desofs_phase1_tm

Owner: desofs2025_wed_nap_1
Reviewer: nap@isep.ipp.pt
Contributors: 1181031@isep.ipp.pt, 1190830@isep.ipp.pt, 1211739@isep.ipp.pt, 1240465@isep.ipp.pt, 1240466@isep.ipp.pt
Date Generated: Sun Apr 27 2025

Executive Summary

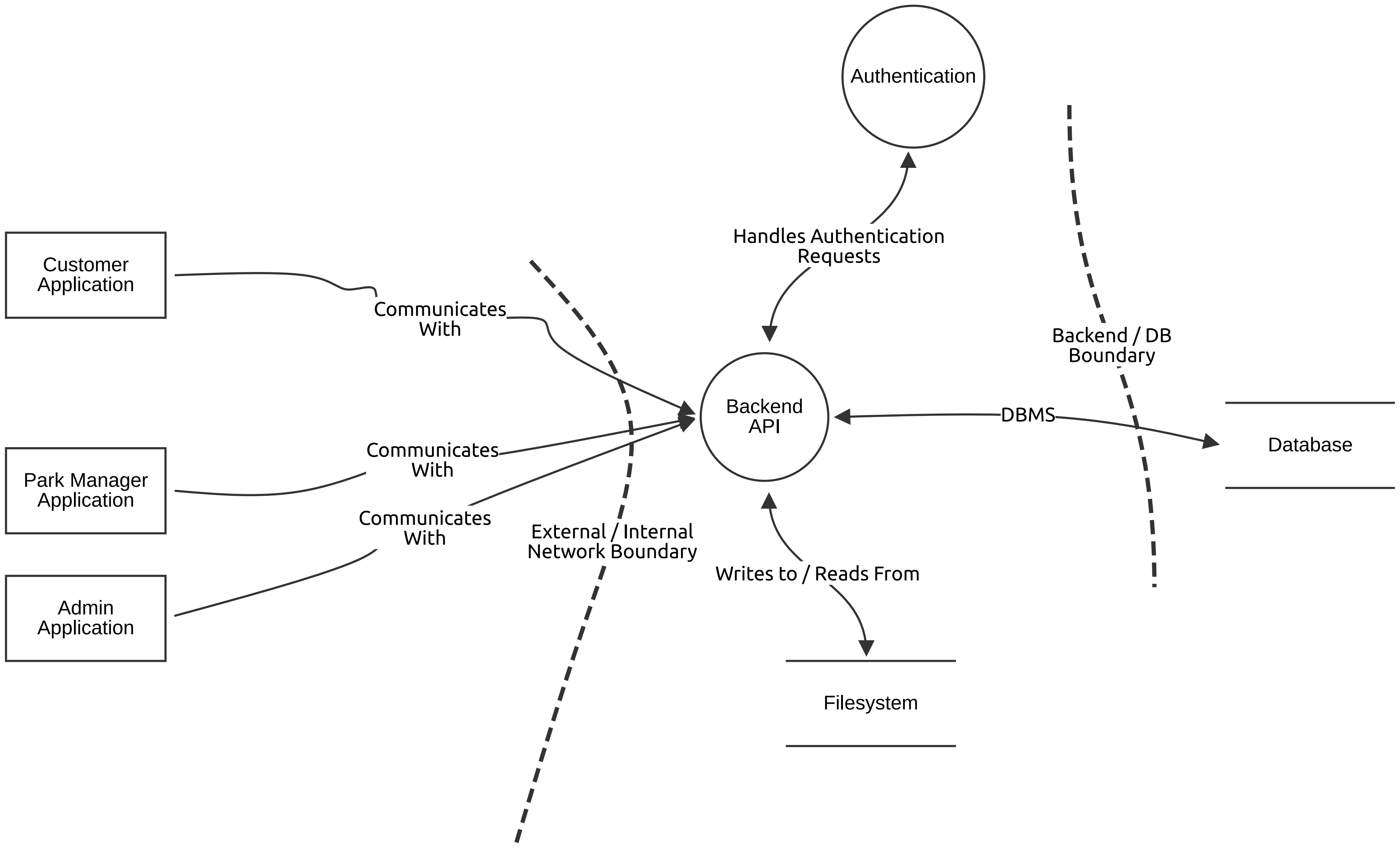
High level system description

Our application aims to provide an easier parking experience for our users. Through it, they can create an account, register their vehicles and gain access to parks whenever they need. Users issue parking requests, which are approved by our park managers, thus granting them permission to park their vehicle.

