

Projektdokumentation

Entwicklung von EnvTrack

Webanwendung zur Auswertung von
spezialisierten Wetter- und Umgebungsdaten

Abgabetermin: 04.02.2025

Auftraggeber:

Herr Horst von Quatschkopf

Projektmitglieder:

Herr Erik Lauter

Herr Felix Janetzki

Einleitung	4
1.1. Zielsetzung des Projekts.....	4
1.2. Projektumfang.....	4
1.3. Motivation und Hintergrund	4
1.4. Struktur der Dokumentation.....	5
1.5. Ausgangssituation	5
1.5.1 Anforderungen	5
1.5.2 Wichtige Funktionalitäten.....	5
Projektplanung.....	6
2.1. Projektziele	6
2.2. Meilensteine und Zeitrahmen	6
2.3. Ressourcenplanung	7
Projektmethodik	7
3.1. Vorgehensweise und Arbeitsmethoden.....	7
3.2. Technologische Grundlagen	7
3.3. Werkzeuge und Software	7
3.4. Teamstruktur und Zusammenarbeit	7
Durchführung des Projekts	8
4.1. Analysephase.....	8
4.2. Designphase	8
4.3. Implementierungsphase	10
4.4. Testphase.....	10
4.5. Abnahme und Implementierung.....	11
5.1. Erzielte Ergebnisse.....	11
5.2. Analyse und Interpretation der Ergebnisse.....	12
5.3. Vergleich mit den ursprünglichen Zielen.....	12
5.4. Problemlösungen und Herausforderungen.....	12
Fazit	13
6.1. Zusammenfassung der Projektergebnisse	13
6.2. Kritische Reflexion	13
6.3. Ausblick und mögliche Weiterentwicklungen	13
Anhang	14

Abbildungsverzeichnis.....	14
----------------------------	----

Einleitung

1.1. Zielsetzung des Projekts

Ziel dieses Projekts ist die Entwicklung eines Tools zur Auswertung und Visualisierung von Wetter- und Umgebungsdaten anhand einer intuitiv gestalteten Webanwendung.

Das System soll im Web erreichbar sein, was die Benutzung von praktisch allen Standorten ermöglicht.

Des Weiteren soll das Projekt dem Auftraggeber die Messdatenauswertung, welche vorher anhand von manuellen Messen und in Excel auswerten umgesetzt wurde, erleichtern.

1.2. Projektumfang

Das System bietet Funktionen für die automatisierte Erfassung, Aggregation und Visualisierung von lokalen Wetter- und Umgebungsdaten.

Des Weiteren beinhaltet das System die Fähigkeit, basierend auf durchschnittlichen Messwerten gewisse Pflanzenkulturen zu empfehlen, welche zu den gemessenen Bedingungen angebaut werden können.

Weiterhin ist das System in der Lage, begrenzte Vorhersagen über die kommenden Temperaturen zu treffen.

Das Serverbackend und das Frontend des Systems wird anhand der Technologien TypeScript/JavaScript unter der Laufzeitumgebung Node.JS, der Webserver-Bibliothek Express.JS und der Datenbank MariaDB umgesetzt.

Die Datenerfassung, Aufbereitung und Übermittlung der Wetter- und Umgebungsdaten wird anhand der Technologien C, C++, den IoT-Programmiersplattformen ESP32, STM32 und dem Datenübertragungsprotokoll „LoRa“ umgesetzt.

1.3. Motivation und Hintergrund

Herr Horst von Quatschkopf (von nun an „Der Klient“) ist ein Großbauer spezialisiert auf das Anbauen von diversen Pflanzen.

Der Klient muss regelmäßig auf seine 13 Felder herausfahren und an jedem Feld die Umgebungsdaten manuell anhand von diversen Sensoren und Messgeräten aufzeichnen.

Die gesammelten Messdaten werden daraufhin manuell in sein Tablet-PC eingetippt, welches er in sein Eigenheim bringt, um dort die gemessenen Daten mühsam in MS-Excel auszuwerten.

Der Gesamtaufwand beläuft sich hier auf rund einen vollen Arbeitstag (05:30 – 20:30), was für den Klienten im fortschreitenden Alter nicht mehr tragbar ist.

