevent integer overflows.

2.6.4 Output Encoding

- Encode output data to prevent cross-site scripting (XSS) attacks.

- Use output encoding techniques such as HTML entity encoding or escaping to neutralize user-controlled data before rendering it in the browser.
- Output encoding preserves the user's chosen character set and locale.
- Guarantee that output encoding is not vulnerable against injection attacks.

2.6.5 Secure Configuration

- Configure the application and underlying components securely, following vendor recommendations and security best practices.
- Disable unnecessary services, ports, and functionalities to reduce the attack surface.
- Never reveal Session Tokens in URL parameters.
- Never use customer-side secrets, such as symmetric keys, passwords or API tokens to protect or access sensitive data.
- The application will have CI/CD automation, automated configuration management and automated deployment scripts.
- The application will have the compiler flags so that all available buffer overflow protections and warnings are active.
- The server configuration will be hardened to follow the recommendations of the application server and frameworks in use.
- The application will be able to be re-deployed using automated deployment scripts.
- Only Admin level users will be able to verify the integrity of all security-relevant configurations to detect tampering.
- A dependency checker will be used during build or compilation time to see if all components are up to date.
- Clean-up of unnecessary features, documentation and configuration will be done.
- All application assets will be hosted externally on a content delivery network or external provider.
- All third-party components will come from pre-defined, trusted and continually maintained repositories.
- All third-party libraries in use will be exposed only for the required behavior of the application to reduce attack surface.
- Debug mode of the application will be disabled on the web application to avoid possible security disclosures and debug features.
- System components information will be removed from the HTTP headers/response.
- HTTP response will contain a Content-Type header.
- All API responses will have a Content-Disposition: attachment; filename="api.json" header.
- A Content Security Policy (CSP) will exist to avoid possible XSS attacks.
- All API responses will have a X-Content-Type-Options: nosniff header.
- A Strict-Transport-Header will be used in all responses and for all subdomains.
- A Referrer-Policy header will be used to avoid exposing sensitive information in the URL to untrusted parties.
- Origin header will not be used for authentication or access control decisions, since it can be easily changed by an attacker.
- CORS (Cross-Origin Resource Sharing) will use a strict allow list of trusted domains and subdomains.

2.6.6 Secure Authentication and Session Management

- Implement secure session management practices, including session expiration, session token regeneration, and secure cookie attributes.
- Never reveal Session Tokens in URL parameters.

2.6.7 Error Handling and Logging

- Implement robust error handling mechanisms to provide meaningful error messages to users without exposing sensitive information.
- Log security-related events, errors, and exceptions to facilitate incident response and forensic analysis.
- When an unexpected or security sensitive error occurs, show a generic message, potentially with a unique ID.

2.6.8 Secure Communication

- Use secure communication protocols (e.g., TLS/SSL) to encrypt data transmitted between customers and servers.
- Verify server certificates and use strong cipher suites to prevent man-in-the-middle attacks.
- The communication with Backend must be protected against JSON injection attacks.
- Will run the application on port 443, is the most common port used for encrypted HTTPS traffic.

2.6.9 Secure Data Storage

- Encrypt sensitive data at rest using strong encryption algorithms and key management practices.
- Implement access controls and authorization mechanisms to restrict access to sensitive data based on user roles and permissions.
- Cron job that does a database backup every 24 hours.

2.6.10 Security Testing

- Conduct regular security testing throughout the development lifecycle, including static code analysis, dynamic application security testing (DAST), and penetration testing.
- Perform security reviews and code reviews to identify and remediate security vulnerabilities before deployment.

3 Design

3.1 Secure Design Pattern

3.1.1 Exposure Minization

Least Privilege

- Always give the user the minimum levels of access (or permissions) needed to perform their actions.

Least Information

- Do not provide unnecessary information. In implementation level, the cache should be cleaned when no longer needed.

Secure by Default

- The software must be secure in the default state and not insecure and treated after.

Allowlists over Blocklists

- Define what is possible, not what is not possible.

Avoid Predictability

- Any data (or behavior) that is predictable cannot be kept private.

Fail Securely

- Whenever the system fails, it should fail in a secure state.

3.1.2 Strong Enforcement

Complete Mediation

- Securely check all accesses to protected assets through the same authorization check.

Least Common Mechanism

- Do not share system mechanisms among users or programs except when necessary.

3.1.3 Redundancy

Defense in Depth

- Make security layers with defensive mechanisms to protect an asset.

Separation of Privilege

- Do not grant permission based on a single condition. Segregate parts of an IT environment based on its users and their roles.

3.1.4 Trust and Responsibility

Reluctance to Trust

- Verify the authenticity of the code before installing it, requiring a strong authentication before authorization. User input should not be trusted. Minimize trusted computing base.

