

Projektarbeit – Garage Job Card Manager

Digitale Verwaltungsplattform für Fahrzeugreparaturen mit Django, React und Docker

Beschreibung des Projekts

Ich möchte die Prozesse einer Reparatur-/Wartungswerkstatt für Fahrzeuge verbessern.

Der aktuelle Ablauf ist wie folgt:

- Ein Arbeiter füllt einen Tätigkeitsbericht aus, der die durchgeführten Aufgaben an einem Fahrzeug/Motor beschreibt.
- Dieser Bericht wird vom Sekretariat übernommen, welches ihn erfasst und eine Rechnung erstellt. Wobei der Preis der durchgeführten Aufgaben sowie der verwendeten Teile/Materialien manuell ermittelt wird.
- Der Bestand wird manuell verfolgt.

Das Ziel ist es, eine Anwendung zu erstellen, die das Management einer Reparatur-/Wartungswerkstatt ermöglicht, die Folgendes umfasst:

- Die Erfassung von Arbeitsberichten (Kontrolle/Wartung), die die durchgeführten Aufgaben am Fahrzeug auflisten. Dieser Bericht wird dem Arbeiter zugeordnet, der ihn erstellt hat. Er hat drei Status (ausstehend, in Bearbeitung, abgeschlossen).
- Die Erstellung einer Kunden-/Fahrzeugakte, die direkt mit den erstellten Berichten und Rechnungen verknüpft ist.
- Die Erstellung von Rechnungen auf Basis der erstellten Berichte.
- Die automatische Aktualisierung der verfügbaren Bestände an Materialien/Teilen sowie die Benachrichtigung, wenn der Bestand niedrig ist.

Das Ziel ist es, die Eingabe der Berichte zu vereinfachen und den Papierverbrauch zu reduzieren. Die Anzahl der Eingaben von Informationen zu verringern, sowie die Verbesserung und Automatisierung der Nachverfolgung von Rechnungen und Beständen.

Roadmap

Setup

- [Applikationsidee](#)
- [Technologie Stack](#)
- [System Requirements](#)
- [Architektur](#)
- [Version control](#)
- [Features](#)

Front End Specifications

- [Frontend Anforderungen](#)

Backend Specifications

- [Backend Anforderungen](#)

Data and flow

- [Datenbank](#)
- [Flow Chart](#)

UI/UX and testing

- [Screenshots \(Wireframe, UI , UX\)](#)
- [Style and responsiveness](#)
- [Automated tests \(unit, integration tests\)](#)
- [Containerized with Docker](#)
- [Accessibility](#)
- [Light/Dark mode](#)

Deployment and maintenance

- [Deploy to VPS with Nginx](#)
- [Monitor performance and security](#)
- [User feedback](#)
- [Future updates](#)

Applikationsidee

Meine Anwendung ermöglicht es den Arbeitern in der Werkstatt, Arbeitsblätter zu erfassen, welche die durchgeführten Tätigkeiten, sowie die verwendeten Materialien/Teile beschreiben. Diese Erfassung kann gespeichert und später fortgesetzt oder gelöscht werden, jedoch immer durch denselben Arbeiter (dies könnte je nach Wunsch des Administrators der Anwendung anpassbar sein). Dieses Arbeitsblatt hat vier Status: *"in Wartestellung"*, *"in Bearbeitung"*, *"abgeschlossen"* oder *"exportiert"*.

Die vom Arbeiter verwendeten Materialien und Teile (im Arbeitsblatt vermerkt), werden bei der Speicherung automatisch vom Inventar abgezogen.

Ein *"exportiertes"* Arbeitsblatt kann nicht mehr bearbeitet werden.

Meine Anwendung ermöglicht es auch einem Verwaltungssekretär, die Arbeitsblätter einzusehen und *"abgeschlossene"* Arbeitsblätter zu exportieren, indem automatisch eine Rechnung im pdf-Format erstellt wird.

Der Verwaltungssekretär kann außerdem neue Kunden oder neue Fahrzeuge anlegen, bearbeiten oder löschen.

Das Inventar kann auch über eine dedizierte Seite aktualisiert werden und der Benutzer erhält eine Benachrichtigung, wenn das Inventar beim Login unter einen bestimmten Schwellenwert fällt (der von einem Administrator festgelegt werden könnte).

Die Verwaltung der Anpassung der Anwendung durch den Administrator ist nicht enthalten (wird hinzugefügt, falls die Zeit es zulässt).

Die Nutzung der Daten für statistische Zwecke ist nicht enthalten (wann ist die Hauptarbeitszeit, wer arbeitet wie schnell, welche Materialien oder Teile werden am häufigsten verwendet, usw.).

Alle Benutzer haben denselben Status, sie können alle entweder Arbeiter oder Sekretär sein.

Technologie Stack

Meine Anwendung wird auf React (Javascript) für das Frontend, Django/Python für das Backend und einer MySQL-Datenbank basieren.

Die React-States werden mit Zustand (state management) verwaltet.

Die Webanwendung wird mit Docker containerisiert, um Kompatibilitätsprobleme zu vermeiden.

System requirements

Recommended System Requirements (for production / moderate traffic)

1. CPU: 4 cores or more

- Ermöglicht parallele Verarbeitung für Gunicorn-Worker und MySQL-Threads

2. RAM: 8 GB minimum

- 2 GB for MySQL (mit InnoDB Buffer Pool)
- 2–3 GB for Gunicorn workers
- 500–750 MB for React static file serving + NGINX
- Rest für Docker und Betriebssystem

3. Storage: 50+ GB SSD

- Inklusive Backup-Strategie (z. B. automatische externe Backups) (zukünftige Erweiterung)

4. Network:

- Stabile Verbindung mit **static IP**
- HTTPS eingerichtet über **Let's Encrypt**

5. Other Requirements:

- Swap deaktivieren oder begrenzen, um Performance-Einbußen zu vermeiden

- Firewall mit UFW
- Monitoring tools (z.B. **htop**, **docker stats**, **Netdata**)
- Logging stack (optional: ELK stack or external log shipper)

Security-Specific Recommendations

- **Docker:** Container ohne Root-Rechte ausführen, Berechtigungen einschränken.
- **Database:** MySQL-Benutzer nur mit nötigen Berechtigungen.
- **NGINX:** Rate Limiting aktivieren, Versionsinformationen ausblenden.
- **Django:** DEBUG=False, ALLOWED_HOSTS korrekt setzen, sichere Einstellungen für Secret Keys, Cookies, HTTPS-Weiterleitung etc.

Architektur

Mein Projekt basiert auf einer Client-Server-Architektur und einem MVC-Modell (Model: MySQL – View: React – Controller: Django REST Framework) als Single-Page Application (SPA).

Version Control

Mein Projekt verwendet die Versionskontrollsoftware GIT, wobei die Daten auf der Website GitLab gespeichert werden.