Aufgrunde dessen hat der Klient das Team von unabhängigen Softwareentwicklern, Herrn Erik Lauter und Herrn Felix Janetzki damit beauftragt, eine Lösung zu entwickeln, welche Zeit einspart und den Klienten schont.

1.4. Struktur der Dokumentation

Die Dokumentation ist in mehrere Abschnitte unterteilt. Nach der Einleitung werden die Ziele, Meilensteine, Ressourcen und Risiken des Projekts detailliert dargestellt. Anschließend wird die Projektmethodik beschrieben, die den Arbeitsansatz, die technologischen Grundlagen und die verwendeten Werkzeuge beschreibt. Abschließend werden die Ergebnisse bewertet und ein Fazit gezogen, um mögliche Weiterentwicklungen aufzuzeigen.

1.5. Ausgangssituation

Der Klient ist ein Großbauer spezialisiert auf das Anbauen von diversen Pflanzen und befindet sich im fortschreitenden Alter.

Das Sammeln der Messwerte stellt für ihn eine langwierige, anstrengende und reilegmäßige Aufgabe dar, welcher er auf Grund von diversen Beschwerden nicht mehr lange nachkommen kann.

Aufgrunde dessen hat er Herrn Erik Lauter und Herrn Felix Janetztki damit beauftragt, eine Softwarelösung zu entwickeln, welche ihm diese Arbeit abnimmt.

1.5.1 Anforderungen

Der Klient benötigt eine Software welche die Arbeit der manuellen Wetter- und Umgebungsdatensammlung, der Auswertung und Visualisierung jener übernimmt.

Hier ist es wichtig, dass die Anwendung auf Mobilgeräten und Desktop-Computern funktioniert.

Ebenfalls darf die Benutzeroberfläche der Anwendung nicht zu komplex sein und die Bedienung sollte benutzerfreundlich umgesetzt sein.

Des Weiteren müssen die Daten in einer Datenbank gespeichert werden um eine effiziente Speichernutzung der Hardware zu gewährleisten.

1.5.2 Wichtige Funktionalitäten

- Automatisches Erfassen von Wetter- und Umgebungsdaten
- Geringer Energiebedarf der Messstation
- Autarkiefähigkeit der Messstation
- Fehlerlose Übertragung der Messdaten zum Backend
- Visualisierung der gemessenen Daten in einstellbaren Zeitrahmen
- Empfehlen von Pflanzkulturen, welche in den gemessenen Bedingungen wachsen
- Aggregation der gemessenen Daten (Minimum, Maximum, Durchschnitt)
- Vorhersagen von kommenden Temperaturen

Projektplanung

2.1. Projektziele

Unser Ziel ist es, ein benutzerfreundliches webbasierendes System zur Wetter- und Umgebungsdatenauswertung und Visualisierung zu entwickeln.

Die Entwicklung erfolgt unter den Programmiersprache C, C++, TypeScript/JavaScript unter den IoT-Programmierplattform ESP32 und STM32.

Die Programm-Module sind so erweiterbar wie möglich umzusetzen, um eine einfache und kostengünstige Weiterentwicklung zu ermöglichen.

Die Benutzung einer modularen Router- und Service-Architektur ermöglicht das schnelle hinzufügen von neuen Endpunkten und Schnittstellen.

Die Programm-Module der Hardware (Treiber, und diverse Aufbereitungsfunktionen) sind simpel zu halten um so die Wartbarkeit zu gewähren.

2.2. Meilensteine und Zeitrahmen

- Analysephase (13. November – 3. Dezember): Definition der Anforderungen und Erstellung eines grundlegenden Konzepts
- Designphase (4. Dezember – 7. Dezember): Planung der Systemarchitektur im Backend und in der Hardware der Messstation und Erstellung der Schnittstellendefinitionen
- Implementierungsphase (8. Dezember – 15. Januar): Entwicklung der Kernfunktionen und des Übertragens der Daten der Messstation zum Backend
- Testphase (16. Januar – 25. Januar): Durchführung von Whitebox-Tests und Benutzertests anschließend weiterer Reviews und Bugfixes.
- Abschlussphase (25. Januar – 2. Februar): Dokumentation und Bereitstellung des Systems einschließlich einer Kurzschulung.