Summary

Total Threats	38
Total Mitigated	38
Not Mitigated	0
Open / High Priority	0
Open / Medium Priority	0
Open / Low Priority	0
Open / Unknown Priority	0

Threat Model



Threat Model

Backend API (Process)

Description: Backend API which processes requests by users.

Number	Title	Type	Priority	Status	Score	Description	Mitigations
1	Authentication Spoofing	Spoofing	Medium	Mitigated	9	An attacker spoofs a legitimate user by stealing their authentication token.	Implement 2FA with TOTP. Encrypt all data with TLS so that the token can't be stolen in-transit. Use tokens with short expiration windows.
2	Information Tampering	Tampering	Medium	Mitigated	6	Attackers might try to tamper the information stored by the application, performing malicious requests in order to do so.	Use TLS to encrypt every connection. Validate all user-provided input according to well-defined rules. Implement queries with prepared statements to prevent tampering via database operations.
3	Repudiation	Repudiation	Low	Mitigated	4	Users might try to deny having performed certain actions on the platform.	Implement robust logging solutions to ensure traceability and accountability of all performed actions.
4	Information Disclosure	Information disclosure	Medium	Mitigated	8	Attackers might try to release information which should not be accessible to the general public.	Validate all incoming requests. Encrypt data in transactions. Ensure that RBAC is strictly enforced so that access to data is allowed according to the least-privilege principle. Sanitize API responses to ensure no detailed error logs are sent to the client-side.
5	DDoS	Denial of service	High	Mitigated	12	An attacker might attempt to issue a flood of requests to the backend, rendering it unavailable to respond to legitimate requests	Implement rate limiting and request throttling measures to ensure that attackers can't flood the backend with requests.
6	Elevation of Privilege	Elevation of privilege	Medium	Mitigated	8	A user might try to bypass access controls via insecure endpoints to elevate his privilege to an admin or park manager, thus becoming able to tamper with sensitive data.	Enforce strict RBAC policies to ensure that users can't change their privileges. Re-validate the provided authentication token and match it with the operation being requested.

Database (Store)

Description: Relational Database used by the application

Number	Title	Type	Priority	Status	Score	Description	Mitigations
25	DB Tampering	Tampering	Medium	Mitigated	12	Attackers modify database contents	Strong access controls, least privilege
26	DB Information Disclosure	Information disclosure	High	Mitigated	16	Leaking sensitive DB data	Encrypt DB at rest and transit, access controls

Customer Application (Actor)

Description: Application used by regular users to interact with the API

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Number	Title	Type	Priority	Status	Score	Description	Mitigations
2	Impersonation of Legitimate Customer	Spoofing	Medium	Mitigated	8	An attacker may try to impersonate a legitimate customer by exploiting the absence of proper authentication mechanisms on the Customer Application side	Enforce strong authentication (e.g., OAuth2 or OpenID Connect)
3	Action Without Traceability	Repudiation	Medium	Mitigated	7	A customer might perform an action via the app (e.g., open the gate, book a parking slot) and later deny having done so. Without proper logging and authentication, it would be hard to prove otherwise.	Implement strong user authentication and session management and maintain secure, tamper-proof logs of actions with timestamps and user IDs.