Accept Security Responsibility

- Have clear duty to take responsibility for security.

3.2 Threat Modeling

3.2.1 Threat Model Process

Threat modeling is a systematic method used to identify and evaluate security threats and vulnerabilities in an application or system. It plays a vital role in a secure software development lifecycle (SSDLC), helping developers, architects, and security experts understand the potential threats an application might face and plan appropriate countermeasures.

The process starts by identifying what needs to be protected, which could include sensitive data, critical system components, or user information. Once assets are identified, the next step is to assess potential threats. To do this, you can use frameworks like the OWASP Threat Model, which provides a comprehensive guide to common threats and attack vectors. OWASP offers valuable tools and best practices for identifying possible threats.

After identifying threats, you need to assess vulnerabilities that could be exploited to target these assets. This step often involves using static and dynamic analysis tools, along with manual reviews, to uncover security weaknesses. Once vulnerabilities are identified, you analyze the risks by evaluating the likelihood and impact of each threat. This analysis helps you prioritize threats that need immediate attention.

With risks identified and analyzed, you can then develop a mitigation plan. This plan involves creating security controls and strategies to address high-risk threats. Mitigation could include implementing encryption, authentication, or access controls, depending on the nature of the threats and vulnerabilities.

The threat modeling process is iterative, meaning you should regularly revisit and revise the model to keep up with any changes to the system or the evolving threat landscape. This ensures that the security measures in place remain effective and relevant.

OWASP provides a wealth of resources for implementing robust security practices, including guides, tools, and community insights to support the threat modeling process. If you're planning to implement threat modeling, referencing OWASP's resources can offer significant benefits. [2]

3.2.2 Trust Levels

ID	Name	Description
1	Anonymous Web User	A user who has connected to TruckMotion but has not provided valid credentials.
2	User with Invalid Credentials	A user who has connected to TruckMotion and is attempting to log in using invalid login credentials.
3	Manager	Managers can register customers and drivers, approve or reject created jobs by the customers and assign jobs for the drivers.
4	Customer	Customers can create jobs requests and see all the information directly involved to their jobs.
5	Drivers	Drivers can see all the jobs assigned to them, start and finish those jobs.
6	Application FE Administrator	The Administrator FE can configure the application front-end.
7	Application BE Administrator	The Administrator BE can configure the application back-end.
8	Backend User Process	This is the process/user that the backend server executes code as and authenticates itself against the database server as.
9	Database Read User	The database user account used to access the database for read access.
10	Database Read/Write User	The database user account used to access the database for read and write access.
11	Database Server Administrator	The database server administrator has read and write permissions and can configure the database.
12	FileSystem Administrator	The file system administrator has read and write permissions and can configure the filesystem server

ID	Name	Description
13	FileSystem Read User	The file system user used to access the filesystem with read permissions
14	FileSystem Read/Write User	The file system user/write used to access the filesystem with read and write permissions
15	Logging System Read User	The logging system user account used to access loggings for read access.
16	Logging System Read/Write User	The logging system user account used to access loggings for read and write access.
17	Logging System Administrator	The logging system server administrator has read and write permissions and can configure the server.

Table 1 - Trust Levels

3.2.3 Entry Points

ID	Name	Description	Trust Levels
1	HTTPS Port	The TruckMotion website will only be accessible via TLS. All pages within this website are layered on this entry point.	1, 2, 3, 4, 5
1.1	Login Page	The login page for all the users of the system.	1, 2, 3, 4, 5
1.2	Login Function	The login function accepts user supplied credentials and compares them with those in the database.	2, 3, 4, 5
1.3	Customer Job Requests Status List Page	The customer job requests status list page lists all the requests that the customer have request, without any interaction and only pagination	3, 4
1.4	Driver Job Requests Assigned List Page	The driver job requests assigned list page lists all the drivers job requests with the information needed. This page also does not need user interaction	3, 5

ID	Name	Description	Trust Levels
1.5	Customer Adds Locations to his Account	This page allows user to add Locations to his Account.	4
1.6	Customer sees the progress of his Requested Service	This page shows the progress of the User's requested services.	3, 4
1.7	Customers request a delivery service.	This page allows the user to do a delivery request.	4
1.8	Driver signals the start and end of a delivery	This page allows the user to signal the start or end of a delivery	3, 5
1.9	Driver sends a photo or a screenshot to confirm delivery is done	This page allows the user to upload a photo to confirm delivery completion	3, 5
1.10	Manager approves or rejects a service request.	This page allows the user to approve or reject a service request.	3
1.11	Manager dispatch services to drivers.	This page allows the user to dispatch one or more services to a certain driver.	3
1.12	Manager register drivers.	This page allows the user to register drivers.	3
1.13	Manager register Customers.	This page allows the user to register customers.	3
1.14	User Changes his password.	This page allows the user to change his password.	2, 3, 4, 5

