

UAV-Manager

Beschreibung der zu realisierende Applikation

Der UAV-Manager ist eine spezialisierte Softwarelösung, die darauf abzielt, die Verwaltung von Flugstunden für Piloten und unbemannte Fluggeräte (UAVs), wie z. B. Drohnen, zu vereinfachen und zu optimieren. Im Gegensatz zu herkömmlichen Softwarelösungen für die bemannte Luftfahrt ist der UAV-Manager speziell auf die Bedürfnisse von UAV-Betreibern zugeschnitten. Die Applikation bietet eine umfassende Plattform, um Fluggeräte zu erfassen, zu verwalten und die damit verbundenen Flugstunden detailliert zu überwachen.

Kernfunktionen der Applikation:

1. Fluggeräteverwaltung:

- Benutzer können unbemannte Fluggeräte (z. B. Drohnen) erfassen und verwalten.
- Bei der Erfassung stehen sowohl vordefinierte Attribute (z. B. Modell, Seriennummer, Hersteller) als auch freie Textfelder zur Verfügung, um individuelle Informationen hinzuzufügen.
- Jedes Fluggerät verfügt über eine detaillierte Ansicht, in der alle relevanten Daten und Statistiken übersichtlich dargestellt werden.

2. Flugstundenmanagement:

- Piloten können ihre Flugstunden erfassen und verwalten.
- Die Applikation bietet eine Gesamtübersicht aller registrierten Flugstunden, die nach verschiedenen Kriterien gefiltert und durchsucht werden kann.
- Flugstunden können sowohl manuell eingegeben als auch per CSV-Import hinzugefügt werden.

3. Statistiken und Analysen:

- Für jedes Fluggerät werden detaillierte Statistiken angezeigt, z. B. die Gesamtflugzeit oder die Anzahl der Landungen.
- Es werden die letzte Wartung an Propeller, Motor und Frame sowie der Termin für die nächste Wartung angezeigt, sofern diese definiert wurde.
- Diese Statistiken helfen Betreibern, den Zustand und die Nutzung ihrer Fluggeräte besser zu verwalten.

4. Protokollierung von Ereignissen und Wartung:

- Im Fluggerätebereich können Benutzer Ereignisse (z. B. Flugunfälle oder besondere Vorkommnisse) und Wartungshinweise erfassen.
- Dieses Protokoll dient als zentrales Nachschlagewerk für die Historie jedes Fluggeräts und unterstützt die Wartungshistorie.

5. Benutzerverwaltung:

- Benutzer müssen sich registrieren und ein persönliches Konto anlegen.
- Jeder Benutzer hat Zugriff ausschliesslich auf seine eigenen Daten, was die Datensicherheit und den Datenschutz gewährleistet.
- Administratorrechte sind in dieser Version nicht vorgesehen, um die Komplexität der Applikation gering zu halten.

6. Einfache Bedienbarkeit:

- Die Benutzeroberfläche ist intuitiv gestaltet und auf die Bedürfnisse von UAV-Betreibern zugeschnitten.
- Die Navigation zwischen den verschiedenen Ansichten (z. B. Fluggeräteverwaltung, Flugstundenübersicht) ist einfach und logisch.

Einschränkungen:

- Die Applikation dient **nicht** zur **Navigationsplanung** oder **Flugroutenberechnung**.
- Sie ist ausschliesslich auf die Verwaltung von Fluggeräten und Flugstunden ausgerichtet.

Zielgruppe

Die Applikation richtet sich primär an:

- **UAV-Betreiber**, die mehrere Drohnen oder unbemannte Fluggeräte verwalten müssen.
- **Piloten**, die ihre Flugstunden dokumentieren und analysieren möchten.
- **Unternehmen**, die UAVs für geschäftliche Zwecke einsetzen und eine effiziente Verwaltung ihrer Flotte benötigen.

Technologie Stack

Folgende Technologien werden verwendet:

Kategorie	Technologie / Tool	Beschreibung
Frontend	React	Frontend-Framework für die Benutzeroberfläche
Frontend	react-leaflet	Frontend-Framework für Kartenfunktion
Backend	Django	Backend-Framework für Logik und Datenverarbeitung
Datenbank	PostgreSQL	Relationale Datenbank zur Speicherung von Flugdaten
Styling	Tailwind CSS	CSS-Framework für modernes und flexibles Styling
Testing	Jest (React), Pytest (Django)	Testing-Frameworks für Unit- und Integrationstests
Deployment	Docker	Containerisierung
Versionierung	Git	Versionskontrolle für Codeverwaltung
API Schnittstellen	REST API, Django REST Framework	API Architektur zur Kommunikation zwischen Frontend und Backend
Backend	psycopg2-binary	Adapter für schnittstelle Django PostgreSQL
Entwicklungsumgebung	Visual Studio Code	Hauptwerkzeuge für die Entwicklung
Backend	Djoser, simplejwt	Login & Token API Framework

Anforderungen

Kurze Auflistung der MUSS und KANN Anforderungen:

Muss

- FAFE-01 – Benutzer Registrierung
- FAFE-02 – Daten gelangen vom Frontend ins Backen
- FAFE-03 – Daten gelangen vom Backend ins Frontend
- FAFE-04 – Benutzer Anmeldung
- FAFE-05 – Hauptansicht Pilot Log
- FAFE-06 – Pilot Log kann gefiltert werden
- FAFE-07 – Manuelle Flightlog-Erfassung
- FAFE-08 – Maske für Benutzerverwaltung
- FAFE-09 – Erfassungsformular für Fluggeräte
- FAFE-10 – Maske mit Auflistung der Fluggeräte
- FAFE-11 – Maske pro Fluggerät mit Übersicht über alle Attribute
- FAFE-12 – Statistik pro Fluggerät anzeigen

Kann

- FAFE-0-01 – Standort auf Karte anzeigen
- FAFE-0-02 – CSV-Importfunktion für Flugstunden
- FAFE-0-03 – Bericht-Exportfunktion für Flugstunden
- FAFE-0-04 – Files können an Fluggeräte angehängt werden
- FAFE-0-05 – Wartungs Alerts können gesetzt werden

(Bemerkung: FAFE - Funktionale Anforderung Front-End)

(Bemerkung: FAFE - Funktionale Anforderung Back-End)

Spezifikation der Anforderungen Muss Kriterien:

* b.P. steht für: benutzende Person

Name	Sprechender Name
ID	FAFE-01 - Benutzer Registrierung
Ziel	Neue Benutzer können sich registrieren
Ereignis	Der Button auf der Login-Seite (FAFE-03 – Benutzeranmeldung) führt zur Registrierungsmaske.
Vorbedingung	<ol style="list-style-type: none"> 1. Die b.P.* befindet sich auf einer Login Seite. 2. Die b.P.* hat sich noch nicht eingeloggt.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P.* besucht die Login Maske 2. Die b.P.* klickt auf Registrieren 3. Die b.P.* füllt alle Felder aus 4. Die b.P.* klickt auf "Senden", die Anmeldung wird an das Backend geschickt und in die Datenbank geschrieben. (FABE-02)
Alternativablauf	Die b.P.* bricht den Vorgang ab, indem sie zurückgeht oder den Browser schliesst.
Nachbedingungen Erfolg	Die b.P. erhält eine Bestätigungsmeldung und wird zur Hauptansicht des Pilot Logs (FAFE-04) weitergeleitet.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung mit Hinweis auf die auszufüllenden Felder.
Klassifizierung	Funktional, MUSS
Aufwand	EINFACH

Name	Sprechender Name
ID	FABE-02 - Daten gelangen vom Frontend ins Backen
Ziel	Die Benutzerdaten werden an das Backend übermittelt und in einer DB gespeichert.
Ereignis	Durch einen Klick auf den Button werden die Benutzerdaten in eine Datenbank geschrieben.
Vorbedingung	<ul style="list-style-type: none"> • Daten wurden in der Input box eingegeben.
Standardablauf	<ul style="list-style-type: none"> • Daten werden via ORM in die DB geschrieben.
Alternativablauf	Falls Daten nicht in die DB geschrieben werden können, sollte eine Fehlermeldung angezeigt werden.
Nachbedingungen Erfolg	Die Daten wurden in die DB eingetragen und b.P. wird zur entsprechenden Seite weitergeleitet.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FABE-03 – Daten gelangen vom Backend ins Frontend
Ziel	Daten könne überprüft werden und angezeigt werden.
Ereignis	Nach der Eingabe von Daten, wie z. B. Anmeldedaten FAFE-04 oder einer Abfrage für bestimmte Benutzerdaten, werden die entsprechenden Daten im Frontend angezeigt.
Vorbedingung	<ul style="list-style-type: none"> • Daten müssen in der DB vorhanden sein.
Standardablauf	<ul style="list-style-type: none"> • Daten sind vorhanden und können abgerufen werden.
Alternativablauf	Falls die Daten in der DB nicht vorhanden sind, soll eine Fehlermeldung angezeigt werden.
Nachbedingungen Erfolg	Die Daten wurden gefunden und erfolgreich an das Frontend übergeben.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-04 – Benutzer Anmeldung
Ziel	Die b.P. erhält eine Anmeldeseite zur Eingabe seiner Login-Daten.
Ereignis	Nach der Eingabe der Login-Daten werden diese an das Backend zur Verifizierung gesendet, um anschliessend die persönlichen Informationen anzeigen zu können.
Vorbedingung	<ul style="list-style-type: none"> • Anmeldefelder müssen korrekt und mit gültigen Daten ausgefüllt werden.
Standardablauf	<ol style="list-style-type: none"> 1. Die Daten werden in einer Anmeldemaske eingegeben und nach dem Klicken auf den Anmeldebutton an das Backend gesendet. 2. Das Backend verifiziert die Daten und gibt die entsprechenden Ergebnisse ans Frontend zurück. FAFE-03
Alternativablauf	Falls die Daten nicht in der DB vorhanden sind, soll eine Fehlermeldung angezeigt werden.
Nachbedingungen Erfolg	Die Daten stimmen mit den Daten in der DB überein, und Die b.P wird zur Hauptansicht des Pilot Logs (FAFE-04) weitergeleitet.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung.
Klassifizierung	Funktional, MUSS
Aufwand	EINFACH

