

# Projektdokumentation

## Fishdex

**Im Modul Softwaretechnik**

Peter Strößner

Hong Lam Nguyen

Finja Adam

Nico Liebig

Finn Geneviève Hieber

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis.....</b>	<b>2</b>
1. Projektbeschreibung.....	4
1.1. Aufgabenstellung.....	4
1.2. Projektbeschreibung.....	5
1.3. Inspiration.....	5
2. Pflichtenheft.....	7
2.1. Zielbestimmung.....	7
2.1.1. Musskriterien.....	7
2.1.2. Wunschkriterien.....	9
2.1.3. Abgrenzungskriterien.....	9
2.2. Produkteinsatz.....	9
2.2.1. Anwendungsbereiche.....	9
2.2.2. Zielgruppen.....	10
2.3. Produktübersicht.....	10
2.4. Funktionale Anforderungen.....	10
2.5. Nicht-Funktionale Anforderungen.....	11
2.6. Produktdaten.....	11
2.7. Laufzeitumgebung - Hardware und Software.....	12
2.8. Entwicklungsumgebung.....	12
2.8.1. Software.....	12
2.8.2. Hardware.....	13
2.8.3. Orgware: Coding-Richtlinien.....	13
3. Diagramme.....	15
3.1. Klassendiagramme.....	15
3.2. Sequenzdiagramm.....	16
4. Entwicklerdokumentation.....	17
4.1. Einrichtung der Entwicklungsumgebung.....	17
4.2. Einrichtung der Laufzeitumgebung.....	18
4.3. Aufbau und Struktur des Projekts.....	18
4.4. Versionsverwaltung.....	19
4.4.1. Feature-Branch-Workflow.....	19
4.4.2. Tickets und Meilensteine.....	19
5. Vorgehensmodell.....	20
6. Projektdokumentation.....	21
6.1. Aufgabenverteilung.....	21
6.2. Herangehensweise an das Projekt.....	21
6.3. Entscheidungsfindungsprozess.....	22
6.4. Roadmap.....	23

7. Backend und Frontend.....	25
8. Projektergebnis.....	26
9. Anhang.....	27
10. Unterzeichnung.....	28

# 1. Projektbeschreibung

## 1.1. Aufgabenstellung

Im Rahmen des Moduls Softwaretechnik haben wir die Möglichkeit, in Gruppen von 3-5 Personen ein eigenständiges Projekt zu erarbeiten. Die wichtigsten Punkte der Aufgabenstellung sind wie folgt definiert:

### Termine:

- Abgabe Gruppenmitglieder und Projektidee: 18. März 2024 (per E-Mail)
- Projektabgabe: 05. Mai 2024 (per E-Mail)
- Zwischenberichte werden individuell mit den Gruppen vereinbart.

### Anforderungen:

- Das Projekt soll ein objektorientiertes Design (OOD) mit Vererbung, Aggregation und Delegation implementieren.
- eine Präsentation von ca. 20 min

### Die Projektpräsentation umfasst folgende Punkte:

- Aufgabenverteilung: In der Präsentation werden die Rollen und die Aufgabenverteilung innerhalb des Teams vorgestellt, um einen Einblick in den Beitrag jedes Teammitglieds zu geben.
- Projektidee: Die Präsentation enthält eine detaillierte Darstellung der Projektidee, einschließlich ihrer Entstehung, Ziele und angestrebten Funktionalitäten.
- Design: Die Präsentation enthält Informationen zum Design der Anwendung, einschließlich der Benutzeroberfläche und des visuellen Designs.
- Frameworks, Architektur: Es wird erläutert, welche Frameworks und Architekturprinzipien bei der Entwicklung der Anwendung verwendet wurden und warum diese ausgewählt wurden.
- Besondere Merkmale: Besondere Funktionen oder Merkmale der Anwendung werden hervorgehoben und erläutert, um ihre Bedeutung und ihren Mehrwert zu verdeutlichen.
- Probleme während der Entwicklung: Die Präsentation behandelt alle Herausforderungen oder Probleme, die während der Entwicklung aufgetreten sind und wie diese überwunden wurden.

- Live-Demo: Abschließend wird eine Live-Demo der Anwendung durchgeführt, um ihre Funktionalität und Benutzerfreundlichkeit zu veranschaulichen.

## 1.2. Projektbeschreibung

Das Projekt umfasst eine Webanwendung, die mit ReactJS im Frontend und Spring Boot (Java) im Backend umgesetzt ist. Die Kommunikation zwischen Frontend und Backend erfolgt über eine von Spring Boot bereitgestellte API.