Park Manager Application (Actor)

Description: Application used by Park Managers to interact with the API

Number	Title	Type	Priority	Status	Score	Description	Mitigations
7	Park Manager Identity	Spoofing	Medium	Mitigated	12	Someone could impersonate a Park Manager to gain access to park management functionalities	Enforce Multi-Factor Authentication (MFA) and implement Role-Based Access Control (RBAC)
8	Park manager repudiates changes	Repudiation	Medium	Mitigated	6	A Park Manager deletes or modifies data and later denies it	Maintain a secure, immutable audit log with user ID, timestamp, and affected data

Admin Application (Actor)

Description: Application used by Administrators to interact with the API

Number	Title	Type	Priority	Status	Score	Description	Mitigations
17	Admin Identity	Spoofing	Medium	Mitigated	12	Someone could impersonate an admin to gain access to park management functionalities	Enforce Multi-Factor Authentication (MFA) and implement Role-Based Access Control (RBAC)
18	Admin repudiates changes	Repudiation	Medium	Mitigated	6	An Admin deletes or modifies data and later denies it	Maintain a secure, immutable audit log with user ID, timestamp, and affected data

Writes to / Reads From (Data Flow)

Description: Communication flow between the Backend API and the filesystem

Number	Title	Type	Priority	Status	Score	Description	Mitigations
7	Tampering with system files	Tampering	Low	Mitigated	4	An attacker might try to tamper with data that's written to (or read from) the filesystem.	Run the backend process as non-root so that it can't alter critical files.
8	DDoS	Denial of service	High	Mitigated	12	An attacker might try to perform several actions which require writing to / reading from the filesystem in a short period of time, thus rendering the system unavailable due to the processing of these requests.	Limit the number of concurrent filesystem operations the backend can perform. Limit the maximum file size for user uploads or generated reports.

Number	Title	Type	Priority	Status	Score	Description	Mitigations
9	Obtaining Information	Information disclosure	Medium	Mitigated	6	An attacker might try to obtain information that's being written to or read from the filesystem without being allowed to do so.	Ensure that potentially sensitive data being stored on the filesystem (or read from it) through this data flow is encrypted, so that it cannot be disclosed.

Communicates With (Data Flow)

Description: Communication between Administrators and the Backend API

Number	Title	Type	Priority	Status	Score	Description	Mitigations
20	Modification of REST request content	Tampering	High	Mitigated	9	An attacker intercepts and alters requests sent from the admin application to the backend API, potentially allowing unauthorized access to administrative functionality or alteration of sensitive data.	Implement HTTPS (TLS) in all communications and signing of requests to ensure message integrity
21	Data leak	Information disclosure	High	Mitigated	15	An attacker could gain unauthorized access to sensitive data (e.g., user data, system configurations) by exploiting vulnerabilities in the communication channel between the Admin Application and the Backend API	Enforce HTTPS for all communication. Implement strict access controls and input validation on the backend to prevent unauthorized data access. Minimize the data transmitted in each request and response
22	REST endpoint flooding	Denial of service	High	Mitigated	12	An attacker could flood the Backend API with requets from the Admin Application, exhausting system resources and making the system unavailable to the users.	Implement rate limiting and request throttling on the backend API. Consider using a web application firewall to detect and block malicious traffic. Employ DDoS protection mechanims

Communicates With (Data Flow)

Description: Communication flow between the Park Manager application and the Backend API

Number	Title	Type	Priority	Status	Score	Description	Mitigations
9	Modification of REST request content	Tampering	High	Mitigated	12	The data exchanged via REST API is vulnerable to interception and modification	Encrypt all communications with HTTPS, verify request signatures server-side
10	Data leak	Information disclosure	High	Mitigated	15	Data such as park configurations or access tokens are leaked over an insecure connection	Enforce HTTPS, avoid sending excessive data in API responses and apply data minimization
11	REST endpoint flooding	Denial of service	High	Mitigated	12	Someone sends a flood of REST calls, degrading performance or crashing the backend	Use rate limiting and DDoS protection mechanisms

Handles Authentication Requests (Data Flow)

Description: Communication flow between the Backend API and the authentication service