Name	Sprechender Name
ID	FAFE-05 – Hauptansicht Pilot Log
Ziel	In der Hauptansicht soll das Logbuch des eingeloggten Piloten erscheinen. Dieses muss folgendes enthalten: <ul style="list-style-type: none"> • Tabelle mit Flugdaten des Piloten • Filterfunktion • Eingabemaske zur Erfassung neuer Flugeinträge. Zusätzlich soll von der Hauptansicht auf weitere Tabs navigiert werden können.
Ereignis	Nach dem erfolgreichen Login erscheint das Pilot-Log mit den Daten (FABE-03) des eingeloggten Piloten.
Vorbedingung	<ul style="list-style-type: none"> • Die b.P. muss registriert sein. • Das Login muss erfolgreich sein.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P. loggt sich ein und sieht ihre persönlichen Logbuchdaten. 2. Die b.P. ist bereits eingeloggt und gelangt über einen anderen Tab auf die Hauptseite. Die persönlichen Logbuchdaten werden ebenfalls angezeigt.
fAlternativablauf	Wenn noch kein Flug erfasst wurde, ist das Logbuch leer, wird aber trotzdem angezeigt.
Nachbedingungen Erfolg	Die b.P. kann ihre Logbuchdaten einsehen und weiter navigieren.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-06 – Pilot Log kann gefiltert werden
Ziel	Das Pilot-Log kann nach verschiedenen Kriterien gefiltert werden, um spezifische Flugdaten anzuzeigen.
Ereignis	Die b.P wählt Filterkriterien aus, um im Logbuch nach bestimmten Flügen zu durchsuchen.
Vorbedingung	<ul style="list-style-type: none"> • Die b.P. muss angemeldet sein. • Das Pilot-Log muss bereits Flugdaten enthalten.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P. wählt Filterkriterien aus (z.B. Datum, Flugzeugtyp, Flugzeit usw.) 2. Das Logbuch wird entsprechend gefiltert und aktualisiert.
Alternativablauf	Wenn keine Filterkriterien ausgewählt werden, wird das gesamte Logbuch angezeigt.
Nachbedingungen Erfolg	Die b.P. sieht die gefilterten Flugdaten.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung, falls keine Daten den Filterkriterien entsprechen.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-07 – Manuelle Flightlog-Erfassung
Ziel	Piloten können Flugdaten manuell im Logbuch erfassen.
Ereignis	Die b. P. möchte Flugdaten manuell in das Pilot-Log eintragen.
Vorbedingung	<ul style="list-style-type: none"> Die b. P. muss angemeldet sein.
Standardablauf	<ol style="list-style-type: none"> Die b. P. navigiert zur Flightlog-Übersicht oder zur entsprechenden Maske für manuelle Erfassung. Die b. P. füllt die erforderlichen Felder aus (z. B. Datum, Flugzeit, Flugzeugtyp, Start- und Zielflughafen usw.). Die b. P. bestätigt die Eingaben durch Klicken auf "Speichern". Die eingetragenen Daten werden im Pilot-Log gespeichert.
Alternativablauf	Falls die b. P. nicht alle erforderlichen Felder ausfüllt, wird eine Fehlermeldung angezeigt und die Speicherung erfolgt nicht. Eingegabener Text bleibt bestehen.
Nachbedingungen Erfolg	Die neu eingetragenen Daten erscheinen im Pilot-Log.
Nachbedingung Fehler	Falls die Speicherung fehlschlägt (z. B. aufgrund fehlender Pflichtfelder oder technischer Probleme), erhält die b. P. eine Fehlermeldung.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-08 – Maske für Benutzerverwaltung
Ziel	Die Benutzerverwaltungsmaske ermöglicht es registrierten Benutzern, ihre eigenen Daten inklusive Pilotenlizenzen zu verwalten.
Ereignis	Ein registrierter Benutzer möchte seine persönlichen Daten oder Lizenzdaten einsehen, anpassen oder löschen.
Vorbedingung	<ul style="list-style-type: none"> Die b. P. muss angemeldet sein.
Standardablauf	<ol style="list-style-type: none"> Die b.P öffnet die Benutzerverwaltungsmaske über einen entsprechenden Link oder Menüpunkt im System. In der Maske werden die persönlichen Daten des Benutzers (z. B., Name, E-Mail-Adresse, Kontaktdaten) sowie seine Lizenzdaten (Falls vorhanden) angezeigt. Die b.P kann seine persönlichen Daten bearbeiten und aktualisieren. Die b.P kann seine Lizenzdaten verwalten
Alternativablauf	Wenn Die b.P unvollständige Daten eingibt, wird eine Meldung in der entsprechenden Eingabebox angezeigt, und die Speicherung wird blockiert, bis alle Pflichtfelder korrekt ausgefüllt sind.
Nachbedingungen Erfolg	Die aktualisierten Daten werden in der Maske angezeigt.
Nachbedingung Fehler	Falls die Speicherung fehlschlägt (z. B. aufgrund fehlender Pflichtfelder oder technischer Probleme), erhält die b. P. eine Fehlermeldung.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-09 – Erfassungsformular für Fluggeräte
Ziel	Die b.P kann neue Fluggeräte erfassen oder bestehende anpassen und in der Datenbank speichern.
Ereignis	Die b.P möchte ein neues Fluggerät registrieren oder bestehende Daten zu einem Fluggerät aktualisieren.
Vorbedingung	<ul style="list-style-type: none"> Die b. P. muss angemeldet sein.
Standardablauf	<ol style="list-style-type: none"> Die b.P navigiert zur Registerkarte "Fluggeräte" und wählt die Option "Neues Fluggerät registrieren" oder öffnet das Bearbeitungsformular für ein bestehendes Gerät. Die b.P füllt erforderliche Felder wie Flugmodelltyp (Drohne, Flugzeug), Hersteller, Modellname, Technische Spezifikationen usw. aus. Die Eingaben werden validiert: Pflichtfelder sind ausgefüllt. Die Daten werden gespeichert.
Alternativablauf	Falls keine Änderungen vorgenommen wurden, kann die b. P. das Formular abbrechen / Schliessen.
Nachbedingungen Erfolg	Die Fluggerätedaten wurden erfolgreich gespeichert.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung, falls die Daten nicht korrekt eingegeben wurden.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-10 -- Maske mit Auflistung der Fluggeräte
Ziel	Die b.P kann eine Liste aller erfassten Fluggeräte einsehen.
Ereignis	Die b.P öffnet die Maske mit der Auflistung der Fluggeräte.
Vorbedingung	<ul style="list-style-type: none"> Die b. P. muss angemeldet sein. Es müssen bereits Fluggeräte erfasst sein. (Sonst sieht er eine Leere liste)
Standardablauf	<ol style="list-style-type: none"> Die b.P. öffnet die Maske mit der Auflistung der Fluggeräte. Die b.P. sieht eine Liste aller erfassten Fluggeräte.
Alternativablauf	Wenn keine Fluggeräte erfasst sind, wird eine entsprechende Meldung angezeigt.
Nachbedingungen Erfolg	Die b.P. sieht die Liste der erfassten Fluggeräte.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung, falls keine Fluggeräte erfasst sind.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-11 -- Maske pro Fluggerät mit Übersicht über alle Attribute
Ziel	Die b.P kann detaillierte Informationen zu einem bestimmten Fluggerät einsehen.
Ereignis	Die b.P wählt ein Fluggerät aus der Liste aus und sieht die detaillierten Attribute.
Vorbedingung	<ul style="list-style-type: none"> Die b. P. muss angemeldet sein. Es müssen bereits Fluggeräte erfasst sein.
Standardablauf	<ol style="list-style-type: none"> Die b.P. wählt ein Fluggerät aus der Liste aus. Die b.P. sieht die detaillierten Attribute des Fluggeräts.
Alternativablauf	Wenn keine Fluggeräte erfasst sind, wird eine Leere liste angezeigt.
Nachbedingungen Erfolg	Die b.P. sieht die detaillierten Attribute des ausgewählten Fluggeräts.
Nachbedingung Fehler	Die b.P. sieht das keine Fluggeräte erfasst sind.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Name	Sprechender Name
ID	FAFE-12 -- Statistik pro Fluggerät anzeigen
Ziel	Die b.P kann statistische Daten zu einem bestimmten Fluggerät einsehen.
Ereignis	Die b.P wählt ein Fluggerät aus der Liste aus und sieht die statistischen Daten.
Vorbedingung	<ul style="list-style-type: none"> Die b. P. muss angemeldet sein. Es müssen bereits Fluggeräte erfasst sein.
Standardablauf	<ol style="list-style-type: none"> Die b.P. wählt ein Fluggerät aus der Liste aus. Die b.P. sieht die statistischen Daten des Fluggeräts.
Alternativablauf	Wenn Fluggeräte noch nie in Betrieb waren, steht der Zähler auf 0.
Nachbedingungen Erfolg	Die b.P. sieht die statistischen Daten des ausgewählten Fluggeräts.
Nachbedingung Fehler	Die b.P. sieht das keine Fluggeräte erfasst sind.
Klassifizierung	Funktional, MUSS
Aufwand	MITTEL