2.3. Ressourcenplanung

- Hardware: 1 ESP32-Mikrocontroller, 1 STM32-Mikrocontroller, 1 Sensor-Array, div. Kabel, Lötutensilien, 3D-Drucker, 2 Windows 10-PCs
- Software: JetBrains WebStorm IDE, STM32 IDE, Arduino IDE, Git für die Versionskontrolle.
- Personal: 2 Entwickler.
- Zeit: Vom 13. November bis 3. Februar.

Projektmethodik

3.1. Vorgehensweise und Arbeitsmethoden

Das Projekt wird iterativ umgesetzt. Jede Iteration umfasst die Schritte Analyse, Design, Implementierung und Test. Dies ermöglicht eine schrittweise Verbesserung und Anpassung der Software an die Anforderungen.

3.2. Technologische Grundlagen

- Node.JS, TypeScript: Die Basis-Technologien für das Backend und das Frontend
- C, C++, Assembly: Die Basis-Technologien für die Programmierung der Messstation
- MariaDB: Die Datenbank zur Speicherung der Messdaten

3.3. Werkzeuge und Software

- JetBrains WebStorm: Entwicklungsumgebung für die Programmierung des Backends und des Frontends.
- STM32 IDE, Arduino IDE: Entwicklungsumgebungen für die Programmierung der Messstations-Hardware
- Git: Versionskontrollsystem zur Nachverfolgung von Änderungen.
- Web: Zielplattform für die Software.

3.4. Teamstruktur und Zusammenarbeit

Herr Lauter und Herr Janetzki konzeptionieren zusammen die Systemarchitektur und die Schnittstellen zwischen der Messstation und dem Backend.

Herr Lauter übernimmt die Programmierung der Hardware der Messstation und der Datenübertragung in das Backend.

Herr Janetzki übernimmt die Implementierung des Backend, Frontend und der Datenbankstruktur.

Durchführung des Projekts

4.1. Analysephase

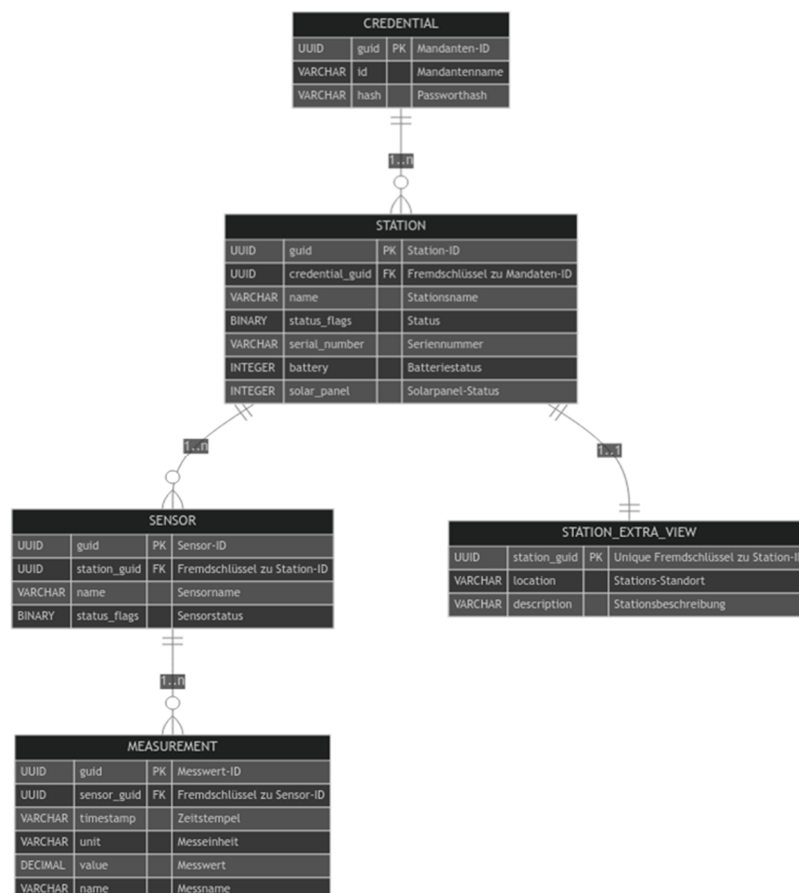
In der Analysephase wurden die Anforderungen und die Grundbedürfnisse des Kleinten ermittelt.