Features

Core features

- User authentication (FAFE-001 to 005, 008, 009, FAFE-001 to 003)
- Reports, Clients, Vehicles, Inventory, Tasks => CRUD (FAFE-011, 018 to 021, FAFE-004 to 007)
- Automatic inventory updates (FAFE-009 to 010)

- Automatic invoices generation (FAFE-022, FAFE-011)

Advanced features

- Sort and filter (FAFE-017, FAFE-008)
- dashboard
- notifications
- multi-client support
- data inputs verification and help
- light/dark theme

Front-End Anforderungen

Name	Login	ID	FAFE-001
Ziel	Der Benutzer muss sich über die Login-Seite in der Anwendung anmelden können.		
Ereignis	Der Benutzer hat seine Anmeldeinformationen eingegeben und klickt auf den Button "Login".		
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). • Der Benutzer besitzt seine Anmeldeinformationen (er ist registriert). 		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer gibt seine Anmeldeinformationen ein (E-Mail-Adresse und Passwort). 2. Die Informationen werden während der Eingabe validiert. 3. Der Benutzer klickt auf den Button "Login". 4. Die Anmeldeinformationen werden über die Authentifizierungs-API überprüft (FAFE-001). 5. Der Benutzer erhält eine Erfolgsmeldung zur erfolgreichen Anmeldung. 6. Der Benutzer wird zur Dashboard-Seite weitergeleitet. 		
Alternativablauf	<ul style="list-style-type: none"> • E-Mail-Adresse falsch formatiert: Der Benutzer erhält eine Benachrichtigung, die ihn darauf hinweist, dass das Format der E-Mail-Adresse ungültig ist. • E-Mail-Adresse oder Passwort falsch: Der Benutzer erhält eine Benachrichtigung, die ihn darauf hinweist, dass die eingegebene E-Mail-Adresse oder das Passwort falsch ist. • Der Benutzer klickt auf den Link zur Registrierung. 		

Nachbedingungen Erfolg	<ul style="list-style-type: none"> • Der Benutzer erhält eine Erfolgsmeldung. • Nach der Authentifizierung wird der Benutzer direkt auf die Dashboard-Seite weitergeleitet. 		
Nachbedingungen Fehler	<ul style="list-style-type: none"> • Der Benutzer erhält eine Fehlermeldung. • Der eingegebene Text bleibt im Textfeld bestehen. 		
Klassifizierung	Funktional, MUSS	Aufwand	MITTEL

Name	Logout	ID	FAFE-002
Ziel	Der Benutzer muss sich über die Logout-Schaltfläche auf der Anwendung abmelden können.		
Ereignis	Der Benutzer klickt auf den Button "Logout".		
Vorbedingungen	Der Benutzer ist angemeldet.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf den Button "Logout". 2. Die Sitzung des Benutzers wird über die Authentifizierungs-API geschlossen (FAFE-002). 3. Der Benutzer erhält eine Erfolgsmeldung. 4. Der Benutzer wird zur Login-Seite weitergeleitet. 		
Alternativablauf	Kein		
Nachbedingungen Erfolg	<ul style="list-style-type: none"> • Der Benutzer erhält eine Erfolgsmeldung. • Nach der Abmeldung wird der Benutzer direkt auf die Login-Seite weitergeleitet. 		
Nachbedingungen Fehler	<ul style="list-style-type: none"> • Der Benutzer erhält eine Fehlermeldung. 		
Klassifizierung	Funktional, MUSS	Aufwand	MITTEL

Name	Register	ID	FAFE-003
Ziel	Der Benutzer muss sich über die Register-Seite in der Anwendung registrieren können.		
Ereignis	Der Benutzer klickt auf den Button "Register".		
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer befindet sich auf der Register-Seite (Registrierungsseite). • Der Benutzer besitzt seine Registrierungsinformationen (er ist nicht registriert). 		

Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer gibt seine Registrierungsinformationen ein (Username, E-Mail-Adresse und Passwort). 2. Die Informationen werden während der Eingabe validiert. 3. Der Benutzer klickt auf den Button "Register". 4. Die Registrierungsinformationen werden über die Authentifizierungs-API überprüft (FAFE-003). 5. Der Benutzer erhält eine Erfolgsmeldung zur erfolgreichen Registrierung. 6. Der Benutzer wird zur Login-Seite weitergeleitet. 		
Alternativablauf	<ul style="list-style-type: none"> • Der Benutzer gibt einen bereits verwendeten Benutzernamen oder eine bereits verwendete E-Mail-Adresse ein: Der Benutzer erhält eine Benachrichtigung, die ihn darauf hinweist, dass der Benutzername oder die E-Mail-Adresse bereits verwendet wird. • Username falsch formatiert: Der Benutzer erhält eine Benachrichtigung, die ihn darauf hinweist, dass das Format der Username ungültig ist. • E-Mail-Adresse falsch formatiert: Der Benutzer erhält eine Benachrichtigung, die ihn darauf hinweist, dass das Format der E-Mail-Adresse ungültig ist. • Der Benutzer klickt auf den Link zum Login. 		
Nachbedingungen Erfolg	<ul style="list-style-type: none"> • Der Benutzer erhält eine Erfolgsmeldung. • Nach der Registrierung wird der Benutzer direkt auf die Login-Seite weitergeleitet. 		
Nachbedingungen Fehler	<ul style="list-style-type: none"> • Der Benutzer erhält eine Fehlermeldung. • Der eingegeben Text bleibt im Textfeld bestehen. 		
Klassifizierung	Funktional, MUSS	Aufwand	MITTEL
Name	Login to register	ID	FAFE-004
Ziel	Der Benutzer muss über einen Link von der Login-Seite zur		

	Registrierungsseite wechseln können.		
Ereignis	Der Benutzer klickt auf den Link "Register".		
Vorbedingungen	<ul style="list-style-type: none"> Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). 		
Standardablauf	1. Der Benutzer klickt auf den Button "Register". 2. Der Benutzer wird zur Register-Seite weitergeleitet.		
Alternativablauf	Kein		
Nachbedingungen Erfolg	<ul style="list-style-type: none"> Nach den Klick wird der Benutzer direkt auf die Register-Seite weitergeleitet. 		
Nachbedingungen Fehler	Kein		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Register to Login	ID	FAFE-005
Ziel	Der Benutzer muss über einen Link von der Registrierungsseite zur Login-Seite wechseln können.		
Ereignis	Der Benutzer klickt auf den Link "Login".		
Vorbedingungen	<ul style="list-style-type: none"> Der Benutzer befindet sich auf der Register-Seite (Registrierungsseite). 		
Standardablauf	1. Der Benutzer klickt auf den Button "Login". 2. Der Benutzer wird zur Login-Seite weitergeleitet.		
Alternativablauf	Kein		
Nachbedingungen Erfolg	<ul style="list-style-type: none"> Nach den Klick wird der Benutzer direkt auf die Login-Seite weitergeleitet. 		
Nachbedingungen Fehler	Kein		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Show Sidebar	ID	FAFE-006
Ziel	Die Navigationsleiste muss angezeigt werden, wenn der Benutzer authentifiziert ist.		
Ereignis	Der Benutzer hat sich angemeldet.		