Spezifikation der Anforderungen Kann Kriterien:

* b.P. steht für: benutzende Person

Name	Sprechender Name
ID	FAFE-0-01 - Lande- oder Startpunkt auf Karte anzeigen
Ziel	Die b.P kann den Lande- oder Startpunkt eines Fluggeräts auf einer Karte anzeigen lassen.
Ereignis	Die b.P klickt auf einen Start- oder Landepunkt und sieht den entsprechenden Punkt auf einer Karte angezeigt.
Vorbedingung	<ol style="list-style-type: none"> 1. Die b.P.* befindet sich auf der Main Page. 2. Es müssen bereits Fluggeräte erfasst sein. 3. Es müssen Flugstunden erfasst sein.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P klickt im Logbuch auf eine Ortschaft (Start oder Landung). 2. Die b.P. sieht den Lande- oder Startpunkt des Fluggeräts auf einer Karte.
Alternativablauf	Es sind noch keine Flüge erfasst und Maske ist leer.
Nachbedingungen Erfolg	Die b.P. sieht den Lande- oder Startpunkt des Fluggeräts auf der Karte.
Nachbedingung Fehler	Die b.P. sieht das keine Flugstunden erfasst sind.
Klassifizierung	Funktional, KANN
Aufwand	HOCH

Name	Sprechender Name
ID	FABE-0-02 -- CSV-Importfunktion für Flugstunden
Ziel	Die b.P kann Flugstunden per CSV-Datei importieren.
Ereignis	Die b.P lädt eine CSV-Datei mit Flugstunden hoch und importiert die Daten.
Vorbedingung	<ol style="list-style-type: none"> 1. Die b.P. muss eingeloggt sein. 2. CSV-Log muss vorhanden sein.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P navigiert zur Main Page. 2. Die b.P. lädt eine CSV-Datei hoch. 3. Die Daten werden importiert und im Logbuch angezeigt.
Alternativablauf	Die b.P. oder das System bricht den Import ab, ohne die Daten zu speichern.
Nachbedingungen Erfolg	Die Flugstunden wurden erfolgreich importiert und im Logbuch angezeigt.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung, falls die CSV-Datei nicht korrekt formatiert ist.
Klassifizierung	Funktional, KANN
Aufwand	HOCH

Name	Sprechender Name
ID	FABE-0-03 -- Bericht-Exportfunktion für Flugstunden
Ziel	Die b.P kann Flugstunden als Bericht exportieren.
Ereignis	Die b.P exportiert die Flugstunden als Bericht.
Vorbedingung	<ol style="list-style-type: none"> 1. Die b.P. muss eingeloggt sein. 2. Es müssen bereits Flugstunden erfasst sein.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P. exportiert die Flugstunden als Bericht mit dem Button in der Main Page.
Alternativablauf	Die b.P. bricht den Export ab, ohne die Daten zu exportieren.
Nachbedingungen Erfolg	Die Flugstunden wurden erfolgreich exportiert.
Nachbedingung Fehler	Die b.P. erhält eine Meldung, falls keine Flugstunden erfasst sind.
Klassifizierung	Funktional, KANN
Aufwand	NIEDRIG

Name	Sprechender Name
ID	FABE-0-04 -- Files können an Fluggeräte angehängt werden.
Ziel	Die b.P kann Dateien an ein Fluggerät anhängen.
Ereignis	Die b.P lädt eine Datei hoch und hängt sie an ein Fluggerät an.
Vorbedingung	<ol style="list-style-type: none"> 1. Die b.P. muss eingeloggt sein. 2. Es müssen bereits Fluggeräte erfasst sein.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P. wählt ein Fluggerät aus gelangt damit auf die Aircraft Settings Website. 2. Die b.P. lädt eine Datei hoch und hängt sie an das Fluggerät an.
Alternativablauf	Die b.P. bricht den Vorgang ab, ohne die Datei zu speichern.
Nachbedingungen Erfolg	Die Datei wurde erfolgreich an das Fluggerät angehängt.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung, falls die Datei nicht korrekt hochgeladen wurde.
Klassifizierung	Funktional, KANN
Aufwand	HOCH

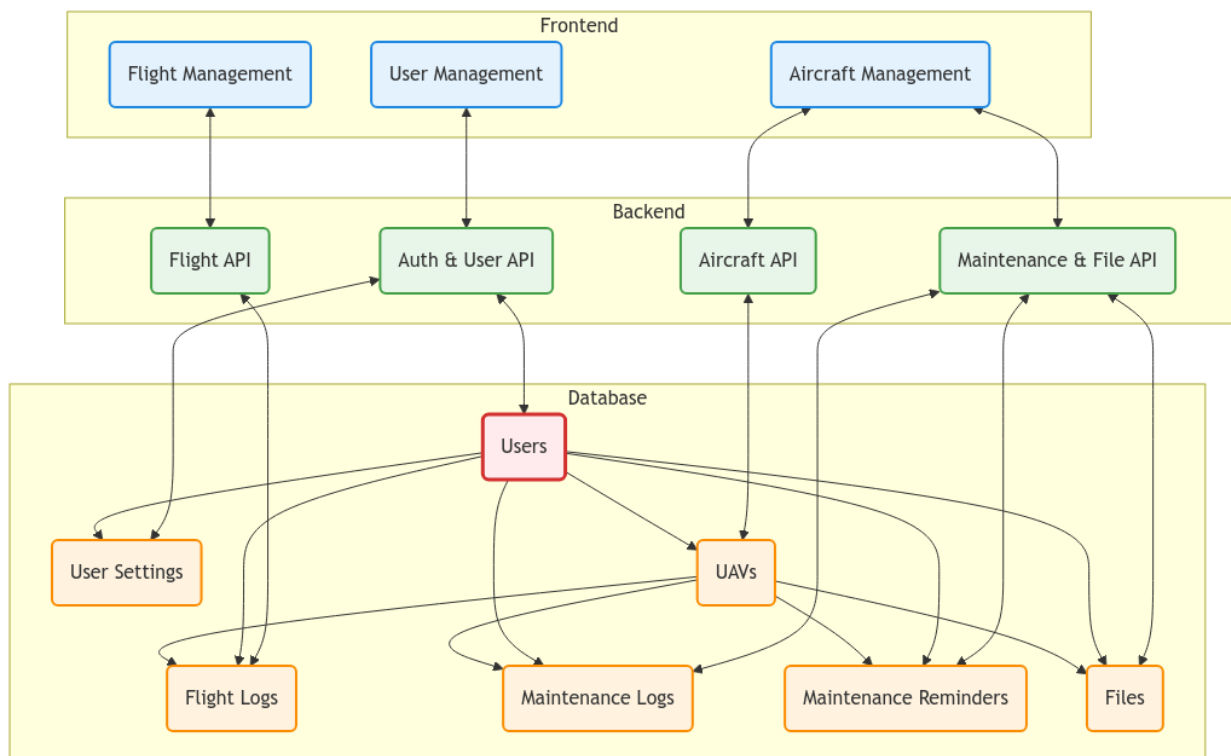
Name	Sprechender Name
ID	FABE-0-05 -- Wartung Alerts können gesetzt werden
Ziel	Die b.P kann Wartung Alerts für Fluggeräte setzen.
Ereignis	Die b.P setzt einen Wartungsalert für ein Fluggerät.
Vorbedingung	<ol style="list-style-type: none"> 1. Die b.P. muss eingeloggt sein. 2. Es müssen bereits Fluggeräte erfasst sein.
Standardablauf	<ol style="list-style-type: none"> 1. Die b.P. wählt ein Fluggerät aus und gelangt damit auf die Aircraft Settings Website. 2. Die b.P. setzt einen Wartungsalert für das Fluggerät.
Alternativablauf	Die b.P. bricht den Vorgang ab, ohne den Alert zu setzen.
Nachbedingungen Erfolg	Der Wartungsalert wurde erfolgreich gesetzt.
Nachbedingung Fehler	Die b.P. erhält eine Fehlermeldung, falls der Alert nicht gesetzt werden konnte.
Klassifizierung	Funktional, KANN
Aufwand	MITTEL

Architektur

Die Architektur des UAV-Managers gliedert sich in drei Hauptbereiche:

1. **Frontend:** Beinhaltet Module für Flug-, Benutzer- und UAV's -verwaltung. Hier erfolgt die Interaktion der Nutzer, z. B. zur Erfassung von Flugdaten, Verwaltung von UAVs oder Anpassung von Benutzereinstellungen.
2. **Backend:** Verarbeitet die Anfragen über spezialisierte APIs:
 - **Flight API** für Flugdatenverwaltung,
 - **Auth & User API** für Authentifizierung und Benutzerverwaltung,
 - **Aircraft API** für Drohnenmanagement,
 - **Maintenance & File API** für Wartungen und Dateiuploads.
3. **Datenbank:** Speichert zentrale Informationen, wie Benutzerdaten, UAV-Details, Flugdaten, Wartungsprotokolle und Dateianhänge. Die Datenstruktur ist so aufgebaut, dass alle Entitäten logisch miteinander verknüpft sind.