Table 2 - Entry Points

3.2.4 External Dependencies

ID	Description
1	The TruckMotion application will run on a free Hosting Service at first when it's still a prototype. However, in the future it is intended to deploy the application on a private machine of ours.
2	The database will be PostGreSQL and will run on a Linux Server. This server will include the installation of the latest operating system and application security patches.
3	The connection between the application and the database server will be over a private network.
4	The Backend and Frontend applications will communicate through HTTP requests on a private network.
5	The application will communicate with an External API for the Geographicals locations.
6	The application will communicate with the File System.

Table 3 - External Dependencies

3.2.5 Assets

ID	Name	Description	Trust Levels
1	**Users of the System**	**Assets relating to Managers, Drivers and Customers**	
1.1	Manager Login Details	The login credentials that the managers, drivers and customers will use to log into the TruckMotion application	3, 8, 9, 10, 11
1.2	Driver Login Details	The login credentials that the managers, drivers and customers will use to log into the TruckMotion application	4, 8, 9, 10, 11
1.3	Customer Login Details	The login credentials that the managers, drivers and customers will use to log into the TruckMotion application	5, 8, 9, 10, 11

ID	Name	Description	Trust Levels
1.4	Personal Data	TruckMotion will store personal information relating to Managers, Drivers and customers	3, 4, 5, 6, 7, 8 , 9, 10, 11
2	**System**	**Assets relating to the underlying system**	
2.1	Availabitily of TruckMotion Application	The TruckMotion Application should be available 24 hours a day and can be accessed by all managers, drivers and customers	6, 7, 11
2.2	Availability to Execute Code as a Backend Server	This is the ability to execute source code on the backend server as backend server user	7, 8
2.3	Availability to Execute SQL as Database Read User	This is the ability to execute SQL select queries on the database, and thus retrieve any information stored within the Truck Motion Database	9, 10, 11
2.3	Availability to Execute SQL as Database Read/Write User	This is the ability to execute SQL select, insert and update queries on the database, and thus have read and write access to any information stored within the TruckMotion database	10, 11
2.4	Availability to Read Files in FileSystem	This is the ability to read files in the filesystem and thus have read access to files stored inside this server	12, 13, 14
2.5	Availability to Read and Write Files in FileSystem	This is the ability to read and write files in the filesystem and thus have read and write access to any information stored inside the filesystem	12, 14
2.6	Availability to Read logging in the logging system	This is the ability to read loggings in the logging system and thus have read access to this system	15, 16, 17

ID	Name	Description	Trust Levels
2.7	Availability to Read and Write Loggings	This is the ability to read and write loggings in the logging system and this have read and write access to this system	16, 17
3	**Application**	**Assets relating to the Application of TruckMotion**	
3.1	Login Session	This is the login session of a user of the TruckMotion application, this User could be a Driver, Customer or Manager	2,3,4
3.2	Access to the Database Server	Access to the database server allows you to administer the database, giving you full access to the database users and all data contained within the database.	11
3.3	Ability to check information about the Customer Requested Services	The ability to check the data about the customer requested services	3, 4, 6, 7
3.4	Ability to check information about the Driver Assigned Services	The ability to check the data related to the driver assigned services	3, 4, 6, 7
3.5	Access to Audit Data	The audit data shows all audit-able events that occurred within the TruckMotion Application by Managers, Drivers and Customers	6, 7
3.6	Ability to ask for a delivery service	The ability to request that a certain product reaches a certain destination.	4,6,7
3.7	Ability to approve/reject a delivery request	The ability to approve or reject a certain delivery request that dictates the flow of it.	3,6,7

ID	Name	Description	Trust Levels
3.8	Ability to add Locations	The ability to add Locations so that it can be chosen on a service.	4,6,7
3.9	Ability to see the progress of an accepted Service	The ability to see the progress of an accepted Service to have full control of it	3,6,7
3.10	Ability to signal the start and end of a Transport.	The ability to signal the start and end of a Transport that has been provided to finish a service.	3,5,6,7
3.11	Ability to send a photo or a screenshot of the maps.	The ability to send a photo or a screenshot of the maps to confirm a delivery was made.	3,5,6,7
3.12	Ability to register Drivers.	The ability to register Drivers so that the Transports can be executed by them.	3,6,7
3.13	Ability to register Customers.	The ability to register Customers in the application so that new Services can be created.	3,6,7
3.14	Ability to login.	The ability to login into the application so that application features are accessible.	3,4,5,6,7
3.15	Ability to change a user password.	The ability to change a password of a user.	3,4,5,6,7