Die Grundidee des Teams ist ein Angler-Logbuch, mit dem Angler ihre Fang-Aktivitäten protokollieren und verwalten können. Die Anwendung soll es Benutzern ermöglichen, Einträge über gefangene Fische zu erstellen, die dann in einem "Sammelbuch", dem Fishdex, angezeigt werden. Innerhalb des Logbuchs können weitere Details zu jedem gefangenen Fisch eingesehen werden, wie z.B. die Anzahl der Fangversuche oder Fotos des Fisches. Die Einträge werden in einer Timeline ähnlich einem Social Media Post dargestellt.

Die Anwendung kann um folgende Funktionen erweitert werden:

- Ein System für Erfolge und Badges, basierend auf den Aktivitäten der Nutzer, wie z.B. die Anzahl der gefangenen Fische.
- Die Implementierung von Benutzerprofilen für personalisierte Erfahrungen.
- Die Integration eines Mehrbenutzersystems, das es den Benutzern ermöglicht, die Profile und Einträge anderer Nutzer einzusehen, wodurch die Anwendung soziale Netzwerkfunktionen erhält.

## 1.3. Inspiration

Das Projekt Fishdex entstand aus einer Reihe von Ideen und Diskussionen innerhalb des Teams. Zuerst dachten wir an die Entwicklung einer Ausleih-App, mit der Benutzer den Ausleihstatus ihrer Bücher verfolgen können. Wir stellten jedoch schnell fest, dass diese Idee nicht ganz unseren Vorstellungen entsprach.

Nach weiteren Überlegungen und dem Wunsch, eine Anwendung zu entwickeln, die sowohl nützlich als auch unterhaltsam ist, entschieden wir uns für die Weiterentwicklung unseres Angler-Logbuchs. Diese Entscheidung ermöglichte es uns, eine Plattform zu schaffen, die nicht nur das Protokollieren von Angel-Aktivitäten ermöglicht, sondern auch die soziale Interaktion zwischen Anglern fördert.

Um unsere Idee weiter zu konkretisieren und einen einprägsamen Namen für unsere Webanwendung zu finden, haben wir uns von der Grundidee des Pokédex inspirieren lassen. Daher haben wir die Anwendung "Fishdex" genannt. So wie der Pokédex eine Sammlung von Pokémon darstellt, soll der Fishdex einen Überblick über alle bereits gefangenen Fische geben.

Diese Entscheidungen spiegeln unser Bestreben wider, ein Projekt zu entwickeln, das nicht nur unseren persönlichen Interessen entspricht, sondern auch einen Mehrwert für die Benutzer schafft und unsere Fähigkeiten in der Softwareentwicklung weiterentwickelt.

## 2. Pflichtenheft

### 2.1. Zielbestimmung

Fishdex stellt eine Internetanwendung dar, die das virtuelle Sammeln von Fischen und Fängen dieser für eine Person, den Benutzer, ermöglicht.

#### 2.1.1. Musskriterien

##### Der Benutzer

- hat einen Benutzernamen, ein Profilbild und eine Benutzerbeschreibung
- hat ein Profil
- kann im Profil seinen Benutzernamen, sein Profilbild und seine Benutzerbeschreibung ändern
- hat eine Timeline
- hat einen Fishdex
- es gibt nur einen Benutzer

##### Das Profil

- enthält für den Benutzer Möglichkeiten, den Benutzernamen, das Profilbild und die Benutzerbeschreibung zu ändern (wie bereits unter *Benutzer* beschrieben)
- zeigt Benutzernamen, Profilbild und Benutzerbeschreibung des Benutzers an

##### Der Benutzername

- ist Text
- darf maximal 15 Zeichen lang sein
- erlaubte Zeichen für den Benutzernamen: a-z, A-Z, 0-9, . , \_
- der Benutzer muss einen Benutzernamen angeben; die Angabe ist verpflichtend

##### Das Profilbild

- wenn nicht gesetzt, soll das Profilbild ein Dummy-Bild zeigen, das vom Entwickler-Team festgelegt wird
- es handelt sich folglich um eine optionale Angabe des Benutzers
- Der Benutzer kann sein Profilbild als PNG oder JPEG hochladen

##### Die Benutzerbeschreibung

- ist Text
- darf maximal 350 Zeichen lang sein
- wird vom Benutzer selbst festgelegt
- falls der Benutzer keine Beschreibung anlegt, gibt es keine Beschreibung
- es handelt sich folglich um eine optionale Angabe des Benutzers