Die Erstellung eines detaillierten Lastenhefts war ein wichtiges Ergebnis dieser Phase, das die gewünschten Funktionen spezifizierte. Außerdem wurden auch nichtfunktionale Anforderungen wie Benutzerfreundlichkeit, erreichbarkeit und Plattformübergreifbarkeit ermittelt.

Darüber hinaus wurde eine erste Entwurfsskizze für die kommende Design- und Implementierungsphase erstellt. Die wichtigsten Use Cases wurden identifiziert.

4.2. Designphase

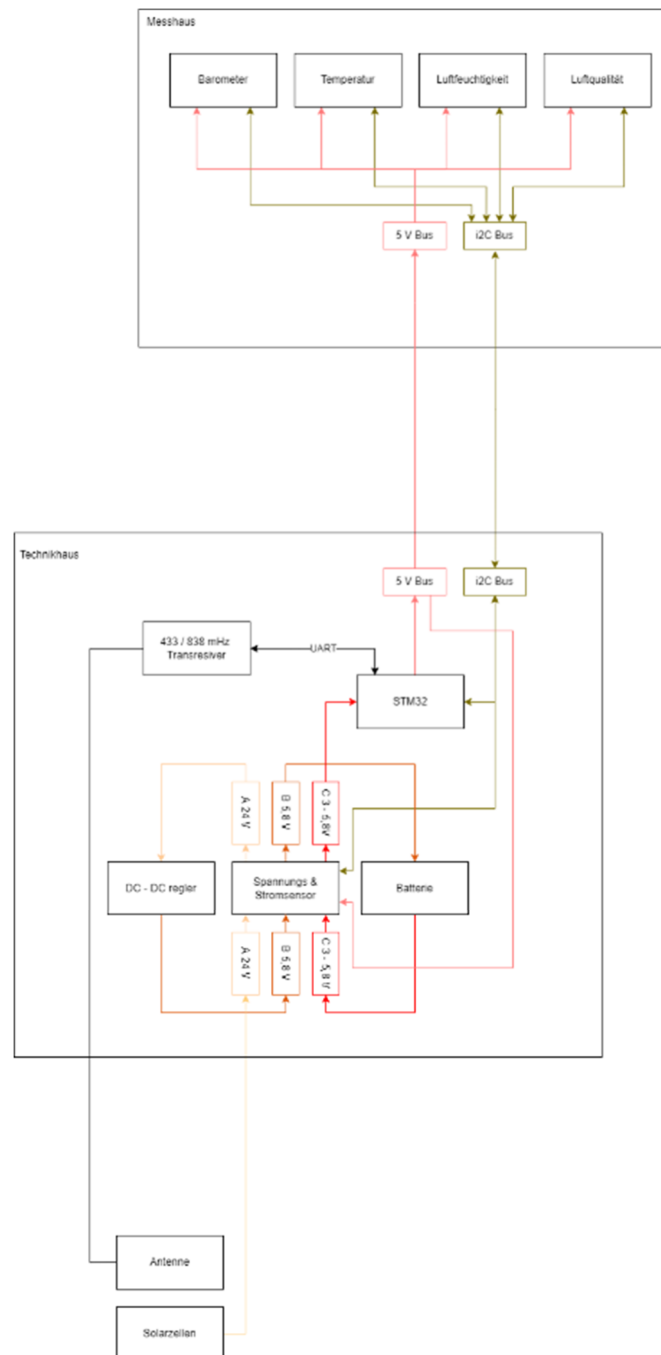
Die Beziehung zwischen den einzelnen Datenbanktabellen (Entitäten) werden hier in einem relationalen Diagramm dargestellt.



Entwicklung von EnvTrack

Webanwendung zur Auswertung von spezialisierten Wetter- und Umgebungsdaten

In dem folgenden Blockdiagramm ist der Hardware- und Kommunikationsaufbau der Wetterstation dargestellt.



4.3. Implementierungsphase

In der Implementierungsphase wurde das Backend und Frontend des Projekts in den Programmiersprachen TypeScript/JavaScript unter Node.JS entwickelt.

Der Fokus lag hier auf einer klaren Abtrennung zwischen Businesslogik und Visualisierung.

Die Saubere Trennung der Businesslogik (Services) und der Präsentation (Routes) kann in Abbildung 1 erkannt werden.