Vorbedingungen	Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite).		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). 2. Der Benutzer klickt auf den Button "Login". 3. Der Benutzer erhält eine Erfolgsmeldung zur erfolgreichen Anmeldung. 4. Der Benutzer wird zur Dashboard-Seite weitergeleitet. 5. Die Navigationsleiste wird angezeigt. 		
Alternativablauf	Kein		
Nachbedingungen Erfolg	Kein		
Nachbedingungen Fehler	Kein		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Hide Sidebar	ID	FAFE-007
Ziel	Die Navigationsleiste muss versteckt werden, wenn der Benutzer abgemeldet ist.		
Ereignis	Der Benutzer hat sich abgemeldet.		
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer ist angemeldet. 		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der Privat-Seite. 2. Der Benutzer klickt auf den Button "Logout". 3. Der Benutzer erhält eine Erfolgsmeldung zur erfolgreichen Abmeldung. 4. Der Benutzer wird zur Login-Seite weitergeleitet. 5. Die Navigationsleiste wird angezeigt. 		
Alternativablauf	Kein		
Nachbedingungen Erfolg	Kein		
Nachbedingungen Fehler	Kein		

Klassifizierung	Funktional, MUSS	Aufwand	TIEF
------------------------	------------------	----------------	------

Name	Show Username	ID	FAFE-008
Ziel	Der Benutzername muss in der Navigationsleiste angezeigt werden, wenn der Benutzer authentifiziert ist.		
Ereignis	Der Benutzer hat sich angemeldet.		
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). • Der Benutzer besitzt seine Anmeldeinformationen (er ist registriert). • Der Benutzer hat auf den Login-Button geklickt. 		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). 2. Der Benutzer klickt auf den Button "Login". 3. Der Benutzer erhält eine Erfolgsmeldung zur erfolgreichen Anmeldung. 4. Der Benutzer wird zur Dashboard-Seite weitergeleitet. 5. Die Navigationsleiste mit der Benutzername wird angezeigt. 		
Alternativablauf	Kein		
Nachbedingungen Erfolg	Kein		
Nachbedingungen Fehler	Kein		
Klassifizierung	UI, KANN	Aufwand	TIEF

Name	Show Logout button	ID	FAFE-009
Ziel	Der Logout-Button muss in der Navigationsleiste angezeigt werden, wenn der Benutzer authentifiziert ist.		
Ereignis	Der Benutzer hat sich angemeldet.		
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). • Der Benutzer besitzt seine Anmeldeinformationen (er ist 		

	registriert). • Der Benutzer hat auf den Login-Button geklickt.		
Standardablauf	1. Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). 2. Der Benutzer klickt auf den Button "Login". 3. Der Benutzer erhält eine Erfolgsmeldung zur erfolgreichen Anmeldung. 4. Der Benutzer wird zur Dashboard-Seite weitergeleitet. 5. Die Navigationsleiste mit der "Logout"-Button wird angezeigt.		
Alternativablauf	Kein		
Nachbedingungen Erfolg	Kein		
Nachbedingungen Fehler	Kein		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Show Menu	ID	FAFE-010
Ziel	Der Menu muss in der Navigationsleiste angezeigt werden, wenn der Benutzer authentifiziert ist.		
Ereignis	Der Benutzer hat sich angemeldet.		
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). • Der Benutzer besitzt seine Anmeldeinformationen (er ist registriert). • Der Benutzer hat auf den Login-Button geklickt. 		
Standardablauf	1. Der Benutzer befindet sich auf der Login-Seite (Anmeldeseite). 2. Der Benutzer klickt auf den Button "Login". 3. Der Benutzer erhält eine Erfolgsmeldung zur erfolgreichen Anmeldung. 4. Der Benutzer wird zur Dashboard-Seite weitergeleitet. 5. Die Navigationsleiste mit dem Menu wird angezeigt.		

Alternativablauf	Kein		
Nachbedingungen Erfolg	Kein		
Nachbedingungen Fehler	Kein		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Link to Seiten	ID	FAFE-011
Ziel	Die Seite soll im rechten Bereich des Bildschirms angezeigt werden, wenn der Benutzer im Menü auf den Link der Seite klickt.		
Ereignis	Der Benutzer hat im Menü auf den Link der Seite geklickt.		
Vorbedingungen	Der Benutzer ist registriert und angemeldet.		
Standardablauf	1. Der Benutzer befindet sich auf einer Privat-Seite. 2. Der Benutzer klickt auf den Link der Seite im Menü. 3. Der Benutzer wird zur Seite weitergeleitet.		
Alternativablauf	Kein		
Nachbedingungen Erfolg	Kein		
Nachbedingungen Fehler	Kein		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Filter Bar	ID	FAFE-017
Ziel	Der Benutzer hat Zugriff auf eine Filterleiste, mit der er die Elemente der Seite nach verschiedenen Kriterien filtern kann. Die Elemente der Seite werden basierend auf den ausgewählten Filtern gefiltert.		
Ereignis	Der Benutzer gibt den Text in eines der Filterfelder ein oder wählt ein Element aus einer Filterliste aus.		
Vorbedingungen	Der Benutzer befindet sich auf einer Privat-Seite (Reports, Owners, Vehicles, Invoices, Inventory)		

Standardablauf	1. Der Benutzer befindet sich auf einer Privat-Seite (Reports, Owners, Vehicles, Invoices, Inventory) 2. Der Benutzer hat Zugriff auf eine Filterleiste, mit der er die Elemente der Seite nach verschiedenen Kriterien filtern kann. 3. Die Elemente der Seite werden basierend auf den ausgewählten Filtern gefiltert. 4. Oder eine Nachricht „Es gibt keine Elemente, die Ihrer Suche entsprechen“ wird angezeigt.		
Alternativablauf	1. Der Benutzer befindet sich auf einer Privat-Seite (Reports, Owners, Vehicles, Invoices, Inventory) 2. Der Benutzer hat die Filter auf den Ausgangszustand zurückgesetzt. 3. Alle Elemente der Seite werden ohne Filter angezeigt.		
Nachbedingungen Erfolg	Die Elemente der Seite werden basierend auf den ausgewählten Filtern gefiltert.		
Nachbedingungen Fehler	Kein		
Klassifizierung	UX, KANN	Aufwand	MITTEL