Table 4 - Assets

3.2.6 Data Flow Diagram and STRIDE

It will be used STRIDE Model to identify the threats. It classifies threats in six categories:

- **S**poofing
- **T**ampering
- **R**epudiation
- **I**nformation Disclosure
- **D**enial of Service
- **E**levation of Privilege

Those categories can be explained in the following table:

Type	Description	Security Control
Spoofing	Threat action aimed at accessing and use of another user's credentials, such as username and password.	• Appropriate Authentication • Protect Secret Data • Geolocation validation
Tampering	Threat action intending to maliciously change or modify persistent data, such as records in a database, and the alteration of data in transit between two computers over an open network, such as the Internet.	• Appropriate Authorization • Input Validations for Injection • Notify the owner if anything changed
Repudiation	Threat action aimed at performing prohibited operations in a system that lacks the ability to trace the operations.	• Audit Trails
Information Disclosure	Threat action intending to read a file that one was not granted access to, or to read data in transit.	• Authorization • Encryption • Protect secrets • Don't store secrets • SMS Verification Code • Robust authentication and authorization verification process
Denial of Service	Threat action attempting to deny access to valid users, such as by making a web server temporarily unavailable or unusable.	• Appropriate authentication • Appropriate authorization • Quality of service • Traffic Filtering • Rate Limiting
Elevation of privilege	Threat action intending to gain privileged access to resources to gain unauthorized access to information or to compromise a system.	• Implement Least Privilege • Logging • Robust authentication and authorization verification process

Table 5 - STRIDE Categories and Security Control

Table 5 explains each one of the six categories for threat categories. Also, it includes certain basic security controls.

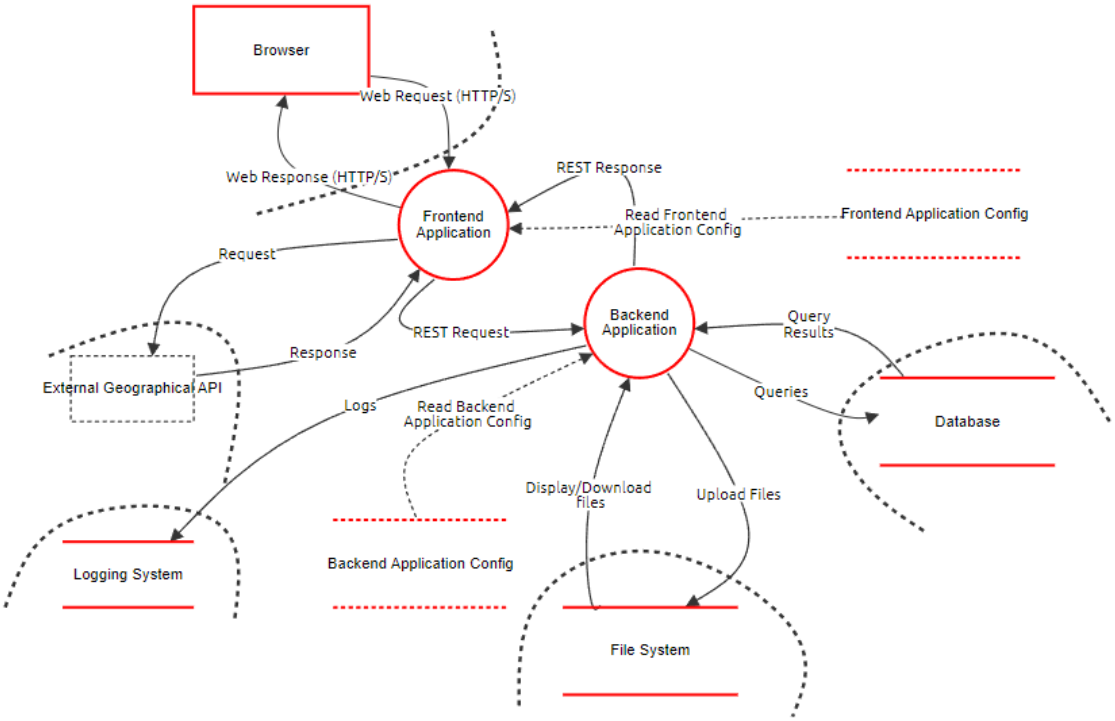

Figure 3 - Data Flow Diagram

The data flow diagram of the application is demonstrated in Figure 3, the tool used was OWASP Threat Dragon, and it generated a full report possible to be seen here, with all the STRIDE threats included, with description and mitigations included.