Die grundlegenden Funktionen wie z.B. die automatisierte Erfassung, Visualisierung und Auswertung der gemessenen Daten wurden implementiert.

Die Nutzung von generischen Schnittstellen sorgte währenddessen für die kontinuierliche Erweiterbarkeit des Systems und regelmäßige Peer-Reviews sorgten für eine saubere und wartbare Codebasis.

Die Wetterstation wurde hardwareseitig umgesetzt und anschließend softwareseitig programmiert, um ihre Funktionalität sicherzustellen.

Der Hauptfokus lag dabei auf einer robusten und energieeffizienten Umsetzung, um einen Ausfall der Wetterstation zu vermeiden. Die Grundfunktion besteht darin, in regelmäßigen Intervallen Messungen durchzuführen und die erfassten Daten anschließend zu übertragen.

Der Code wurde modular aufgebaut, was eine einfache Erweiterung ermöglicht. Zudem konnte jedes Modul einzeln getestet werden, wodurch die Stabilität und Zuverlässigkeit der einzelnen Komponenten sichergestellt wurde.

Der modulare aufbau kann in Abbildung 2 eingesehen werden.

4.4. Testphase

Parallel zu den regelmäßigen Peer-Reviews wurden frequent Benutzertests durchgeführt bei welchen besonderes Augenmerk auf der Benutzerfreundlichkeit lag.

Ebenfalls wurden manuelle Tests durchgeführt, hauptsächlich an der Visualisierung und der Funktion der Hardware.

4.5. Abnahme und Implementierung

Nach dem Beenden der Tests wurde das System dem Auftraggeber vorgestellt und anhand der Anforderungen überprüft.

Die Abnahme verlief erfolgreich, da das System den Anforderungen des Kunden entsprach.

Anschließend wurde das System auf einem Cloud-VPS (Server) aufgesetzt, eine Subdomäne wurde erstellt, unter welcher das System erreichbar ist und die Projektdateien wurden dem Auftraggeber übergeben.

Die Projektmitarbeiter führten daraufhin eine Kurzschulung durch, um den Umgang mit dem System verständlich zu machen und offene Fragen zu klären.

Ergebnisse und Evaluation

5.1. Erzielte Ergebnisse

Das Projekt wurde erfolgreich und für den Kunden zufriedenstellend umgesetzt; Es umfasst praktisch alle Anforderungen und ist benutzerfreundlich.

Dabei bietet es folgende Features:

- Interaktive Menüs: Simple Navigation durch die Seite sorgen dafür, dass Benutzer mit wenig Erfahrung am Computer nicht überfordert werden.
- Autarkie der Messstation: Die Messstation beinhalten einen Leistungsstarken Akku, welcher durch zwei Solarzellen aufgeladen wird.
- Geringer Energiebedarf der Messstation: Durch Low-Energy Komponenten wird viel Energie der Batterie der Messstation gespart.
- Modularer und generischer Aufbau des Programm führt zu einfach Erweiterbarkeit in der Zukunft.
- Die Software ist auf praktisch allen Geräten nutzbar, sofern die Geräte über einen Webbrowser verfügen

Die Implementierung umfasst Funktionen wie:

- Automatische Erfassung und Übertragung von lokalen Wetter- und Umgebungsdaten zum Backend alle 90 Sekunden.
- Simple Navigationsstruktur zur Visualisierung und Auswertung jener Daten.
- Effizientes Speichern der Daten in einer optimierten Datenbank

5.2. Analyse und Interpretation der Ergebnisse

Die erzielten Ergebnisse entsprechen insgesamt den Anforderungen des Klienten.

Die geplante generische und modulare Struktur und Verwendung von einfachen Schnittstellen sorgen dafür, dass sich das System in der Zukunft leicht anpassen lässt.

Die Messstation funktioniert praktisch Autark und ist in ihrer Verdrahtung modular aufgebaut, wodurch auch neue Sensorik und andere Batterien problemlos verbaut werden können.

Die in der Visualisierungsseite angezeigten Graphen werden in Echtzeit aktualisiert, so dass zu jeder Zeit die aktuellsten Werte dargestellt werden.

Die Verwendung von Hochperformanten Programmiersprachen und bewährter Hardware versichert eine schnelle und reibungslose Erfassung und Übermittlung der Daten.