Name	View Item	ID	FAFE-018
Ziel	Ein Modal Fenster öffnet sich, um das vom Benutzer ausgewählte Element im Nur-Lese-Modus anzuzeigen.		
Ereignis	Der Benutzer klickt auf ein Element.		
Vorbedingungen	Der Benutzer ist registriert und angemeldet.		
Standardablauf	1. Der Benutzer klickt auf ein Element in der Liste. 2. Ein Modal Fenster öffnet sich, um die Informationen zum ausgewählten Element anzuzeigen. (FABE-005) 3. Der Benutzer kann das Element bearbeiten oder löschen, indem er auf die entsprechende Schaltfläche klickt, oder das Modal Fenster schließen, indem er auf die Schaltfläche zum Schließen klickt.		
Alternativablauf	1. Der Benutzer klickt auf ein Element in der Liste. 2. Bei einem Problem wird eine Fehlermeldung angezeigt, die dem Benutzer die Ursache des Fehlers		

	mitteilt.		
Nachbedingungen Erfolg	Das Modal Fenster zeigt die Informationen zum ausgewählten Element an.		
Nachbedingungen Fehler	Eine Fehlermeldung wird angezeigt und der Benutzer wird automatisch zur Liste der Elemente zurückgeleitet, in der er sich zuvor befand.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Create Item	ID	FAFE-019
Ziel	Ein Modal Fenster öffnet sich, um ein neues Element zu erstellen.		
Ereignis	Der Benutzer klickt auf die Schaltfläche „Neues Element erstellen“.		
Vorbedingungen	Der Benutzer ist registriert und angemeldet.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf die Schaltfläche „Neues Element erstellen“. 2. Ein Modal Fenster öffnet sich, dieses ermöglicht, die Informationen für ein neues Element einzugeben. 3. Nach der Eingabe der Datei kann der Benutzer das Element speichern, indem er auf die Schaltfläche „Speichern“ klickt (FAFE-004) 		
Alternativablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf die Schaltfläche „Neues Element erstellen“. 2. Ein Modal Fenster öffnet sich, dieses ermöglicht, die Informationen für ein neues Element einzugeben. 3. Eine Fehlermeldung wird angezeigt, die dem Benutzer die Ursache des Problems mitteilt. 4. Der Benutzer kann die Erstellung abbrechen, indem er auf das Schließen-Symbol des Modal Fensters klickt. 		
Nachbedingungen Erfolg	Ein neues Element wird erstellt, der Benutzer erhält eine Bestätigungsnachricht und wird zurück zur Liste der Elemente geleitet.		
Nachbedingungen Fehler	Eine Fehlermeldung wird angezeigt und der Benutzer wird automatisch zum Modal Fenster oder zur Elementliste		

	zurückgeleitet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Edit Item	ID	FAFE-020
Ziel	Ein Modal Fenster öffnet sich, um das vom Benutzer ausgewählte Element im Bearbeitungsmodus anzuzeigen.		
Ereignis	Der Benutzer klickt auf die Schaltfläche „Element bearbeiten“.		
Vorbedingungen	Der Benutzer ist registriert und angemeldet.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf die Schaltfläche „Element bearbeiten“ eines Elements in der Liste. 2. Ein modales Fenster öffnet sich, um die Informationen zum ausgewählten Element anzuzeigen. 3. Nach den Änderungen kann der Benutzer das Element speichern, indem er auf die Schaltfläche „Speichern“ klickt. (FAFE-006) 		
Alternativablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf die Schaltfläche „Element bearbeiten“ eines Elements in der Liste. 2. Ein Modal Fenster öffnet sich, um die Informationen zum ausgewählten Element anzuzeigen. 3. Eine Fehlermeldung wird angezeigt, die dem Benutzer die Ursache des Problems mitteilt. 4. Der Benutzer kann die Änderungen abbrechen, indem er auf das Schließen-Symbol des Modal Fensters klickt. 		
Nachbedingungen Erfolg	Die Änderungen werden gespeichert, der Benutzer erhält eine Bestätigungsnachricht und wird zurück zur Elementliste geleitet.		
Nachbedingungen Fehler	Eine Fehlermeldung wird angezeigt und der Benutzer wird automatisch zum Modal Fenster oder zur Elementliste zurückgeleitet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Delete Item	ID	FAFE-021
-------------	-------------	-----------	----------

Ziel	Ein Modal Fenster öffnet sich, um die Löschung des vom Benutzer ausgewählten Elements zu bestätigen.		
Ereignis	Der Benutzer klickt auf die Schaltfläche „Löschen“.		
Vorbedingungen	Der Benutzer ist registriert und angemeldet.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf die Schaltfläche „Löschen“ eines Elements in der Liste. 2. Ein Modal Fenster öffnet sich, um die Löschung des ausgewählten Elements zu bestätigen. 3. Nach der Bestätigung wird das Element gelöscht. (FABE-007) 		
Alternativablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf die Schaltfläche „Löschen“ eines Elements in der Liste. 2. Ein Modal Fenster öffnet sich, um die Löschung des ausgewählten Elements zu bestätigen. 3. Eine Fehlermeldung wird angezeigt, die dem Benutzer die Ursache des Problems mitteilt. 4. Der Benutzer kann die Löschung abbrechen, indem er auf das Schließen-Symbol des Modal Fensters oder auf die Schaltfläche „Löschung abbrechen“ klickt. 		
Nachbedingungen Erfolg	Das ausgewählte Element wird gelöscht und der Benutzer wird zurück zur Elementliste geleitet.		
Nachbedingungen Fehler	Eine Fehlermeldung wird angezeigt und der Benutzer wird automatisch zur Elementliste zurückgeleitet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	Export report	ID	FAFE-022
Ziel	Ein Modal Fenster öffnet sich, um den Export des vom Benutzer ausgewählten Berichts und die Erstellung einer zugehörigen Rechnung zu bestätigen.		
Ereignis	Der Benutzer klickt auf die Schaltfläche „Bericht exportieren“.		
Vorbedingungen	Der Benutzer ist registriert und angemeldet. Der Benutzer befindet sich auf den "Invoices"-Seite.		

Standardablauf	<ol style="list-style-type: none"> 1. Der Benutzer klickt auf die Schaltfläche „Rechnung generieren“ eines der Elemente in der Liste. 2. Eine Rechnung wird generiert und gespeichert. (FABE-011) 3. Der Benutzer erhält eine Bestätigungsnachricht und die Liste der Rechnungen wird aktualisiert. 		
Alternativablauf	Kein		
Nachbedingungen Erfolg	Eine Rechnung wird generiert und der Benutzer kann sie öffnen/herunterladen.		
Nachbedingungen Fehler	Eine Fehlermeldung wird angezeigt und der Benutzer wird automatisch zur Seite „Rechnungen“ zurückgeleitet.		
Klassifizierung	Funktional, MUSS	Aufwand	MITTEL

Back-End Anforderungen

Name	API Login	ID	FABE-001
Ziel	Anmeldeinformationen über die Authentifizierungs-API prüfen		
Ereignis	Das Backend erhält eine Anfrage vom Frontend mit den Anmeldeinformationen des Benutzers.		
Vorbedingungen	Kein		
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Login-Anfrage mit den Anmeldeinformationen des Benutzers. 2. Nach Überprüfung der Anmeldeinformationen erstellt der Server ein "Access Token" sowie ein "Refresh Token" und sendet sie zusammen mit einer Erfolgsbenachrichtigung an das Frontend. 		
Alternativablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Login-Anfrage mit den Anmeldeinformationen des Benutzers. 2. Falls die Anmeldeinformationen falsch sind, sendet der Server eine Fehlermeldung an das Frontend. 		
Nachbedingungen Erfolg	Ein Zugriffstoken und ein Refresh-Token werden an das Frontend gesendet.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	HOCH