Our application has two actors:

- External Geographical API
  - This is an external API that will be used for the coordinates management and live tracker for the user.
  - This actor is out of scope because it will not be managed by our application. It is not our priority to secure because is external.
- Browser
  - The user will use the browser to interact with our application.

It has two Processes:

- Frontend Application
  - This process contains all the interactions needed with the frontend application. It will render the UI and it will send REST Requests and receive REST Responses from another process, the Backend Application.
  - It reads the frontend application configuration.
- Backend Application

    - This process is responsible to handle requests from the frontend process, handle them as needed and answer them with responses. It will gather data from the file system and from the database to answer the requests. It also writes logging into the logging system for every action made.
    - This process will contain all the business logic. So, it will log all the actions performed by the users.

And finally, it has five Stores:

- Frontend Application Config
    - This store will store the configuration for the frontend process. It will only be read by this process.
    - It is out of scope, but it must be secure because it can contain secrets.
- Backend Application Config
    - This store will store the configuration for the backend process. It will only be read by this process.
    - It is out of scope, but it must be secure because it can contain secrets.
- File System
    - This store represents the file system, and it contains files.
    - The backend process can upload files into it and download them from it.
- Database
    - This store contains all the data that the applications ask to persist.
    - The backend application will ask for the data in form of queries and the database will respond with the results.
- Logging System
    - This store contains all the logging of the backend application. It is only accessed by the backend process, and it does not send any data to any process.

To see in detail the STRIDE Model Threat for each one of these components, you can read them in the this PDF, including descriptions and mitigations.

3.3 Security Test Planning

3.3.1 Determine the Scope

- Identify the System or Application for Testing: Our security testing will encompass evaluating the truck management and dispatch application.
- Define the Target Environment: Our testing focus will center on the production environment where the application is deployed, alongside any pertinent development and testing environments.
- Establish Security Objectives: Our primary security objectives are to pinpoint and mitigate potential vulnerabilities that could compromise the confidentiality, integrity, and availability of the application and its data.

- Identify Stakeholders: Stakeholders involved in the security testing process include the development team, IT operations team, security team, and any relevant business stakeholders. Additionally, customers, managers, and drivers will play crucial roles. Customers will input service requests into the application, managers will dispatch corresponding services to drivers, and drivers will be responsible for updating service status and viewing assigned tasks.
- Define Roles and Responsibilities: It's imperative to clearly outline the roles and responsibilities of each stakeholder involved in the security testing process, including testers, developers, system administrators, and management. Moreover, in our academic environment project, we'll involve customers, managers, and drivers in specific roles as described above.
- Specify Constraints and Assumptions: We must identify any constraints or limitations that may impact the security testing process, such as resource limitations due to academic environment constraints, such as restricted access to paid databases or cloud resources for project deployment. Time constraints will also be a consideration, divided into academic project deadlines and timeframes.
- Determine Types and Levels of Security Testing: We'll specify the types and levels of security testing to be performed, including vulnerability assessment, penetration testing, code review, and compliance audit.
- Ensure Clarity, Realism, and Agreement: The scope must be clearly defined, realistic, and unanimously agreed upon by all parties involved, including academic project stakeholders, to ensure alignment and the successful execution of the security testing plan. To achieve that we will hold some weekly meetings and even daily when needed.

3.3.2 Conduct Risk Assessment

- Identify potential threats, vulnerabilities, and impacts: For this we will conduct a comprehensive assessment to identify potential security threats and vulnerabilities that could affect truck’s management and dispatch application. Considering factors such as unauthorized access, data breaches, denial of service attacks, and insider threats. We will also analyze the potential impacts of these threats on the confidentiality, integrity, and availability of the application and its data.
- Prioritize risks based on likelihood and severity: To achieve that we will evaluate the identified risks based on their likelihood of occurrence and potential severity of impact, prioritize risks accordingly to focus resources on addressing the most significant threats first.
- Define risk mitigation strategies and actions: In this phase we will develop risk mitigation strategies and action plans to address identified risks effectively. This may include implementing security controls, applying patches and updates, enhancing access controls, and improving monitoring and detection capabilities.
- Document and update the risk assessment: Finally, we will document the findings of the risk assessment, including identified threats, vulnerabilities, impacts, and mitigation strategies. Update the risk assessment regularly throughout the project lifecycle to account for changes in the threat landscape or the application's environment. Ensure

that stakeholders are aware of and aligned with the risk assessment findings and mitigation efforts.