Die Vorhersagefunktion stellte das größte Problem dar, welches bis zum Projektabschluss nicht vollständig gelöst werden konnte.

5.3. Vergleich mit den ursprünglichen Zielen

Das Projekt erfüllt die definierten Ziele.

- Die Software bietet Möglichkeiten zur Auswertung und Visualisierung der Messdaten anhand einer Webanwendung
- Die Software ist im Web erreichbar und somit auf praktisch allen erdenklichen Geräten, welche über einen Webbrowser verfügen nutzbar.
- Nach der Testphase wurde bestätigt, dass die Software dem Auftraggeber eine deutliche Arbeitserleichterung bietet.
- Das neuronale Netzwerk auf dem die Vorhersagefunktion basiert passt sich zu schnell und zu inkonsistent den Messdaten an; Hier ist geschäht Overfitting.
 - Das Problem könnte mit Nacharbeiten gelöst werden, z.B. in dem das Trainingsintervall verlängert wird.

5.4. Problemlösungen und Herausforderungen

Wie bereits erwähnt stellte die Vorhersage die größten Probleme dar, nicht zuletzt zugrunde der doch eher geringen Erfahrung des Projektpersonals im Bezug zu Vorhersagemodellen und Machine Learning.

Weiterhin problematisch war die korrekte Verteilung der Energie an die einzelnen elektronischen Komponenten in der Messstation; Dies erforderte einige Nacharbeiten.

Aufgrund der konsequenten Nutzung von Abstraktion und einfachen Schnittstellen wurde der Implementierungsaufwand erhöht, was zu einer Abweichung der Dauer von c.a 1 Tag der Implementierungsphase führte.

Fazit

6.1. Zusammenfassung der Projektergebnisse

Das Projekt hat eine funktionierende, benutzerfreundliche Webanwendung und Messstation zur automatisierten Datenerfassung, Darstellung und Auswertung hervorgebracht.

Es stellt eine solide Basis dar, die zukünftig durch modulare Erweiterung problemlos Angepasst werden.

6.2. Kritische Reflexion

Positiv anzusehen ist die klar aufgebaute Struktur der Anwendung die die hochqualitativen, genauen Messdaten.

Des Weiteren ist die Codebasis des Projekts durch regelmäßiges Prüfen sauber gehalten worden und strikt nach dem DRY-Prinzip geschrieben worden.

Die Entscheidung, eine MariaDB-Datenbank zu benutzen basiert auf der Erfahrung des Projektpersonals mit eben jener; Retrospekt hätte man besser eine PostgreSQL verwenden sollen, da diese generell performanter ist und nativ mit JSON arbeiten kann.

Exportfunktionen der Daten in gewöhnlichen Formaten, wie z.B CSV oder JSON sind nicht gegeben, dies hätte man zur Interoperabilität mit Drittsystemen umsetzen können.