Name	API Logout	ID	FABE-002
Ziel	Die Sitzung des Benutzers über die Authentifizierungs-API schliessen		

Ereignis	Das Backend erhält eine Anfrage vom Frontend mit ungültigen Sitzungs-Token.		
Vorbedingungen	Falls bei einer Anfrage die Zugriffstoken und Refresh-Token nicht mehr gültig sind, sendet der Backend-Server eine Fehlermeldung.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Abmeldeanfrage vom Frontend. 2. Das Backend prüft die übermittelten Token und lehnt die Anfrage ab, falls sie ungültig sind. 		
Alternativablauf	Kein		
Nachbedingungen Erfolg	Der Benutzer wird abgemeldet.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	API Register	ID	FABE-003
Ziel	Benutzerinformationen über die Authentifizierungs-API speichern.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend mit den Registrierungsinformationen des Benutzers.		
Vorbedingungen	Kein		
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Registrierungsanfrage mit den Benutzerinformationen. 2. Das Backend speichert die Benutzerinformationen und erstellt ein zugehöriges Benutzerprofil. 3. Eine Erfolgsbenachrichtigung wird an das Frontend 		

	gesendet.		
Alternativablauf	1. Falls die Informationen ungültig sind, wird nichts gespeichert und eine Fehlermeldung an das Frontend gesendet.		
Nachbedingungen Erfolg	Die Benutzerinformationen werden gespeichert und ein Benutzerprofil wird erstellt.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	MITTEL

Name	API Item creation	ID	FABE-004
Ziel	Ein vom Benutzer erstelltes Item speichern.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Erstellung eines neuen Items.		
Vorbedingungen	Der Benutzer ist registriert und sendet ein gültiges Token mit der Anfrage.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Anfrage zur Erstellung eines neuen Items. 2. Das Backend prüft die Benutzerrechte und speichert das Item in der Datenbank. 3. Eine Erfolgsbenachrichtigung wird an das Frontend gesendet. 		
Alternativablauf	<ol style="list-style-type: none"> 1. Falls das Token ungültig oder abgelaufen ist, wird die Anfrage abgelehnt. 2. Falls die gesendeten Daten ungültig sind, wird eine Fehlermeldung zurückgegeben. 		
Nachbedingungen	Das Item wird gespeichert und dem Benutzer zugeordnet.		

Erfolg			
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	API Item read	ID	FABE-005
Ziel	Ein vom Benutzer erstelltes Item abrufen.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Anzeige eines existierenden Items.		
Vorbedingungen	Der Benutzer ist registriert und sendet ein gültiges Token mit der Anfrage.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Anfrage zur Anzeige eines Items. 2. Der Backend-Server erhält eine Anfrage zur Anzeige eines Items. 3. Die Informationen zum Item werden an das Frontend zur Anzeige gesendet. 		
Alternativablauf	<ol style="list-style-type: none"> 1. Falls das Token ungültig oder abgelaufen ist, wird die Anfrage abgelehnt. 2. Falls das Item nicht existiert, wird eine Fehlermeldung zurückgegeben. 		
Nachbedingungen Erfolg	Das Item wird erfolgreich an das Frontend gesendet.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	API Item update	ID	FABE-006
Ziel	Ein vom Benutzer geändertes Item aktualisieren.		

Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Aktualisierung eines existierenden Items.
Vorbedingungen	Der Benutzer ist registriert und besitzt Bearbeitungsrechte.
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Aktualisierungsanfrage. 2. Das Backend prüft die Benutzerrechte und aktualisiert das Item in der Datenbank. 3. Eine Erfolgsbenachrichtigung wird an das Frontend gesendet.
Alternativablauf	<ol style="list-style-type: none"> 1. Falls das Token ungültig ist, wird die Anfrage abgelehnt.
Nachbedingungen Erfolg	Das Item wird erfolgreich aktualisiert.
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.
Klassifizierung	<div>Funktional, MUSS</div> <div>Aufwand TIEF</div>

Name	API Item delete	ID	FABE-007
Ziel	Ein vom Benutzer ausgewähltes Item löschen.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Löschung eines existierenden Items.		
Vorbedingungen	Der Benutzer ist registriert und besitzt Löschrechte.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Löschanfrage. 2. Das Backend prüft die Benutzerrechte und löscht das Item aus der Datenbank. 		

	3. Eine Erfolgsbenachrichtigung wird an das Frontend gesendet.		
Alternativablauf	1. Falls das Token ungültig ist, wird die Anfrage abgelehnt. 2. Falls das Item nicht existiert, wird eine Fehlermeldung zurückgegeben.		
Nachbedingungen Erfolg	Das Item wird erfolgreich gelöscht.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

Name	API Filter and sort	ID	FABE-008
Ziel	Daten nach Benutzeranforderungen filtern und sortieren.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Anzeige von Items mit Filter- und Sortierparametern.		
Vorbedingungen			
Standardablauf	1. Der Backend-Server erhält eine Anfrage mit Filter- und Sortierparametern. 2. Das Backend verarbeitet die Anfrage und filtert/sortiert die Daten entsprechend. 3. Die gefilterten und sortierten Daten werden an das Frontend gesendet.		
Alternativablauf	1. Falls ungültige Parameter übermittelt werden, wird eine Fehlermeldung zurückgegeben.		
Nachbedingungen Erfolg	Die gefilterten und sortierten Daten werden erfolgreich zurückgegeben.		

Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Non-funktional, MUSS	Aufwand	MITTEL

Name	Inventory quantity update	ID	FABE-009
Ziel	Automatische Aktualisierung der Inventarmengen bei der Erstellung/Aktualisierung eines Berichts.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Erstellung oder Aktualisierung eines Berichts mit zugehörigen Materialien.		
Vorbedingungen			
Standardablauf	<ol style="list-style-type: none"> 1. Das Backend erhält eine Anfrage zur Erstellung oder Aktualisierung eines Berichts mit Materialien. 2. Der Bericht wird erstellt, und die verwendeten Materialien werden vom Inventar abgezogen. 		
Alternativablauf	<ol style="list-style-type: none"> 1. Falls ein Fehler auftritt, wird die Aktion abgebrochen und eine Fehlermeldung an das Frontend gesendet. 		
Nachbedingungen Erfolg	Die Inventarmengen werden aktualisiert.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Non-funktional, KANN	Aufwand	MITTEL