3.3.3 Select Methods and Tools

- Assess the Scope, Objectives, and Risks: We will evaluate the scope, objectives, and identified risks of our security testing project, considering the unique characteristics and requirements of our truck management and dispatch application.
- Consider Available Resources and Skills: Considering the resources, expertise, and skills available within our academic project team, we'll ensure that the selected methods and tools can be effectively utilized to meet our security testing needs.
- Ensure Consistency, Reliability, and Effectiveness: Our focus will be on choosing methods and tools that demonstrate consistency, reliability, and effectiveness in identifying and addressing security weaknesses specific to our truck management application. Factors such as accuracy, coverage, and usability will guide our selection process.
- Compatibility with System Architecture and Technology: We'll select methods and tools that align with the architecture, technology stack, and functionality of our truck management application. It's crucial to ensure that the chosen tools can adequately test the various components and features of our application.
- Explore a Range of Methods and Tools: Our evaluation will encompass a range of methods and tools, including manual or automated testing, static or dynamic analysis, and open-source or commercial tools such as Sonarqube, Dependency Track, and ZAP. Each option will be carefully assessed for its suitability based on project requirements and constraints. Sonarqube gives us access to a SAST report and with Dependency Track we can have the SCA.
- Document the Selection Process: We will meticulously document the rationale behind the selection of specific methods and tools, outlining their strengths, limitations, and applicability to our academic project. Transparent documentation will ensure that all stakeholders are well-informed and aligned with our chosen approach to security testing.

3.3.4 Execute Security Testing

- Perform Security Testing Activities: We will conduct security testing activities, such as scanning, unit testing, manual testing, or auditing, on our truck management and dispatch application as outlined in our academic project's defined scope, methods, and tools.
- Follow Predefined Schedule, Procedure, and Checklist: We will adhere rigorously to a predefined schedule, procedure, and checklist to ensure consistency and comprehensiveness in executing our security testing endeavors. Our adherence to the testing plan will ensure systematic coverage of all facets of the application's security.
- Adhere to Ethical and Legal Standards: It is imperative for us to maintain adherence to ethical and legal standards throughout the security testing process, ensuring that all

activities are conducted in a manner that upholds the privacy and integrity of our application and its data. Compliance with relevant regulations, standards, and industry best practices will be paramount to maintaining integrity and legality.

- Monitor and Record Results and Findings: We will continuously monitor the progress of our security testing activities and diligently record the results, findings, and evidence acquired during the testing process. Thorough documentation of identified vulnerabilities, weaknesses, or issues, including severity ratings and potential impacts, will be essential for informed decision-making.
- Communicate Progress and Issues: We will foster transparent communication within our team and with all stakeholders involved in our academic project, including the development team, management, and other relevant parties. Regular updates on the progress and findings of our security testing efforts will be provided, with any significant issues or concerns promptly reported to facilitate timely resolution and decision-making achieved with meetings.

3.3.5 Report and Communicate

- Prepare a Comprehensive Security Testing Report: We will compile all relevant information obtained from the security testing process, including the scope, objectives, methods, tools, results, findings, and recommendations, into a detailed report tailored for our truck management and dispatch application.
- Summarize Key Findings and Results: We will provide a concise summary of the key findings and results derived from the security testing process. This summary will emphasize any significant vulnerabilities, weaknesses, or areas of concern identified during the evaluation.
- Include Strengths and Weaknesses Analysis: Our team will conduct an evaluation of the strengths and weaknesses of truck management and dispatch application security based on the observed results. We aim to provide an objective assessment of the overall security posture of the application.
- Provide Recommendations for Improvement: We will offer practical recommendations and suggestions aimed at enhancing the security of the application. These recommendations will be prioritized based on their potential impact and feasibility of implementation.
- Distribute the Report to Relevant Stakeholders: We will ensure that the security testing report is shared with all relevant stakeholders, including the development team, management, IT operations team, and other key decision makers. Timely distribution of the report will ensure that it reaches the appropriate audience for review and consideration.
- Facilitate Discussions and Decision-Making: Our team will schedule meetings or discussions to review the security testing report with stakeholders, facilitating informed decision-making. We will encourage open dialogue and collaboration to address any questions or concerns raised during the review process.