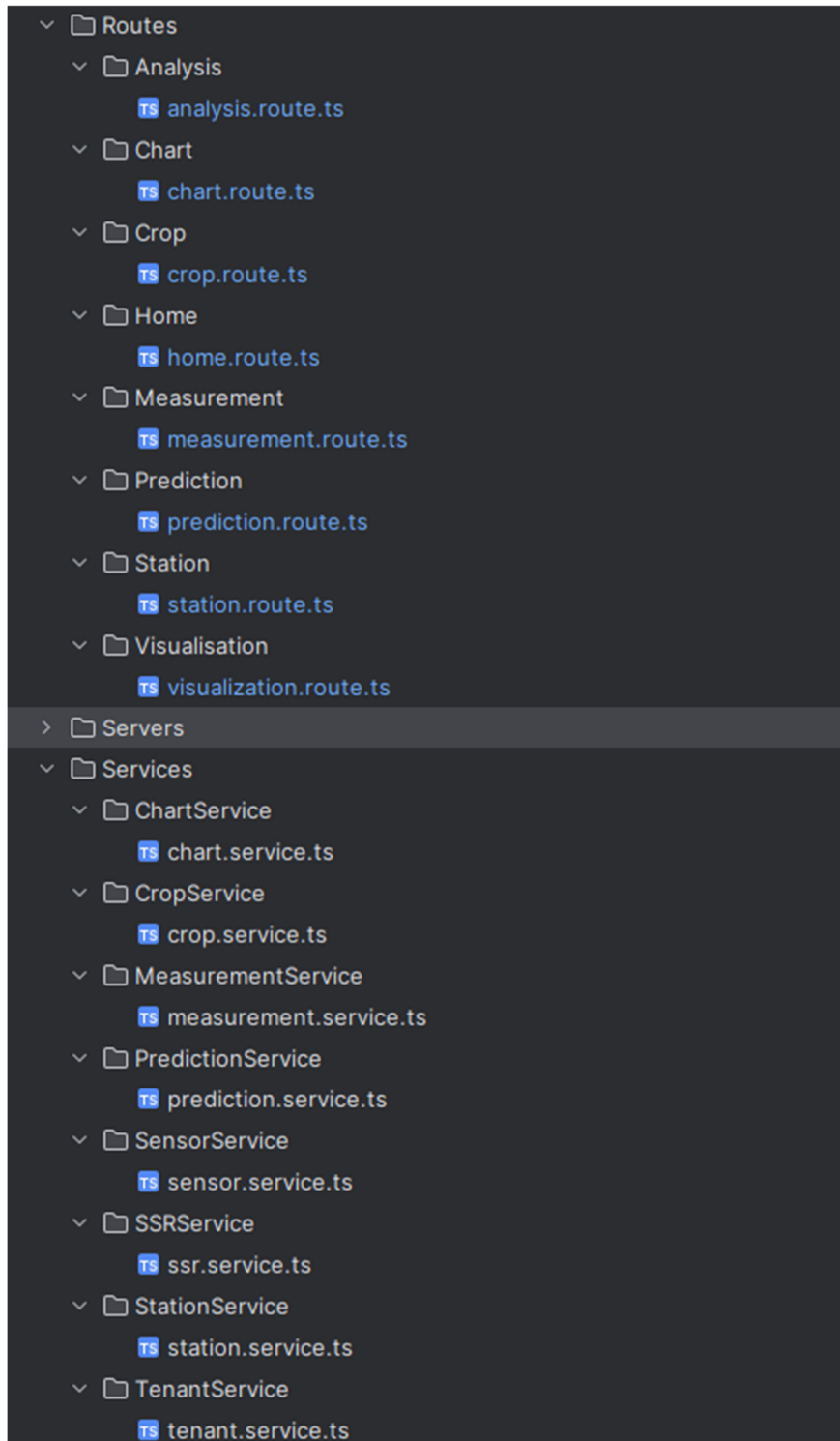
6.3. Ausblick und mögliche Weiterentwicklungen

- Exportfähigkeit:
 - Exportoptionen zum Generieren von CSV, JSON oder PDF-Berichten über die Messdaten.
- Neuronales Netzwerk verbessern:
 - Größere Anzahl an Ausgabeneuronen hinzufügen um die Temperaturen für mehrere Tage vorherzusagen.
- Auf PostgreSQL wechseln:
 - Höhere Abfrageperformance und native Unterstützung von JSON.