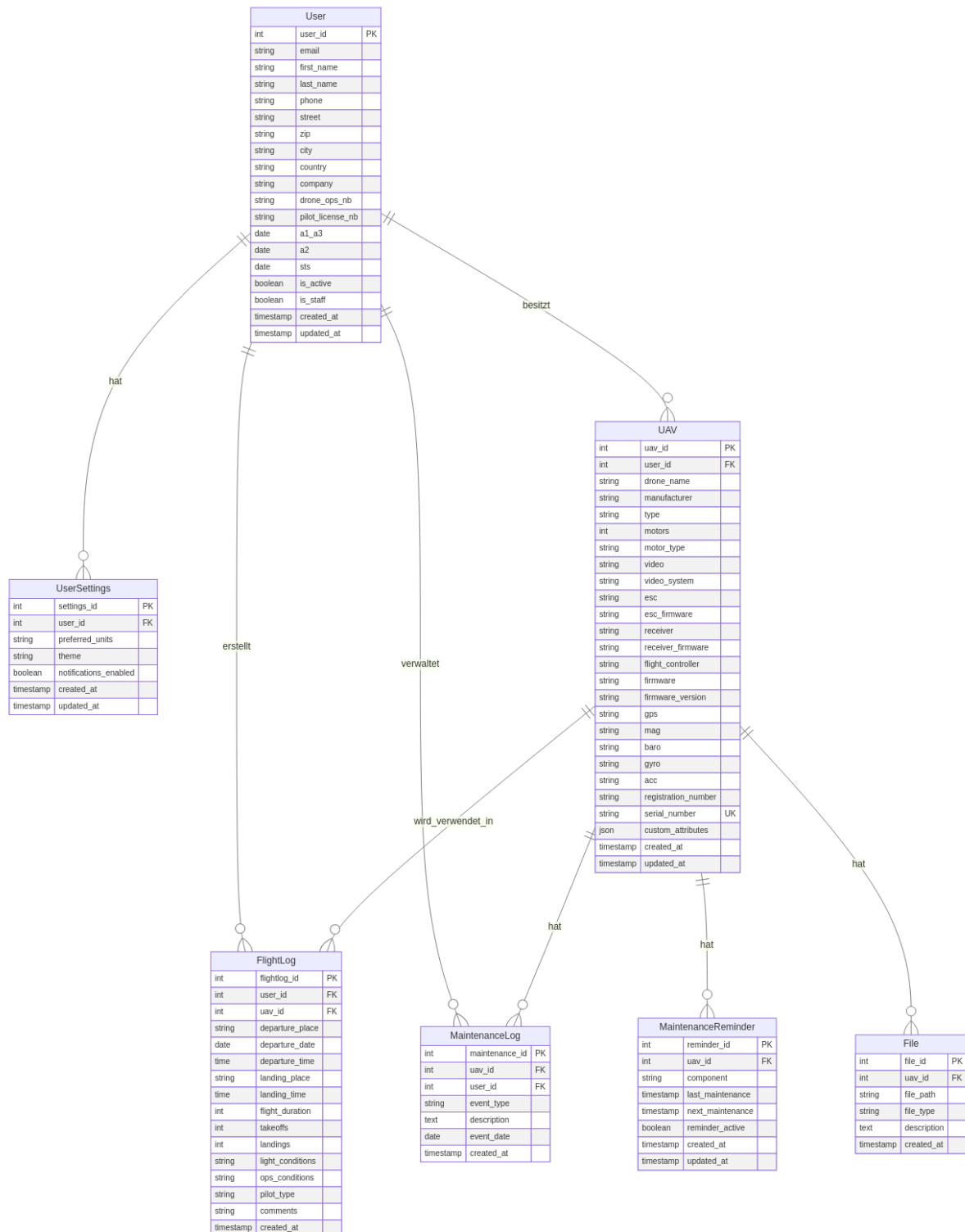
Diese Architektur ermöglicht eine klare Trennung der Verantwortlichkeiten, effiziente Datenverarbeitung und einfache Erweiterbarkeit, beispielsweise für Funktionen wie CSV-Importe, Berichterstellung oder Wartungserinnerungen. **Muss sicherlich noch überprüft werden!**



Datenbankschema

Das Schema visualisiert die vorgesehen Datenbankstruktur und bietet einen klaren Überblick über die Komponenten. Änderungen an der Struktur sind im Laufe des Projekts möglich, um zukünftigen Anforderungen gerecht zu werden. Die Tabellen „User Setting“ und „UVA Maintenance“ wurden bewusst ausserhalb der Kerntabellen „User“ und „UVA“ angelegt, um Anpassungen flexibler und einfacher umsetzen zu können.

Link: [Online FlowChart & Diagrams Editor - Mermaid Live Editor](#)



Filestruktur

Die Projektstruktur des UAV-Managers ist in ein **Django-Backend** für die API-Logik und ein **React-Frontend** mit Tailwind CSS für die Benutzeroberfläche unterteilt. Das Backend übernimmt die Verwaltung von Datenmodellen, Serialisierung und API-Routen, während das Frontend wiederverwendbare Module bereitstellt und direkt mit der API interagiert. Diese Struktur ist ein Entwurf, der sich im Verlauf des Projekts durch neue Anforderungen oder Erkenntnisse weiterentwickeln kann. **Wird gegen End auf die definitive Version angepasst!**

```
uav-manager/
|
|— backend/                                # Django Backend (nur API)
|   |— manage.py
|   |— requirements.txt
|   |— config/
|   |   |— settings.py
|   |   |— urls.py                # Enthält nur API-Routen
|   |— api/                         # Eine einzelne api für alle Endpunkte
|   |   |— models.py              # Datenbank-Modelle (User, FlightLogs, Aircrafts, etc.)
|   |   |— serializers.py         # JSON-Serialisierung
|   |   |— views.py               # API-Logik
|   |   |— urls.py               # API-Routen
|   |— tests/
|
|— frontend/                             # React Frontend mit Tailwind CSS
|   |— public/                          # Statische Dateien
|   |— src/
|   |   |— assets/                # Bilder, Icons|
|   |   |— components/           # Reusable UI-Komponenten
|   |   |   |— Topbar.jsx
|   |   |   |— Sidebar.jsx
|   |   |— pages/                # Seiten
|   |   |   |— Login.jsx
|   |   |   |— Register.jsx
|   |   |   |— Flightlog.jsx      # Hauptseite mit Flightlogs
|   |   |   |— AircraftList.jsx
|   |   |   |— NewAircraft.jsx
|   |   |   |— AircraftSettings.jsx
|   |   |   |— FlightDetails.jsx
|   |   |   |— UserSettings.jsx
|   |   |— api/                  # API-Calls an Django Backend
|   |   |   |— auth.js
|   |   |   |— flightlog.js
|   |   |   |— aircraftlist.js
|   |   |— App.jsx               # Haupt-Komponente
|   |   |— main.jsx              # Einstiegspunkt für Vite
|   |   |— tailwind.config.js    # Tailwind Konfiguration
|   |   |— index.css             # Globales CSS
|   |— vite.config.js            # Vite Konfiguration
|   |— package.json              # Abhängigkeiten
|
|— docker-compose.yml            # Docker für Backend, Frontend und PostgreSQL
|— README.md                     # Dokumentation
|— .env                          # Umgebungsvariablen
```

Roadmap

Phase 1: Vorbereitung und Dokumentation (bis zur 2. PVA am 08.03.2025)

- **Ziel:** Erstellung der Dokumentation, Definition der Anforderungen und Planung der Roadmap.
- **Aufgaben:**
 1. **Anforderungsanalyse:**
 - Finalisierung der MUSS- und KANN-Anforderungen.
 - Priorisierung der Anforderungen (MUSS zuerst, KANN später).
 2. **Technologie-Stack:**
 - Festlegung der Stack Technologien von React, Django, PostgreSQL usw.
 - Einrichtung der Entwicklungsumgebung (z.B. Git-Repository, python venv, vite, Tailwind CSS usw.).
 3. **Projektplanung:**
 - Erstellung eines detaillierten Projektplans mit Meilensteinen und Anforderungen.
 4. **Dokumentation:**
 - Erstellung der Projektdokumentation inkl. Anforderungen, Roadmap und grober Architektur.
 5. **Design:**
 - Erstellung von Wireframes für die Benutzeroberfläche (Login, Register, Hauptseite, Aircrafts settings usw.).