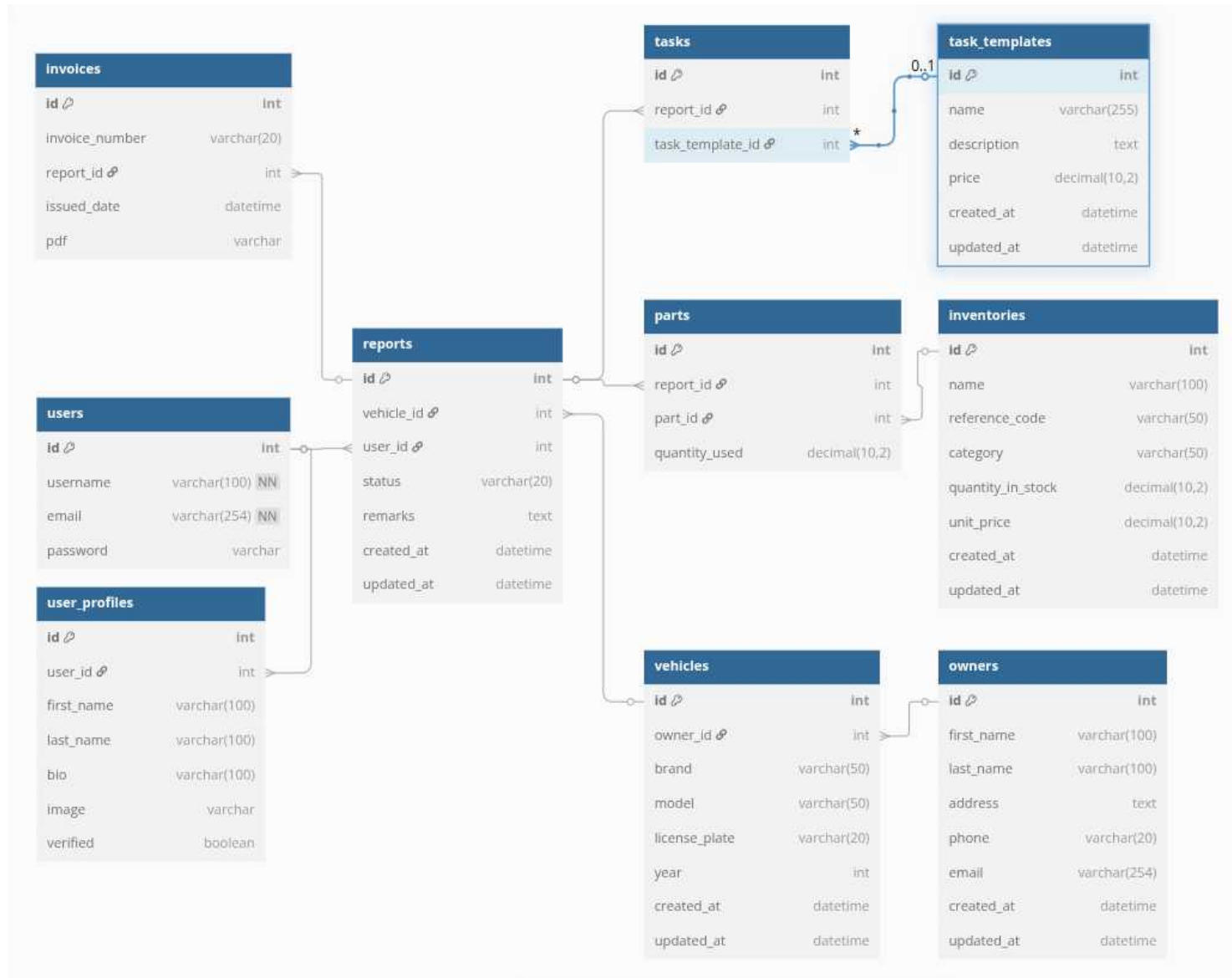
Name	Inventory quantity delete	ID	FABE-010
Ziel	Automatische Aktualisierung der Inventarmengen bei der Aktualisierung/Löschung eines Berichts mit Materialien.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Löschung oder Aktualisierung eines Berichts mit zugehörigen Materialien.		

Vorbedingungen	Ein Bericht mit Materialien existiert und ist gespeichert.		
Standardablauf	<ol style="list-style-type: none"> 1. Das Backend erhält eine Anfrage zur Aktualisierung oder Löschung eines Berichts mit Materialien. 2. Der Bericht wird aktualisiert oder gelöscht und nicht verwendete Materialien werden dem Inventar wieder hinzugefügt. 		
Alternativablauf	<ol style="list-style-type: none"> 1. Falls ein Fehler auftritt, wird die Aktion abgebrochen und eine Fehlermeldung an das Frontend gesendet. 		
Nachbedingungen Erfolg	Die Inventarmengen werden aktualisiert.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	TIEF

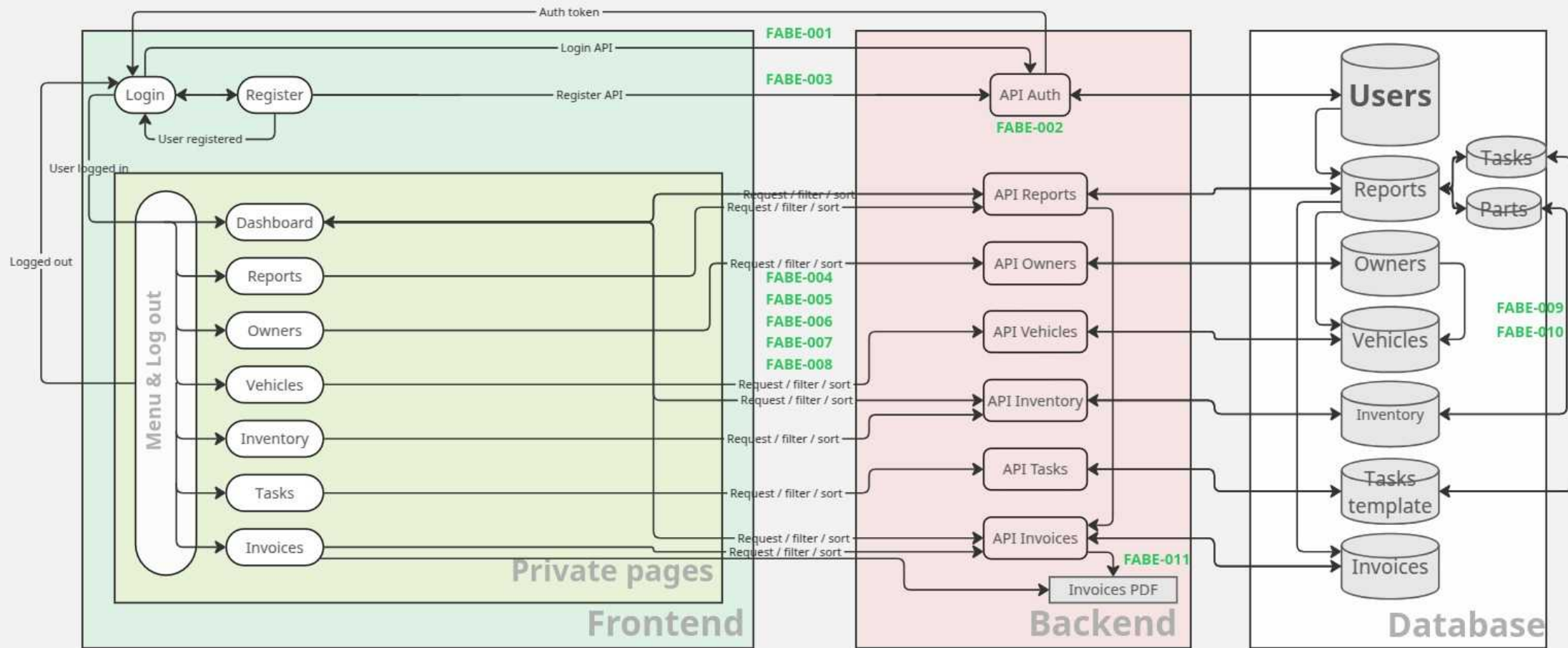
Name	Generate invoice	ID	FABE-011
Ziel	Automatische Erstellung einer PDF-Rechnung beim Export eines Berichts.		
Ereignis	Das Backend erhält eine Anfrage vom Frontend zur Generierung einer PDF-Rechnung.		
Vorbedingungen	Ein Bericht existiert und ist gespeichert und der Benutzer wird den "Status" zu "exported" ändern.		
Standardablauf	<ol style="list-style-type: none"> 1. Der Backend-Server erhält eine Aktualisierungsanfrage für "Status" zu "exported" ändern. 2. Das Backend generiert eine PDF-Rechnung basierend auf den gespeicherten Berichtsdaten. 3. Die Rechnung wird dem Benutzer zur Verfügung gestellt. 		