## Der Fishdex

- stellt eine Übersicht der vom Benutzer bereits gepflegten Fischarten dar
- kann also durch den Benutzer um Fischarten erweitert werden
- zeigt an, welche Fischart wie oft gefangen worden ist bzw. ob eine Fischart bereits gefangen wurde

## Eine Fischart

- kann von dem Benutzer durch einem Formular angelegt werden
- hat folgende Eigenschaften, die vom Benutzer direkt angegeben werden (können):
  - o Name (verpflichtend)
  - o Gewässer, in dem die Art normalerweise zu finden ist (verpflichtend)
  - o Köder, mit dem die Art normalerweise zu fangen ist (optional, mehrere möglich)
  - o essbar oder nicht (optional, entweder-oder-Angabe)
  - o weitere Informationen (optional)
  - o Bild (optional; falls nicht angegeben, wird ein Dummy-Bild benutzt)
- hat folgende Eigenschaften, die der Benutzer nur indirekt beeinflusst:
  - o Counter/Anzahl der bereits gefangen Fische dieser Art
  - o wird erhöht, wenn ein neuer Fang dieser Fischart vom Benutzer angelegt wird

## Die Timeline

- stellt eine Übersicht über die vom Benutzer bereits gepflegten Fänge von Fischen an
- erweitert sich, wenn der Benutzer einen Fang anlegt
- stellt die letzten Fänge sortiert nach der Zeit dar (letzte Fänge oben, erster Fang unten)

## Ein Fang

- kann vom Benutzer durch ein Formular angelegt werden
- hat folgende Eigenschaften, die vom Benutzer direkt angegeben werden (können):
  - o Datum (verpflichtend)
  - o Ort des Fangs (verpflichtend)
  - o Länge des Fangs (verpflichtend)
  - o Fischart des Fangs (verpflichtend)
  - o Beschreibung (z.B. vom Tag des Fangs, optional)
  - o Bild (optional; falls nicht angegeben, wird ein Dummy-Bild benutzt)



Es wird empfohlen, dass Bilder als PNG- oder JPEG-Datei vom Benutzer bereitgestellt werden.

### 2.1.2. Wunschkriterien

#### Die Timeline

- Kann nach Eigenschaften der Fänge gefiltert werden
- Kann nach Eigenschaften der Fänge sortiert werden (A-Z, Datum des Fangs, ...)

Bilder werden beim Hochladen automatisch in Web-Datei umgewandelt.

Mehrbenutzersystem (Social Media Plattform)

Responsives Design

### 2.1.3. Abgrenzungskriterien

Eingeschränkter Funktionsumfang:

Es ist nicht vorgesehen, umfassende Community-Funktionen wie Kommentare, Likes oder direkte Interaktionen zwischen Nutzern anzubieten.

Keine Integration externer Dienste:

Fishdex ist eine eigenständige Anwendung und wird keine direkte Integration mit externen Diensten wie Social Media, Wettervorhersagen oder Anbietern von Angelzubehör bieten

Keine Garantie für die Verfügbarkeit der Plattform:

Da Fishdex als akademisches Projekt entwickelt wird, kann keine dauerhafte Verfügbarkeit oder Wartung der Plattform garantiert werden.