Phase 2: Backend-Entwicklung und Frontend-Grundlagen (bis zur 3. PVA am 05.04.2025)

- **Ziel:** Aufbau des Backends und Beginn der Frontend-Entwicklung (Register und Login).
- **Aufgaben:**
 1. **Backend-Entwicklung:**
 - Einrichtung der Django-Umgebung mit PostgreSQL.
 - Erstellung der Datenbankmodelle für:
 - Benutzer (User)
 - Fluggeräte (UAVs)
 - Flugstunden (Flight Logs)
 - Wartungsprotokolle (Maintenance Logs)
 - Implementierung der API-Endpunkte für:
 - Benutzerregistrierung und -anmeldung (FAFE-01, FAFE-04)
 - Integration von Authentifizierung und Autorisierung (z.B. JWT).
 2. **Frontend-Entwicklung:**
 - Einrichtung der React-Umgebung.
 - Implementierung der Register- und Login-Seite (FAFE-01, FAFE-04).
 - Anbindung des Frontends an das Backend (FAFE-02, FAFE-03).
 - Beginn der Hauptseite (Pilot Log) mit grundlegender Darstellung der Flugstunden (FAFE-05).
 3. **Testing:**
 - Erste Unit-Tests für Backend-Funktionen (z.B. Benutzerregistrierung).
 - Manuelle Tests der Register- und Login-Seite.

Phase 3: Frontend-Entwicklung mit MUSS-Anforderungen (bis zur 4. PVA am 10.05.2025)

- **Ziel:** Vollständige Implementierung des Frontends mit allen MUSS-Anforderungen.
- **Aufgaben:**
 1. **Frontend-Entwicklung:**
 - Implementierung der Hauptseite (Pilot Log) mit Filterfunktionen (FAFE-05, FAFE-06).
 - Erstellung der Maske für die manuelle Erfassung von Flugstunden (FAFE-07).
 - Implementierung der Fluggeräteverwaltung (FAFE-09, FAFE-10, FAFE-11).
 - Anzeige der Statistiken pro Fluggerät (FAFE-12).
 - Implementierung Die b.P Verwaltungsmaske (FAFE-08).
 2. **Backend-Anpassungen:**
 - Fluggeräteverwaltung (FAFE-09, FAFE-10, FAFE-11)
 - Flugstundenverwaltung (FAFE-05, FAFE-07)
 - Erweiterung der API-Endpunkte für Filterfunktionen und Statistiken.
 3. **Testing:**
 - Unit-Tests für Frontend-Komponenten.
 - Integrationstests für die Kommunikation zwischen Frontend und backend.
 - Manuelle Tests aller MUSS-Funktionen.

Phase 4: Implementierung der KANN-Anforderungen und Finalisierung (bis zur 5. PVA am 07.06.2025)

- **Ziel:** Implementierung der KANN-Anforderungen, Testing und Deployment.
- **Aufgaben:**
 1. **Implementierung der KANN-Anforderungen:**
 - CSV-Importfunktion für Flugstunden (FABE-0-02).
 - Bericht-Exportfunktion für Flugstunden (FABE-0-03).
 - Anhängen von Dateien an Fluggeräte (FABE-0-04).
 - Setzen von Wartung Alerts (FABE-0-05).
 - Anzeige von Start- und Landepunkten auf einer Karte (FAFE-0-01).
 2. **Testing:**
 - Manuelle Tests der KANN-Funktionen.
 - Bugfixing und Optimierung.
 3. **Deployment:**
 - Deployment des Backends und Frontends.
 4. **Dokumentation:**
 - Finalisierung der Projektdokumentation.
 - Erstellung einer Benutzeranleitung.

Zusammenfassung der Meilensteine

- **2. PVA (08.03.2025):** Dokumentation, Anforderungen, Design und Planung erstellt.
- **3. PVA (05.04.2025):** Backend mit grundlegenden Funktionen, Frontend mit Register und Login.
- **4. PVA (10.05.2025):** Frontend mit allen MUSS-Anforderungen fertig.
- **5. PVA (07.06.2025):** Projekt mit allen KANN-Anforderungen fertig, Testing und Deployment abgeschlossen.

Grobe Aufgabenteilung

- **Backend:** Django, Datenbank Konfiguration und Datenbankmodellierung, API-Entwicklung.
- **Frontend:** React, UI/UX-Design (Tailwind CSS), Anbindung an das Backend.
- **Testing:** Unit-Tests, Manuelle Tests.
- **Dokumentation:** Projektdokumentation, Benutzeranleitung.

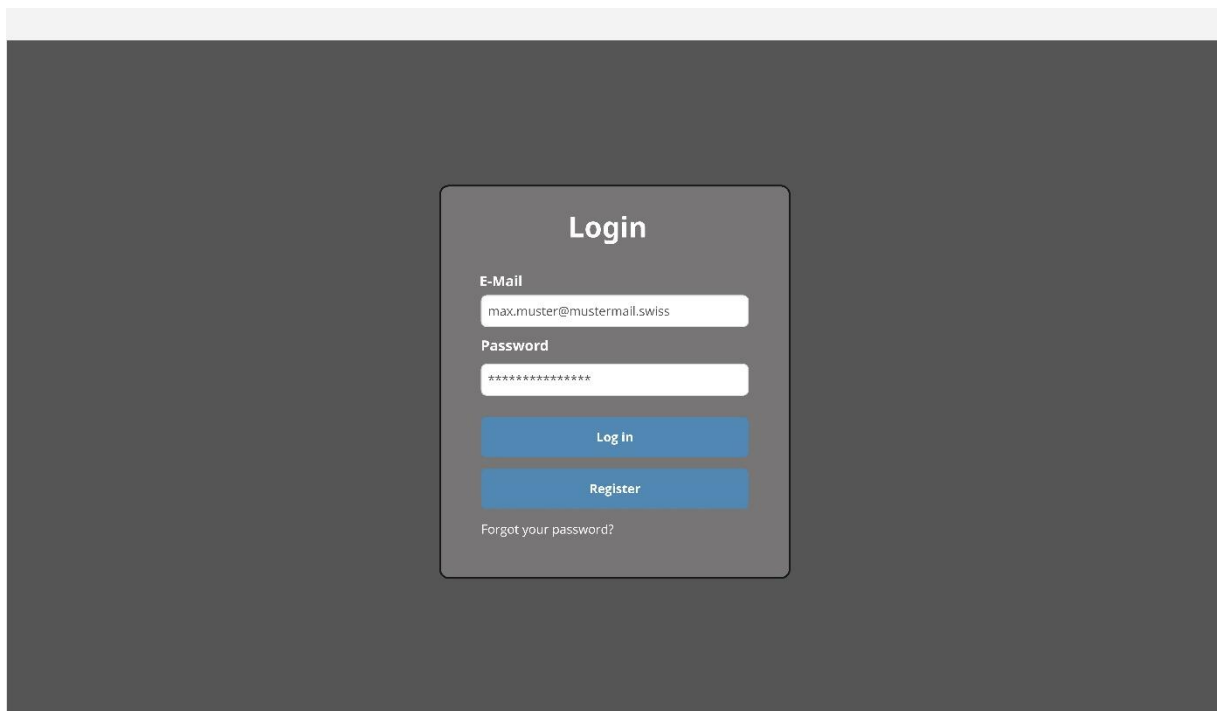
Mermaid Roadmap Visualizer

Es wurde bewusst auf ein Bild im Dokument verzichtet, da dieses aufgrund seiner Länge sehr unleserlich wäre. Über diesen Link gelangt man jedoch zu Mermaid, einer Applikation, die solche Visualisierungen anbietet.

<https://www.mermaidchart.com/app/projects/0a9438c2-25e7-48c5-8839-045d55d878e0/diagrams/033726ff-2d66-430e-9794-6fb56e36aeda/version/v0.1/edit>

Wireframes Desktop

Login:



A wireframe of a login form on a desktop screen. The form is centered on a dark gray background. It has a title 'Login' at the top. Below the title are two input fields: 'E-Mail' with the text 'max.muster@mustermail.swiss' and 'Password' with masked characters '*****'. Below these fields are two blue buttons: 'Log In' and 'Register'. At the bottom of the form is a link 'Forgot your password?'.