Anhang

Abbildungsverzeichnis

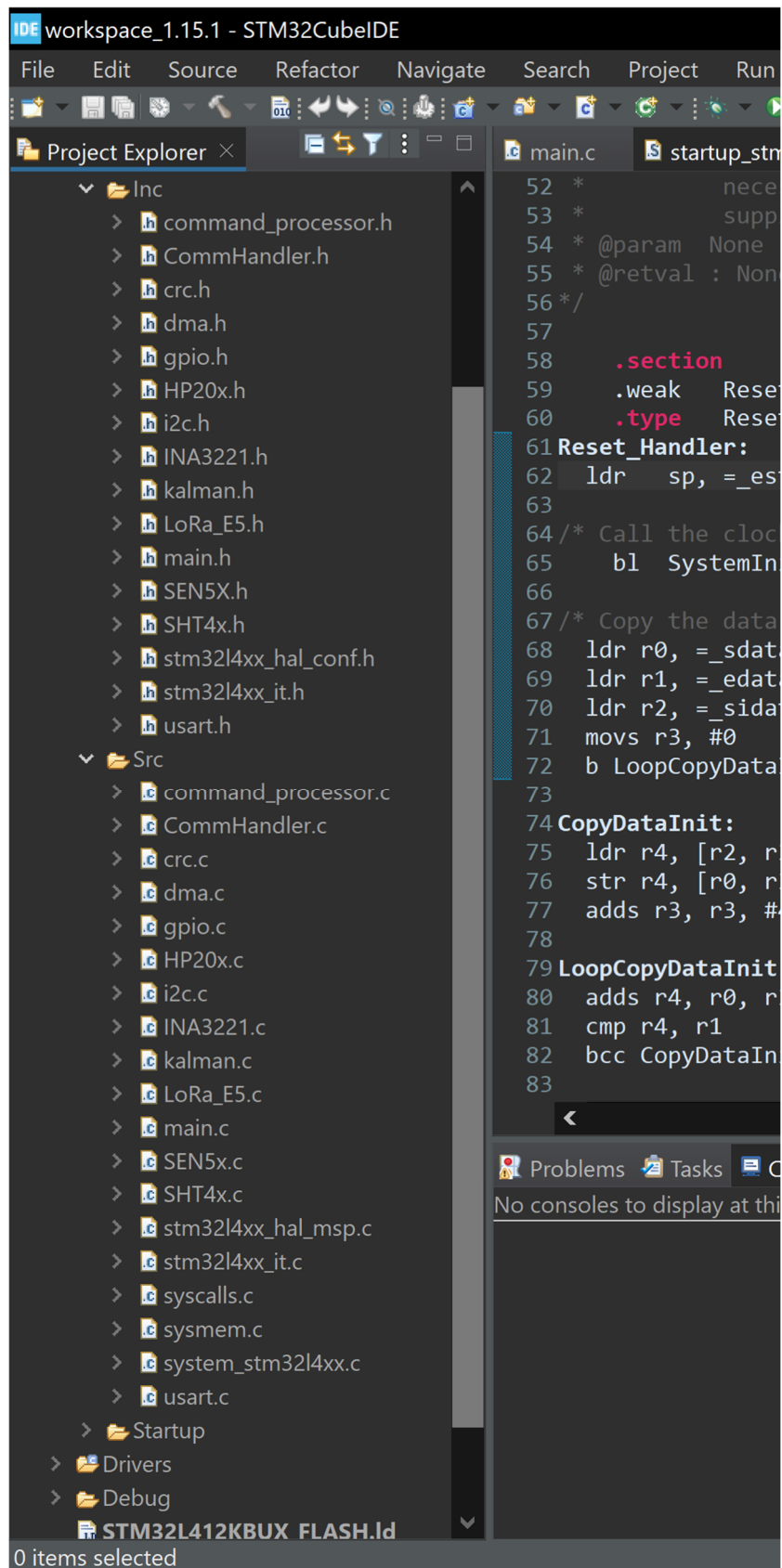
Abbildung 1:



Entwicklung von EnvTrack

Webanwendung zur Auswertung von spezialisierten Wetter- und Umgebungsdaten

Abbildung 2:



7.1. Glossar

- **JavaScript**
 - Eine Programmiersprache, welche ursprünglich für Programmierung im Web eingesetzt wurde, allerdings seit 2009 auch Serverseitig anwendbar ist
- **TypeScript**
 - Eine Programmiersprache, welche als Endprodukt JavaScript erzeugt. Sie stellt Sicherheitsfeatures zur Kompilierzeit bereit, welche Standard-JavaScript nicht enthält
- **JSON**
 - Ein Format zur Datenübertragung, welches eng mit JavaScript verwandt ist
- **Abstraktion**
 - Das Verwenden von Schnittstellen, um die Funktionen von Klassen zu trennen und flexibler zu gestalten
- **DRY-Prinzip**
 - Ein Programmierstil, der Wiederholungen im Code vermeidet
- **JetBrains WebStorm**
 - Die Entwicklungsumgebung, mit der der Backend- und Frontend Projektcode geschrieben wurde
- **STM32-IDE**
 - Die Entwicklungsumgebung, in welcher der Code der Messstation (Treiber, etc.) geschrieben wurden
- **Arduino-IDE**
 - Die Entwicklungsumgebung, in welcher der Code zur Messdatenübermittlung zum Backend-Server geschrieben wurde
- **Modularität**
 - Das System wurde in kleinere, unabhängige Teile aufgeteilt, die separat funktionieren
- **Portierbarkeit**
 - Das Programm funktioniert auf diversen endgeräten einwandfrei