Number	Title	Type	Priority	Status	Score	Description	Mitigations

Number	Title	Type	Priority	Status	Score	Description	Mitigations
10	Tampering with authentication data	Tampering	Medium	Mitigated	6	An attacker might try to tamper with data being sent from the backend to the authentication service in order to elevate his privileges.	Encrypt all connections with TLS. Always verify claims being issued with a token.
11	Disclose User Authentication Details	Information disclosure	Medium	Mitigated	8	An attacker might try to disclose sensitive data being transmitted in this communication channel.	Ensure connections are encrypted using TLS. Ensure only the necessary data is transmitted in each transaction.
12	DDoS of Authentication Service	Denial of service	Medium	Mitigated	9	Attackers spam login/MFA endpoints to exhaust backend or authentication service resources.	Implement rate limiting between these services to ensure that resources are not exhausted. Use account lockout after repeated failures of login attempts.

Communicates With (Data Flow)

Description: Communication flow between the Customer application and the Backend API

Number	Title	Type	Priority	Status	Score	Description	Mitigations
4	Modified API Requests	Tampering	High	Mitigated	9	An attacker intercepts and alters requests sent from the customer application to the backend API, potentially changing parameters (e.g., altering parking reservation details).	Enforce HTTPS (TLS) on all communications.
5	Sensitive Data Leak in Transit	Information disclosure	High	Mitigated	9	Sensitive user data (e.g., location, personal info) might be exposed if intercepted during transmission between the app and backend.	Encrypt all communications using TLS and, avoid exposing sensitive data in URLs (using POST body instead).
6	API Request Flooding	Denial of service	Medium	Mitigated	7	An attacker or malicious user could flood the backend API with requests from the customer application, making the system unavailable to other users.	Implement rate limiting and throttling on the API endpoints and monitor and block abusive IPs or behavior patterns

DBMS (Data Flow)

Description: Communication flow between the API and the database

Number	Title	Type	Priority	Status	Score	Description	Mitigations
32	API ↔ DB Connection Data Leak	Information disclosure	Medium	Mitigated	12	Data leaks during connection	Secure connection
33	API ↔ DB Connection Tampering	Tampering	Medium	Mitigated	4	Malicious query modification	Strong connection security

Filesystem (Store)

Description: Filesystem used by the Backend API to store/read data necessary to the fulfillment of some use cases.

Number	Title	Type	Priority	Status	Score	Description	Mitigations
12	Tampering with stored data	Tampering	Medium	Mitigated	8	A malicious actor modifies stored data, such as logs, user-uploaded content, or application configurations.	Use containerization technologies to isolate filesystem interactions from the host. Run the application with a dedicated OS user to prevent modifications to system files.

Number	Title	Type	Priority	Status	Score	Description	Mitigations
13	Repudiation of file operations	Repudiation	Medium	Mitigated	8	A user might deny having performed an action with system files.	Implement logging on file operations, detailing what application user performed what action for a certain file or set of files. Ensure logs are immutable.
14	Information Disclosure	Information disclosure	Medium	Mitigated	9	An attacker might disclose sensitive information sourced from files stored in the filesystem, such as license plate information.	Ensure that files which contain sensitive data are encrypted at-rest. Ensure the least-privilege principle when accessing files.
15	DoS to filesystem	Denial of service	Low	Mitigated	6	A malicious actor might try to flood the filesystem with user uploads or log flooding (i.e. performing several operations which generate logs in the system in quick succession), causing a system failure.	Ensure that no user upload can exceed a maximum file size. Implement log rotation according to file size limits.

Authentication (Process)

Description: Authentication service which handles the authentication process for the application

Number	Title	Type	Priority	Status	Score	Description	Mitigations
16	Authentication Spoofing	Spoofing	Medium	Mitigated	4	Fake user logins	Rely on SSO/OAuth securely
27	Authentication Tampering	Tampering	Low	Mitigated	4	Forged tokens	Token signing, validation
28	Authentication DoS	Denial of service	Medium	Mitigated	12	Overloading auth service	Add rate limiting, CAPTCHA