Alternativablauf	1. Falls der Bericht fehlerhafte oder unvollständige Daten enthält, wird eine Fehlermeldung zurückgegeben.		
Nachbedingungen Erfolg	Die PDF-Rechnung wird erfolgreich generiert.		
Nachbedingungen Fehler	Eine Fehlermeldung wird an das Frontend gesendet.		
Klassifizierung	Funktional, MUSS	Aufwand	MITTEL

MySQL Datenbank

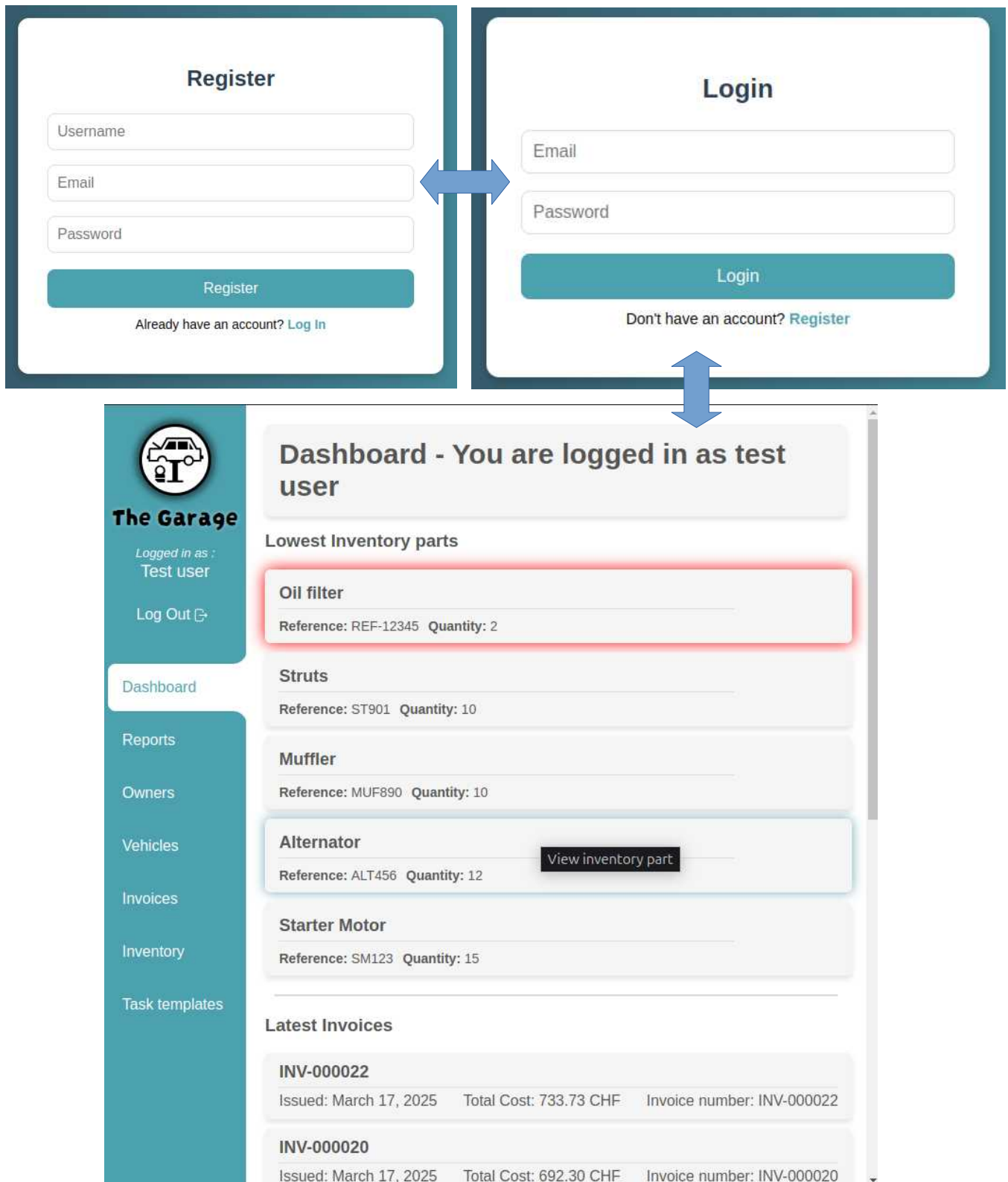


Flowchart



UI/UX AND TESTING

Screenshots



FILTER AND SORT

NOTIFICATIONS

MENU



The Garage

Logged in as :
Test user

Log Out

Dashboard

Reports

Owners

Vehicles

Invoices

Inventory

Task templates

+

Vehicle

Search for a vehicle

Owner

All Owner

Users

All Users

Status

All Status

Success

Report updated successfully!