## 2.2. Produkteinsatz

### 2.2.1. Anwendungsbereiche

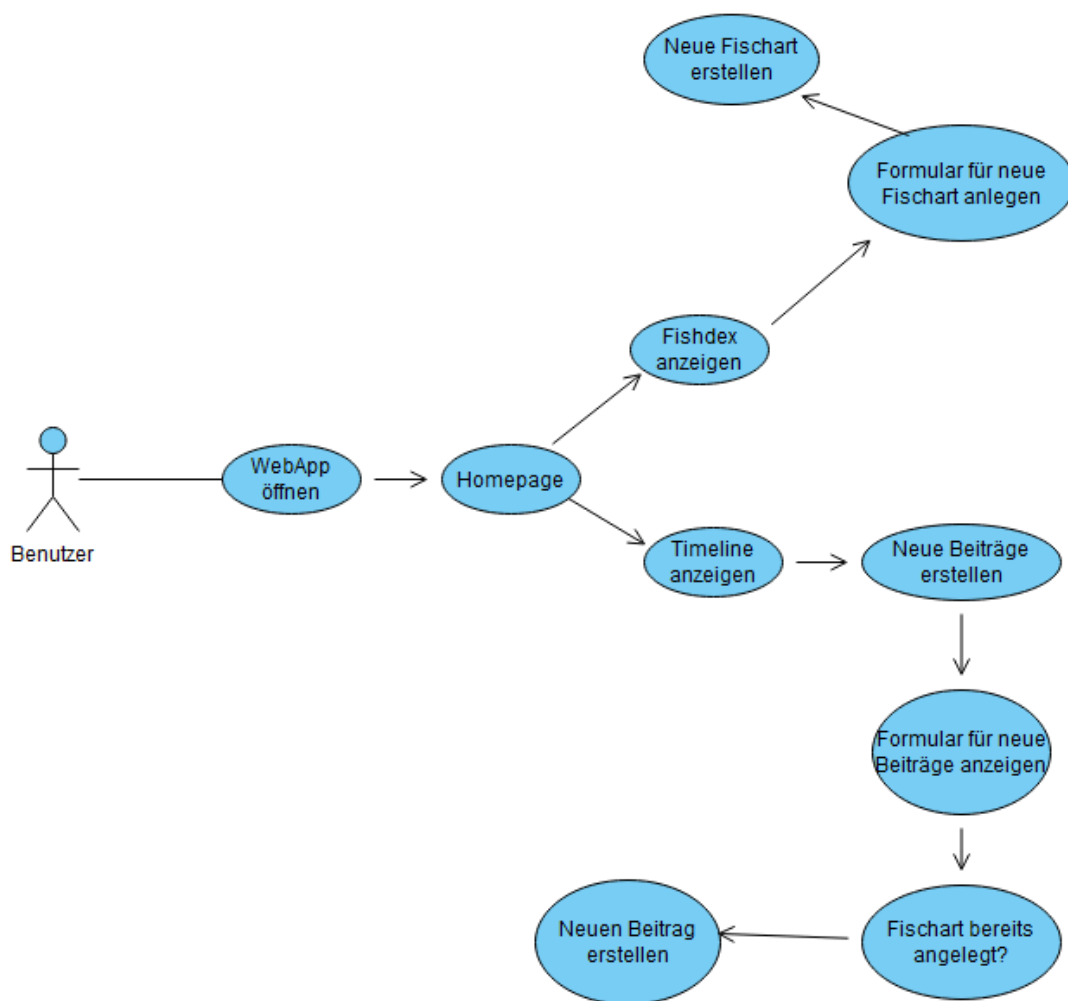
Die Anwendung Fishdex bietet die Möglichkeit, Fischfänge nach Fischarten zu sortieren und zu dokumentieren. Der Fishdex kann als digitales Sammelalbum bezeichnet werden. Angler und Anglerinnen können die Anwendung dazu benutzen, ihre in der Realität gefangenen Fische digital durch Metadaten zu archivieren.

### 2.2.2. Zielgruppen

Die Zielgruppe sind Angler und Anglerinnen, die durch digitale Werkzeuge ihre Angel-Erfahrungen archivieren möchten.

### 2.3. Produktübersicht

#### Use Case Diagramm



### 2.4. Funktionale Anforderungen

F001- Fishdex-Ansicht:

Benutzer können eine Übersicht aller von ihnen gefangenen Fische in ihrem Fishdex anzeigen lassen.

F002- Fischart hinzufügen:

Benutzer können neue Fischarten zu ihrem Fishdex hinzufügen, indem sie die relevanten Informationen eingeben.

F003 - Fang hinzufügen:

Benutzer können neue Fänge zu ihrem Fishdex hinzufügen, indem sie das Datum, den Ort, die Länge und die Fischart des Fangs angeben.

F004- Timeline-Ansicht:

Benutzer können eine chronologische Ansicht aller ihrer Fänge in ihrer Timeline anzeigen lassen.

## 2.5. Nicht-Funktionale Anforderungen

NFR001 - Performance:

Fishdex soll eine akzeptable Ladezeit haben, um eine reibungslose Nutzung zu gewährleisten. Seiten sollten innerhalb von 3 Sekunden oder weniger geladen werden.

NFR002 - Benutzerfreundlichkeit:

Die Benutzeroberfläche von Fishdex soll intuitiv und benutzerfreundlich sein, um eine einfache Navigation und Interaktion für Benutzer aller Erfahrungsstufen zu ermöglichen.