Login

E-Mail
max.muster@mustermail.swiss

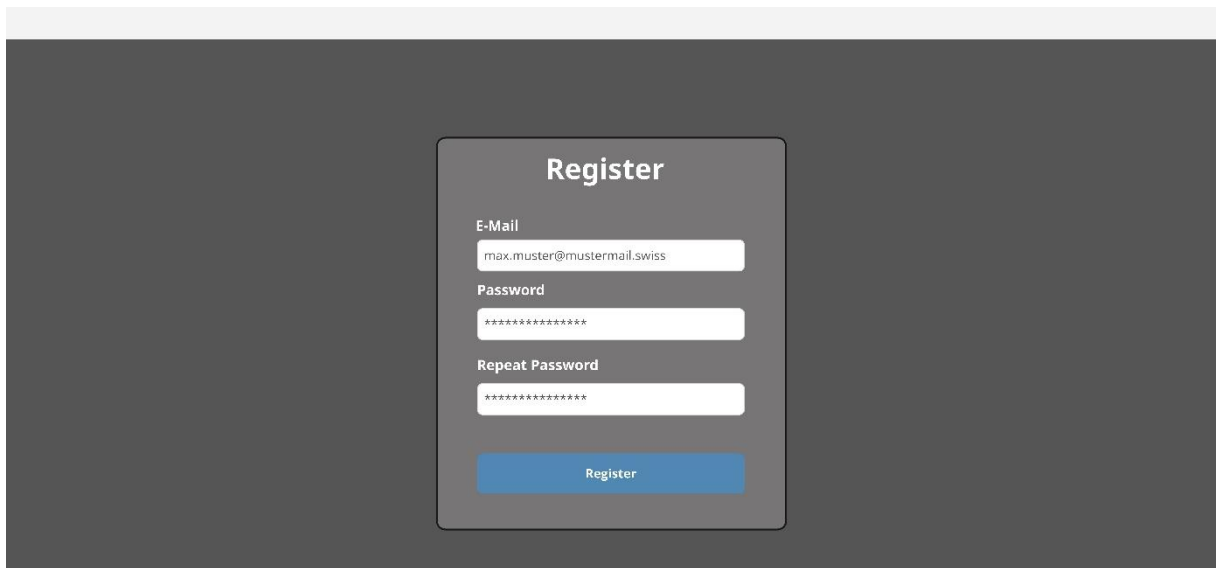
Password

Log In

Register

[Forgot your password?](#)

New User Registration:



A wireframe of a new user registration form on a desktop screen. The form is centered on a dark gray background. It has a title 'Register' at the top. Below the title are three input fields: 'E-Mail' with the text 'max.muster@mustermail.swiss', 'Password' with masked characters '*****', and 'Repeat Password' with masked characters '*****'. Below these fields is a blue button labeled 'Register'.

Register

E-Mail
max.muster@mustermail.swiss

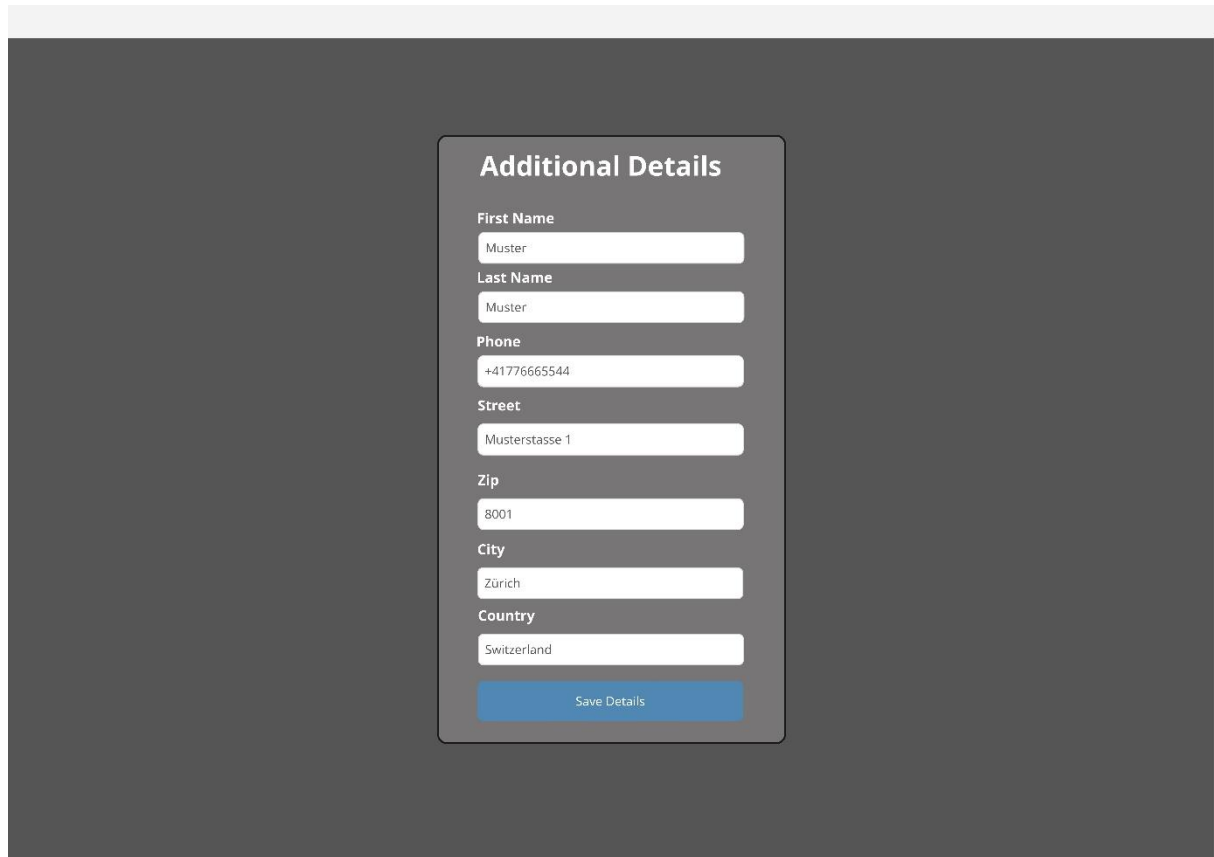
Password

Repeat Password

Register

Additional Details:

Um die Registrierungsmaske nicht zu überladen, wurde entschieden, dass die Registrierungsmaske aufgeteilt wird. Zuerst wird das Wesentliche abgefragt: E-Mail und Passwort, danach die persönlichen Daten.



The image shows a dark-themed web interface with a central form titled "Additional Details". The form contains the following fields and values:

- First Name:** Muster
- Last Name:** Muster
- Phone:** +41776665544
- Street:** Musterstasse 1
- Zip:** 8001
- City:** Zürich
- Country:** Switzerland


At the bottom of the form is a blue button labeled "Save Details".

Flight Logs (Main Page):

Mit dem Bleistift-Icon rechts sollen die Felder in Bezug auf den Flug angepasst werden können, z. B. Departure Place, Date, Departure Time, Landing Time, Landing Place, LDG, Aircraft, OPS Condition und Pilot Type. Die anderen Felder sind fix, da sie in Relation zum Aircraft automatisch ausgefüllt werden. Das Aircraft wird voraussichtlich über den Namen aufgelöst. Beim Klicken auf eine Ortschaft, z. B. Departure Place oder Landing Place, gelangt man zu den Flight Details (siehe nächstes Wireframe).

UAV Manager










Flight Log

 Jonatan Carvalhais

Flight Log

Aircraft List

User Setting

Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter		
Dept Place	Date	Dept Time	Ldg Time	Ldg Place	LDG	mm:ss	Aircraft	Registration	OPS Cond.	Pilot Type	
Hütten	01.05.2024	08:03	08:07	Hütten	1	2.4	DJI Air	CHEadsfkh..	VLOS D V	PIC	
Hütten	01.05.2024	08:53	08:58	Hütten	1	1.3	DJI Air	CHEadsfkh..	VLOS D V	PIC	
47.18, 8.66	12.06.2024	09:53	09:58	47.18, 8.66	1	1.5	Mobula 8	CHEadsfkh..	VLOS D V	Dual	
47.18, 8.66	12.06.2024	10:03	10:28	47.18, 8.66	1	2.3	Mobula 8	CHEadsfkh..	VLOS D V	PIC	
Zürich	01.07.2024	07:03	07:06	Zürich	1	2.1	Crux35	CHEadsfkh..	VLOS D V	Instrctur	
Horgen	08.09.2024	10:53	11:00	Horgen	1	2.3	Crux35	CHEadsfkh..	VLOS D V	PIC	
Hütten	01.02.2025	19:23	19:32	Hütten	1	1.1	SIM	N/A	BLOS N I	PIC	
Hütten	01.02.2025	09:23	09:30	Hütten	1	0.3	SIM	N/A	BLOS D I	PIC	
Hütten	01.02.2025	15:03	16:02	Hütten	1	3.3	Crux35	CHEadsfkh..	VLOS D V	PIC	

Total Flight Time: 23:50h Total Landing: 153

New	New	New	New	New	New	New	New	New	New
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Save new Flight


import CSV

Export CSV

Flight Details:

UAV Manager

Flight Details

 Jonatan Carvalhais

Flight Log

Aircraft List

User Settings

Drone Name: Mobula8-HDZero

Manufacturer: Happymodel

Type: Quad

Motors: 4

Type of Motor: Electric

Date: 21.10.2024

Dpt Time: 08:08

Registration Number: CHedkl92454dfg325

Serial Number: SN567896531248

OPS Conditions: VLOS D

Pilot Type: PIC

Landings: 4

Ldg Time: 08:12

Total Flight Time: 04min 23s

Takeoff

MAP View

Landing

Aircraft List:

Hier werden die verschiedenen UAVs (Aircraft) aufgelistet. Mit einem Klick in der Tabelle gelangt man zu den "Aircraft Settings", wo das Luftfahrzeug genauer betrachtet werden kann. Mit "New Aircraft" gelangt man zur Maske "New Aircraft", in der ein neues Luftfahrzeug erfasst werden kann.