BMW X3 (795 9RW)

Created Date: March 17, 2025

Owner: Terri Hurst

Status: In Progress

Created By: test user

Edit

Delete

BMW X5 (4TR T66)

Created Date: March 17, 2025

Owner: Vicki Quinn

Status: Completed

Created By: walternelson

Edit

Delete

Citroen C4 (M27 0EY)

Created Date: March 17, 2025

Owner: Alexa Day DVM

Status: Pending

Created By: walternelson

Edit

Delete

Dacia Sandero (362 KS2)

Created Date: March 17, 2025

Owner: Allen Davis

Status: Pending

Created By: kcarroll

Edit

Delete

Ford Focus (UVB Y64)

Created Date: March 17, 2025

Owner: Dana Phillips

Status: Pending

Created By: sryan

Edit

Delete

Ford Focus (UVB Y64)

Created Date: March 17, 2025

Owner: Dana Phillips

Status: Completed

Created By: nrivera

Edit

Delete

ITEM CREATION

ITEM LIST

Page 34 of 40

CLOSE / CANCEL

Create Report

FORMULAR

Vehicle:

Select a vehicle

Status:

Pending

Remarks:

Enter remarks

Repair Tasks

Select a repair task

Add Task

No tasks available

Repair Parts

Select a repair part

1

Add Part

No parts available

Create Report

ACTIONS

Confirm Deletion

Are you sure you want to delete this report?

Yes, Delete

Cancel

Auflösung und Responsivität

Meine Anwendung wird ausschließlich auf Computern oder Tablets verwendet (bereitgestellt vom Arbeitsplatz). Daher werden unterstützte Auflösungen zwischen 768x1024 und höher liegen, mit einer Nutzung im Querformat.

Automated testing

Unit- und Integrationstests werden in Django mit dem Standard-Framework „unittest“ durchgeführt.

Unit- und Integrationstests werden in React mit der Bibliothek „React Testing Library“ und „Vitest“ durchgeführt.

Docker

Die Anwendung ist containerisiert mit Docker-Containern – für den MySQL-Server, das Backend und das Frontend – sowie einer *Docker-compose*-Datei zur Koordination aller Komponenten.

Zwei *Docker-compose*-Dateien ermöglichen das alternative Starten der Container entweder im Entwicklungsmodus (zum Ausführen der Tests) oder im Produktionsmodus (bereitgestellt über einen NGINX-Webserver).

Accessibility

Um die Zugänglichkeit der Anwendung zu gewährleisten, habe ich die Empfehlungen der Web Content Accessibility Guidelines (WCAG) befolgt.

Ich habe semantisches HTML und bei Bedarf ARIA-Rollen verwendet. Die Navigation ist über die Tastatur mittels Tab/Shift-Tab möglich.

Ich habe zudem sichergestellt, dass ein ausreichender Kontrast die Lesbarkeit der Inhalte ermöglicht und dass alle Bilder – sofern erforderlich – mit einem „alt“-Attribut versehen sind.

Eine Eingabehilfe wurde implementiert, um den Benutzer beim Ausfüllen von

Formularen zu unterstützen (Überprüfung der Eingabefelder, automatische Korrektur, Fehlermeldungen und Kennzeichnung von Pflichtfeldern).

Light/Dark mode

Ein Umschalter für den Hell-/Dunkelmodus wurde implementiert, um die Anwendung an die Präferenzen der Benutzer anzupassen.

Der Modus kann unabhängig von den Systemeinstellungen ausgewählt werden.

DEPLOYMENT AND MAINTENANCE

Deploy to VPS with NGINX

Die Anwendung kann auf jedem beliebigen Server (physisch oder virtuell) bereitgestellt werden. Ich habe meine Anwendung auf einem virtuellen privaten Server (VPS) deployiert.

Die Anwendung läuft in Docker und wird über einen NGINX-Webserver ausgeliefert.

Monitor performance and security

Performance

- Verwendung von Database Indexes auf häufig abgefragten Feldern (z.B. Fahrzeug-ID, Kunden-ID).
- Paginierung für Listenansichten (z.B. Rechnungen, Berichte) zur Reduzierung der Antwortzeiten.
- Performance-Optimierung in React durch Memoization und Lazy Loading von Komponenten.
- Setzen von `DEBUG=False` in Django für die Produktion, um Debugging-Overhead zu vermeiden.
- Einsatz von Gunicorn mit automatischer Skalierung der Worker unter NGINX.

Security

- HTTPS-Erzwingung über NGINX (mit Let's Encrypt).
- Containerisierung / Isolation der Betriebssysteme.
- Verwendung von Django-Sicherheitsmiddleware (z.B. `SecurityMiddleware`, `XContentTypeOptionsMiddleware` usw.).
- Aktivierter CSRF-Schutz bei API-Aufrufen.
- Starke Passwortregeln.

- JWT- oder Token-Ablauf und Rotation.
- MySQL-Benutzerberechtigungen auf notwendige Operationen beschränken – keine globalen Rechte (zukünftige Erweiterung).
- Regelmäßige Sicherheitsupdates mit automatisierten Tools.

User feedback

Feedback Mechanism (future enhancement):

- Feedback-Formular im Dashboard zur Meldung von Fehlern oder zur Abgabe von Verbesserungsvorschlägen.
- Erhebung des Net Promoter Score (NPS) über ein Modal alle paar Monate.

Usage Metrics (future enhancement):

- Verwendung von Google Analytics zur Verfolgung der Feature-Nutzung.
- Backend-Logging von Aktionen (z.B. Berichtserstellung, Rechnungserstellung) für Audits und Nutzungseinblicke.

Admin Dashboard (future enhancement):

- Anzeige von Trends (z.B. meistgenutzte Teile, ausgelastetster Mechaniker).
- Anzeige von Fehlerberichten oder kritischen Logs.

Future updates

- **Role-based access control (RBAC)**, um zwischen Mechanikern, Sekretärinnen und Admins zu unterscheiden.
- **Statistics dashboard** (z.B. Zeit pro Aufgabe, verwendete Teile, Rechnungsumsätze).
- **Multi-language support** (starting with DE and EN).
- **Mobile-friendly interface** for on-the-go updates.

- **Interner Live-Chat** zwischen App-Nutzern.
- **Progressive Web App (PWA)** für Offline-Fähigkeit.
- **Two-factor authentication (2FA)** für sicheren Login.
- **Backup & Restore Concept**
 - Backup-Umfang: Es wird ein vollständiger Dump der MySQL-Datenbank erstellt (``mysqldump``), der alle Tabellen, Daten und Strukturen enthält.

Alle generierten Rechnungsdateien, die lokal im Ordner ``/app/invoices/`` gespeichert werden, werden archiviert.
 - Zeitpunkt: Ein täglicher Cronjob führt das Backup automatisch um 03:00 Uhr morgens durch.
 - Ablageort: Die Datenbank-Dumps (``sql.gz``) und Rechnungsarchive (``tar.gz``) werden in einem separaten Docker Volumen gespeichert. Zusätzlich wird eine verschlüsselte Kopie per ``rclone`` auf ein externes Nextcloud-Backup-System hochgeladen.
 - Speicherstrategie: Die letzten 7 Tage werden aufbewahrt, ältere Backups werden automatisch gelöscht.
 - Wiederherstellung: Die Wiederherstellung erfolgt durch Ausführen des Skripts ``restore-db.sh`` und ``restore-invoices.sh``, das den letzten Dump importiert.
 - Validierung: Nach jedem Backup-Lauf wird ein Log-Eintrag erstellt. Bei Fehlern wird eine Benachrichtigung an den Administrator gesendet.