3.3.6 Review and Improve

- Evaluate Effectiveness, Efficiency, and Quality: We will conduct a comprehensive evaluation of the effectiveness, efficiency, and quality of our security testing activities, methods, tools, and report. We will assess whether our security testing plan successfully achieved its objectives and whether the desired outcomes were realized.
- Identify Lessons Learned and Best Practices: We will reflect on the lessons learned throughout the security testing process and identify best practices that contributed to the success of our testing activities. These lessons learned and best practices will be documented for future reference and to inform our approach in subsequent projects.
- Address Challenges and Opportunities for Improvement: Our team will identify any challenges or obstacles encountered during the security testing process and explore opportunities for improvement or enhancement. We will consider factors such as resource constraints, technical limitations, and process inefficiencies to identify areas for improvement.
- Provide Feedback and Recommendations: We will offer constructive feedback and recommendations for improving the security testing process and outcomes. Our recommendations will include actionable steps and initiatives to address identified areas for improvement and capitalize on opportunities for enhancement.
- Incorporate Feedback into Future Initiatives: It is essential to integrate the feedback and recommendations gathered from the review process into future security testing projects or initiatives. We will use the lessons learned to refine methodologies, streamline processes, and enhance the overall effectiveness of our security testing efforts.
- Foster a Culture of Continuous Improvement: Our team will promote a culture of continuous improvement within the organization by encouraging ongoing reflection, feedback, and collaboration among stakeholders involved in security testing activities. We will emphasize the importance of learning from past experiences and striving for excellence in future endeavors. This will contribute to the refinement and evolution of our security testing practices over time. [3]

4 ASVS v4 Checklist

In each iteration of this project, the final step will be to answer the ASVS v4 Checklist. ASVS stands for “Application Security Verification Standard” and it is a list of application security requirements or tests. This list is used across all the roles in software development such as architects, developers, testers, security professionals, tool vendors and consumers. This list intends to define, build, and verify secure applications.

There is a level associated with the results. There are 3 levels, which are:

- Level 1 – Opportunistic (Basic). Provides the most basic level of security. It is typically enough for small applications with low-security tasks.
- Level 2 – Standard. This level provides a higher level of security. It is typically required for application with medium security risks.
- Level 3 – Advanced. Provides the highest level of security. It is typically required for applications with high-security tasks. [4]

Security Category	Valid criteria	Total criteria	Validity Percentage	ASVS Level Acquired
Architecture, Design and Threat Modelling	19	39	0.05	L1
Authentication	21	57	0.04	L0
Session Management	10	13	0.08	L1
Access Control	6	9	0.07	L0
Validation, Sanitization and Encoding	16	26	0.06	L0
Stored Cryptography	9	16	0.06	L1
Error Handling and Logging	8	12	0.07	L1
Data Protection	12	16	0.08	L0
Communication	4	6	0.07	L1
Malicious Code	7	10	0.07	L0
Business Logic	8	8	0.10	L2
Files and Resources	13	14	0.09	L0
API and Web Service	7	9	0.08	L0
Configuration	20	24	0.08	L0
Total	160	259	0.06	L0