UAV Manager

Aircraft List

Jonatan Carvalhais

Flight Log

Aircraft List

User Setting

Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter			
Aircraft	Manufactro	Type	Motors	Type of Motor	LDG	Firmware	Video System	GPS	MAG	BARO	GYRO	ACC
DJI Air	DJI	Quad	4	Electric	1	Betaflight	HD-ZERO	1	1	1	1	1
Mobula 8	Happymodel	Quad	4	Electric	1	Betaflight	HD-ZERO	N/A	N/A	1	1	1
Mobula 7	Happymodel	HEX	6	Electric	1	Betaflight	HD-ZERO	1	N/A	1	2	1
Crux35	Happymodel	Quad	4	Electric	1	Betaflight	HD-ZERO	N/A	1	N/A	1	1
CruxHD	Happymodel	Wing	1	Jet	1	INAV	HD-ZERO	2	1	2	2	1
Mobula 6	Happymodel	Tricopter	3	Electric	1	Betaflight	HD-ZERO	1	1	1	1	1

New Aircraft

Import CSV

New Aircraft Registration:

UAV Manager

New Aircraft

Jonatan Carvalhais

Flight Log

Aircraft List

User Settings

Drone Name

Mobula8 HDZero

Manufacturer

Happymodel

Type

Quad

Motors

4

Type of Motor

Electric

Video

Analog

Video System

HD-zero

ESC

Happymodel x12 AIO 12A

ESC Firmware

bludjay_0.21.0

Receiver

RadioMaster RP1

Receiver Firmware

dlrs v3.5.3

Flight Controller

Happymodel x12 AIO

Firmware

Betaflight

Firmware Version

4.5.5

GPS

1

MAG

1

Baro

1

Gyro

1

ACC

1

Registration Number

CHEDN92454ofg325

Serial Number

SN567896531248

Last Maintenance:

31.07.2021

Next:

31.07.2026

Reminder:

X

Props Maint:

21.12.2022

Next:

21.12.2029

Reminder:

X

Motor Maint:

21.10.2020

Next:

21.12.2029

Reminder:

X

Frame Maint:

21.10.2020

Next:

21.12.2029

Reminder:

X

INAV

DJI

KISS

Cleanflight

Baseflight

Emuflight

ArduPilot

Other

2

N

Save changes

Aircraft Settings:

Mit "Modify Aircraft" gelangt man zurück zur Maske "New Aircraft", die jedoch mit den Daten des aktuellen Luftfahrzeugs ausgefüllt ist. Dort können Anpassungen vorgenommen werden. Ein Löschen des Luftfahrzeugs ist nicht vorgesehen, da es in der Vergangenheit mit Flügen verknüpft wurde. Es kann jedoch im Log ein Vermerk mit "End of Life" hinzugefügt werden.

UAV Manager

Aircraft Settings

Jonatan Carvalhais

Flight Log

Aircraft List

User Settings

Drone Name: Mobula8-HDZero

Manufacturer: Happymodel

Type: Quad

Motors: 4

Type of Motor: Electric

Video: Analog

Video System: HD-Zero

ESC: Happymodel x12 AIO 12A

ESC Firmware: bluejay_0.21.0

Receiver: RadioMaster RP1

Receiver Firmware: elrs v3.5.3

Flight Controller: Happymodel x12 AIO

Firmware: Betaflight

Firmware Version: 4.5.5

GPS 1

Mag 1

Baro 1

Gyro 1

ACC 1

Registration Number: CHedkI92454dfg325

Serial Number: SN567896531248

Total Flight Time: 10:53 HH:MM

Total Landings: 252

Last Maintenance:

Reminder:

Props Maint: 31.07.2021 Next: 31.07.2028 X

Motor Maint: 21.12.2022 Next: 21.12.2029 X

Frame Maint: 21.10.2020 Next: 21.12.2029 X

Last Maintenance Log:

Date	Log	File
20.05.2024	check up successful	
12.05.2024	Crash in to trees	
Date	Type something	

Add Log

Export Log

Modify Aircraft

User Settings:

UAV Manager

User Settings

Jonatan Carvalhais

Flight Log

Aircraft

User Settings

First nameMax

Last nameMuster

Company NameCarvalhais Aviation GmbH

E-mailmax.muster@musermail.swiss

Drone Operator NumberCHedkI9204dfg345

Phone number+000 111 222 333

Pilot License numberFCLCH.345789

Street AddressSesame Street 000

License Category

Zip Code8000

CityZürich

A1 / A3Valid until: 31.07.2028

CountrySwitzerland

A2Valid until: 21.12.2029

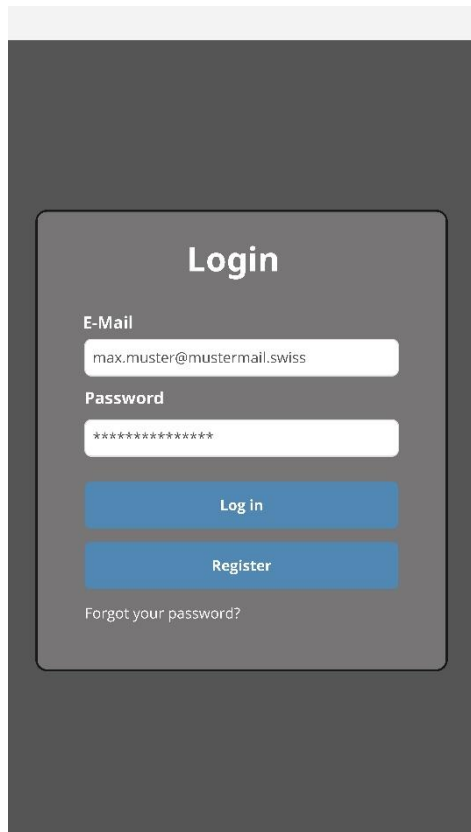
STSValid until: 21.12.2029

Save changes

Wireframes Mobile

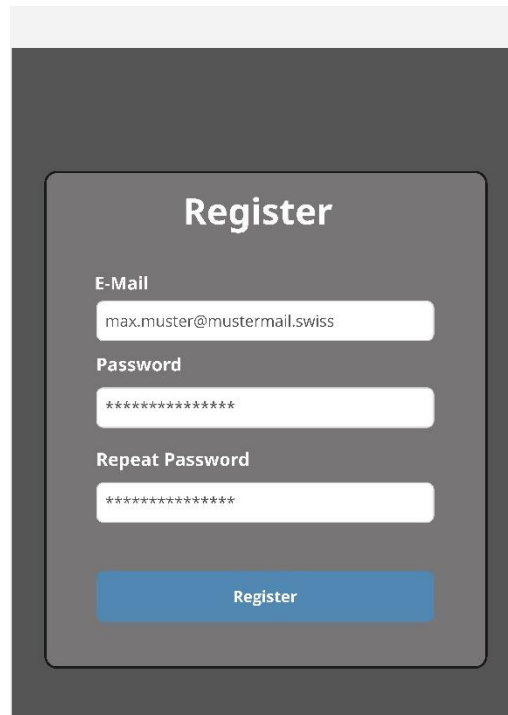
Die mobile Version verhält sich wie die Desktop-Version, jedoch wird die Darstellung so angepasst, dass sie sich leicht auf Smartphones anzeigen und bearbeiten lässt. Daher gehe ich hier nicht erneut auf die einzelnen Verbindungen ein.

Login:



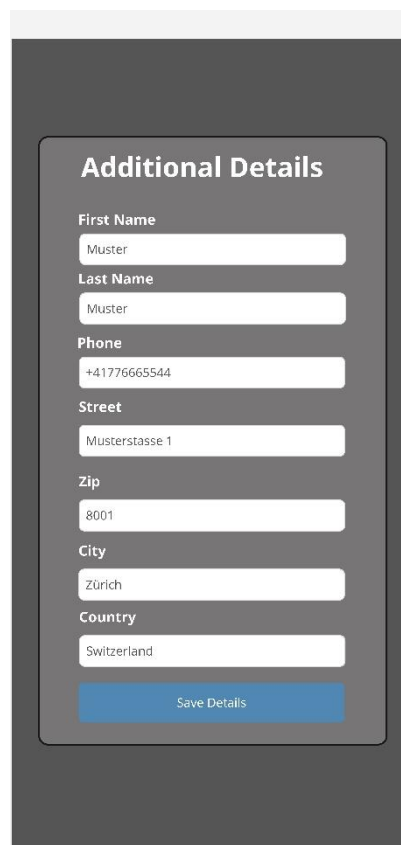
A mobile login wireframe on a dark gray background. It features a light gray header bar at the top. The main content is a rounded rectangle with a dark gray background. At the top of this rectangle is the title "Login" in white. Below the title are two input fields: "E-Mail" with the placeholder "max.muster@mustermail.swiss" and "Password" with masked characters "*****". Below these fields are two blue buttons: "Log in" and "Register". At the bottom of the rounded rectangle is a link that says "Forgot your password?".

Register:



A mobile register wireframe on a dark gray background. It features a light gray header bar at the top. The main content is a rounded rectangle with a dark gray background. At the top of this rectangle is the title "Register" in white. Below the title are three input fields: "E-Mail" with the placeholder "max.muster@mustermail.swiss", "Password" with masked characters "*****", and "Repeat Password" with masked characters "*****". Below these fields is a single blue button labeled "Register".

Additional Details:



A mobile additional details wireframe on a dark gray background. It features a light gray header bar at the top. The main content is a rounded rectangle with a dark gray background. At the top of this rectangle is the title "Additional Details" in white. Below the title are several input fields: "First Name" (placeholder: "Muster"), "Last Name" (placeholder: "Muster"), "Phone" (placeholder: "+41776665544"), "Street" (placeholder: "Musterstasse 1"), "Zip" (placeholder: "8001"), "City" (placeholder: "Zürich"), and "Country" (placeholder: "Switzerland"). Below these fields is a blue button labeled "Save Details".

Flight Log (Main Page)

UAV Manager

Flight Log

Flight Log

Aircraft

User Settings

Filter Dept. Place

Filter Date

Filter Dept. Time

Filter Ldg. Time

Filter Ldg. Place

Filter LDG

Filter mmiss

Filter Aircraft

Filter Registration

Filter OPS

Filter Day / Night

Filter IFR / VFR

Filter Pilot Type

Dept. Place: Hütten

Dept. Date: 01.05.2024

Dept. Time: 08:03

Ldg. Time: 08:07

Ldg. Place: Hütten

LDG: 1

mmiss: 2.4

Aircraft: DJI Air

Registration: CHEadklfsa

OPS: VLOS

Time Cond.: Day

Flight Roles: VFR

Pilot Type: PIC

Dept. Place: Hütten

Dept. Date: 01.05.2024

Dept. Time: 08:03

Ldg. Time: 08:07

Ldg. Place: Hütten

LDG: 1

mmiss: 2.4

Aircraft: DJI Air

Registration: CHEadklfsa

OPS: VLOS

Time Cond.: Day

Flight Roles: VFR

Pilot Type: PIC

Dept. Place: Hütten

Dept. Date: 01.05.2024

Dept. Time: 08:03

Ldg. Time: 08:07

Ldg. Place: Hütten

LDG: 1

mmiss: 2.4

Aircraft: DJI Air

Registration: CHEadklfsa

OPS: VLOS

Time Cond.: Day

Flight Roles: VFR

Pilot Type: PIC

Total Flight Time: 23:50h

Total Landing: 153

New Dept. Place

New Date

New Dept. Time

New Ldg. Time

New Ldg. Place

New LDG

New mmiss

New Aircraft

New Registration

New OPS

New Day / Night

New IFR / VFR

New Pilot Type

Save changes

Flight Details:

UAV Manager

Flight Details

Flight Log

Aircraft List

User Settings

Drone Name: Mobula8-HDZero

Manufactor: Happymodel

Type: Quad

Motors: 4

Type of Motor: Electric

Registration Number: CHEdki92454dfg325

Serial Number: SN567896531248

OPS Conditions: VLOS D V

Pilot Type: PIC

Landings: 4

Date: 21.10.2024

Dpt Time: 08:08

Ldg Time: 08:12

Total Flight Time: 04min 23s

Landing

MAP View

Takeoff

Aircraft List:

UAV Manager

Aircraft List

Flight Log

Aircraft List

User Settings

Filter Aircraft

Filter Manufacturer

Filter Type

Filter Motors

Filter Type of Motor

Filter LDG

Filter Firmware

Filter Video System

Filter GPS

Filter MAG

Filter BARO

Filter GYRO

Filter ACC

Aircraft: Mobula8
Manufacturer: Happymodel
Type: Quad
Motors: 4
Type of Motor: Electric
LDG: 1
Firmware: Betaflight
Video System: HD-ZERO
GPS: 1
MAG: N/A
BARO: 1
GYRO: 1
ACC: 1

Aircraft: Mobula8
Manufacturer: Happymodel
Type: Quad
Motors: 4
Type of Motor: Electric
LDG: 1
Firmware: Betaflight
Video System: HD-ZERO
GPS: 1
MAG: N/A
BARO: 1
GYRO: 1
ACC: 1

Aircraft: Mobula8
Manufacturer: Happymodel
Type: Quad
Motors: 4
Type of Motor: Electric
LDG: 1
Firmware: Betaflight
Video System: HD-ZERO
GPS: 1
MAG: N/A
BARO: 1
GYRO: 1
ACC: 1

Total Aircraft 5

New Aircraft

import CSV

New Aircraft Registration:

UAV Manager

Flight Log

Aircraft List

User Settings

New Aircraft

Drone Name

Mobula8-HDZero

Manufacturer

Happymodel

Type

Quad

Motors

4

Type of Motor

Electric

Video

Analog

Video System

HD-Zero

ESC

Happymodel x12 AIO 12A

ESC Firmware

bluejay_0.21.0

Receiver

RadioMaster RP1

Receiver Firmware

elrs v3.5.3

Flight Controller

Happymodel x12 AIO

Firmware

Betaflight

Firmware Version

4.5.5

GPS

1

Mag

1

Baro

1

Gyro

1

ACC

1

Registration Number

CHEdkI92454dfg325

Serial Number

SN567896531248

Last Maintenance:

Props Maint:

31.07.2021

Next:

31.07.2028

X

Motor Maint:

21.12.2022

Next:

21.12.2029

X

Frame Maint:

21.10.2020

Next:

21.12.2029

X

Save changes

tricopter

Hexacopter

Wing

Airplane

Piston

Jet

Glider

Digital

None

Walksnail

Caddx Vista

DJI O2

DJI O3

DJI O4

Analog

Others

INAV

Dij

KISS

Cleanflight

Baseflight

Emuflight

Ardupilot

Other

2

N

Aircraft Settings:

UAV Manager

Aircraft Settings

Flight Log

Aircraft List

User Settings

Drone Name: Mobula8-HDZero

Manufacturer: Happymodel

Type: Quad

Motors: 4

Type of Motor: Electric

Video: Analog

Video System: HD-Zero

ESC: Happymodel x12 AIO 12A

ESC Firmware: bluejay_0.21.0

Receiver: RadioMaster RP1

Receiver Firmware: elrs v3.5.3

Flight Controller: Happymodel x12 AIO

Firmware: Betaflight

Firmware Version: 4.5.5

GPS

Mag

Baro

Gyro

ACC

1

1

1

1

1

Registration Number: CHedkl92454dfg325

Serial Number: SN567896531248

Total Flight Time: 10:53 HH:MM

Total Landings: 252

Last Maintenance:

Reminder:

Props Maint: 31.07.2021

Next: 31.07.2028

X

Motor Maint: 21.12.2022

Next: 21.12.2029

X

Frame Maint: 21.10.2020

Next: 21.12.2029

X

Last Maintenance Log:

Date	Log	File
20.05.2024	check up successful	
12.05.2024	Crash in to trees	
Date	Type something	

Add Log

Export Log

Modify Aircraft

User Settings:

UAV Manager

User Settings

Flight Log

Aircraft

User Settings

First name

Max

Last name

Muster

E-mail

max.muster@mustermail.swiss

Phone number

+000 111 222 333

Street Address

Sesame Street 000

Zip Code

8000

City

Zürich

Country

Switzerland

Company Name

Carvalhais Aviation GmbH

Drone Operator Number

CHedkl9204dfg345

Pilot License number

FCL.CH.345789

License Category

A1 / A3 Valid until: 31.07.2028

A2 Valid until: 21.12.2029

STS Valid until: 21.12.2029

Save changes

Logbuch:

Datum	Task	time in minutes
11.02.2025	Beginn Projekt Dokumentation	120.00
12.02.2025	Projekt Doku	60.00
13.02.2025	Projekt Doku	180.00
14.02.2025	Wireframe	300.00
15.02.2025	Wireframe	60.00
16.02.2025	Wireframe	90.00
16.02.2025	HTML / CSS / Tailwind / Bootstraps research	90.00
16.02.2025	Projekt Doku	90.00
17.02.2025	Projekt Doku	60.00
17.02.2025	Wireframe	15.00
17.02.2025	Database Konzpetierung	15.00
18.02.2025	Databaseschema	60.00
22.02.2025	Projekt Doku	90.00
23.02.2025	Wireframe	60.00
23.02.2025	Projekt Doku	60.00
24.02.2025	Projekt Doku	60.00
18.03.2025	Anpassung DB Diagram	30.00

Hilfsmittelverzeichnis:

Hilfsmittel?	Wozu?	Betroffene Stellen
ChatGPT 4o	Mermaid Code (Grafiken)	Roadmap, DB & Architektur MAP
Word Copilot	Gramatik Korrektur	Gesamtes Dokument
ChatGPT 4o	Filestruktur nach vorgaben erstellt	Kapitel Filestruktur
Miro	Wireframe Design	Kapitel Wireframes