Table 6 - ASVS v4 Checklist Table Results

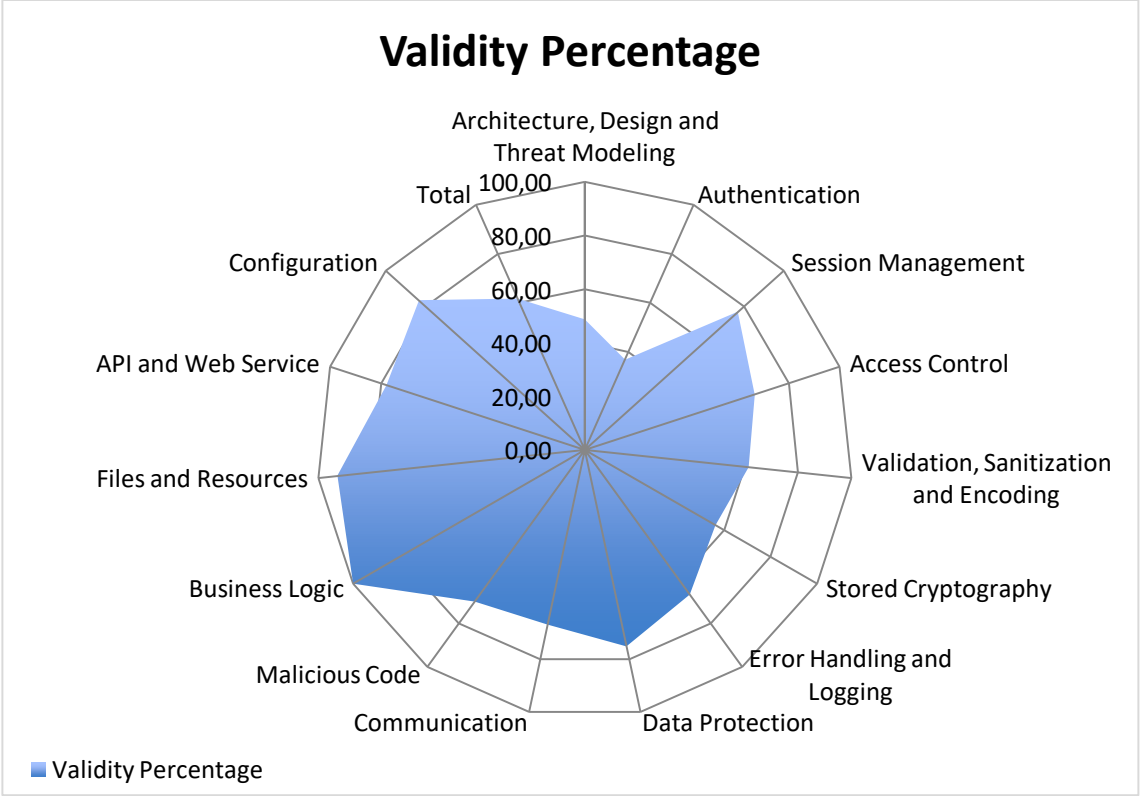

Figure 4 - ASVS v4 Checklist Graphic Results

After filling the checklist, we got the results shown in Table 6, which are represented in a circular graphic in Figure 4.

The first and most common level achieved is level 0. This is because if in each category, exists at least one level 1 check point that is not valid, the section automatically is level 0. As a result, our application achieved a level 0 in the ASVS v4 Checklist in this iteration of the project.

Although the level 1 was not hit, in Business Logic we achieved level 2 and 100% because, in terms of logic, it is well structured and the report and work focused more on this section, for this reason it is possible to have this section more robust and secure.

On the other sections, even though some hit level 1 and others not, they are almost all hitting more than 50%, which is a sign of progressive work towards getting better security.

The only one which did not hit 50% is Authentication. In this section, it is asked about encryption algorithms, cryptographic keys, and forms of authentication. As these are subjects which are new to all the members of the team and, for this reason, there is a bigger learning curve, we decided to not compromise in this section in the beginning because it will take time to develop everything that is asked in the checklist.

If there is a need to check all the details about all the sections and see the formulas executed for the results, the excel file is available here.

In this iteration, the report and work are focused on Analysis and Design. When filling the checklist points, we are compromising to execute the work the point filled is referring. Thus, as we are in short of time, resources and, sometimes, knowledge, we decide to not

compromise in advance and, in case of other iterations, another checklist points are achieved and implemented, those will be changed from Non-Valid to Valid and progressively stepping forward to achieve level 1. This way, we do not compromise work that we can't have assurance that will be done.

Our aim is to achieve level 1 since it is a small application for academic purposes, but in terms of the application business logic, it should, at least, level 2, since it will contain payments, services deliveries, and transportation of goods with live location.

5 Appendix

5.1 Appendix - Glossary

- AC: Access Control
- API: Application Programming Interface
- ASVS: Application Security Verification Standard
- AV: Attack Vector
- C: Confidentiality
- CVSS: Common Vulnerability Scoring System
- CWE/SANS: Common Weakness Enumeration / SANS Institute
- DAST: Dynamic Application Security Testing
- DDoS: Distributed Denial of Service
- DESOFS: Data Exposure Through Unsafe File Sharing
- DoS: Denial of Service
- GDPR: General Data Protection Regulation
- HTML: HyperText Markup Language
- HTTPS: HyperText Transfer Protocol Secure
- H: High (in context of risk rating)
- I: Integrity
- IT: Information Technology
- JSON: JavaScript Object Notation
- N: None (in context of user interaction)
- OAuth2: Open Authorization version 2
- OS: Operating System
- OWASP: Open Web Application Security Project
- PR: Privilege Required
- R: Required (in context of risk assessment)
- RBAC: Role-Based Access Control
- S: Severity
- SAST: Static Application Security Testing
- SCA: Software Composition Analysis
- SMS: Short Message Service
- SQL: Structured Query Language
- SSDLC: Secure Software Development Life Cycle
- TLS/SSL: Transport Layer Security / Secure Sockets Layer

- UI: User Interaction
- UML: Unified Modeling Language
- URL: Uniform Resource Locator
- XXS: Cross-Site Scripting
- ZAP: Zed Attack Proxy

5.2 Appendix – Documentation

The details of our analysis models, diagrams and all the results/conclusions showed within this report can be found in the folder Documentation within .md files in our repository.

6 References

[1] I. Object Management Group®, "UML," Unified Modeling Language, 2023. [Online]. Available: https://www.uml.org/.

[2] OWASP, "OWASP," 2024. [Online]. Available: https://owasp.org/www-community/Threat_Modeling.

[3] S. Sharma, A. Gupta e A. BM, "LinkedIn," 2024. [Online]. Available: https://www.linkedin.com/advice/0/what-main-steps-deliverables-security-testing.

[4] P. B. Sousa, *Secure Software Development Process*, Porto, Porto: Instituto Superior de Engenharia do Porto, 2024.