NFR003 - Sicherheit:

Die Sicherheit der Benutzerdaten hat höchste Priorität. Fishdex muss angemessene Sicherheitsmaßnahmen implementieren, um den Schutz der Benutzerdaten vor unberechtigt Zugriff oder Datenverlust zu gewährleisten.

NFR004 - Barrierefreiheit:

Die Benutzeroberfläche von Fishdex sollte barrierefrei gestaltet sein und den Zugang für Benutzer mit Behinderungen oder Einschränkungen erleichtern, indem sie den WCAG-Richtlinien (Web Content Accessibility Guidelines) entspricht.

NFR005 - Kompatibilität:

Fishdex sollte auf einer Vielzahl von Geräten und Webbrowsern konsistent und fehlerfrei funktionieren, einschließlich Desktop-Computern, Laptops, Tablets und Smartphones sowie verschiedenen Browsern wie Chrome, Firefox, Safari und Edge.

## 2.6. Produktdaten

D001 - Fischesammlung:

## Sammlung von Fischarten

### D002 - Fischdaten:

Daten über eine Fischart, Counter (wie oft, dieser Fisch bereits gefangen wurde)

### D003 - Fangsammlung:

Sammlung von allen Fängen

### D004 - Fangdaten:

Daten über einen einzelnen Fischfang (z.B. Datum)

## 2.7. Laufzeitumgebung - Hardware und Software

Ein Rechner mit einem darauf installierten Browser wird für den Betrieb der Software benötigt. Die Versionen der Browser sind in untenstehender Tabelle zu finden.

Außerdem wird eine Tastatur benötigt.

Chrome	122.0
Safari	17.3
Edge	122
Firefox	124.0

Auf allen Geräten, auf denen Browser verfügbar sind, kann Fishdex über eben diesen aufgerufen werden.

## 2.8. Entwicklungsumgebung

### 2.8.1. Software

Die Technologien, mit denen die Webanwendung entwickelt wird, sind die folgenden.

#### Frontend - ReactJS

Ein Teammitglied hat vor Projektstart bereits Erfahrung mit dieser Bibliothek gesammelt. Dabei hat sich ReactJS als effizientes Werkzeug zur Erstellung von interaktiven Benutzeroberflächen erwiesen.

#### Backend - Spring Boot (Java)

Java ist durch ein vorhergegangenes Modul allen Gruppenmitgliedern bekannt. Spring Boot bietet sich für die Bearbeitung des Projekts an, da Spring Boot eine API bereitstellt, die die Kommunikation zwischen Frontend und Backend erleichtert. Java wird im Backend in Version 21 mit der SDK corretto-21 verwendet. Um das Backend der Anwendung zu testen, wird Gradlew benötigt.

### [API-Schnittstellen testen - Insomnia](#)

### [Versionsverwaltung - Git \(Hosting Github\)](#)

Git ist allen Teammitgliedern geläufig und Änderungen können übersichtlich verfolgt werden. Die Anwendung kann mithilfe von Github kollaborativ entwickelt werden. Außerdem bietet Github weitere Möglichkeiten, das Projekt zu organisieren.

### [IDE - IntelliJ Community Edition bzw. IntelliJ Ultimate bzw. VisualStudio Code](#)

#### 2.8.2. Hardware

Ein Rechner wird benötigt, auf dem die oben beschriebene Software laufen kann.

#### 2.8.3. Orgware: Coding-Richtlinien

##### [Coding-Richtlinien auf Github](#)

##### [Namensgebung](#)

Verwende aussagekräftige Namen für Variablen, Funktionen und Klassen.  
camelCase größte Zeit in Frontend, PascalCase in Backend  
Vermeide Abkürzungen soweit es geht.

##### [Kommentare](#)

Schreibe aussagekräftige Kommentare in Deutsch (Code in Englisch!)  
Kommentiere komplexe Teile des Codes für besseres Verständnis.

##### [Bezüglich Git und GitHub](#)

Wir arbeiten hauptsächlich im dev-Branch mit Feature-Branch-Workflow.

##### [Allgemeiner Ablauf beim Coden](#)

Vor einem neuen Feature-Branch immer den dev-Branch auf den neuesten Stand bringen

```
git checkout dev
```

```
git pull
```

Erstelle für jedes Feature neuen Git-Branch (zB. feature/irgendein-feature)

```
git checkout -b feature/neues-feature
```

Mache deine Änderungen und lade sie in deinen lokalen Branch hoch.

```
git add .
```

```
git commit -m "Eine Commitnachricht hinzufügen"
```

Lade jetzt deine Commits in die Repository hoch.

```
git push
```

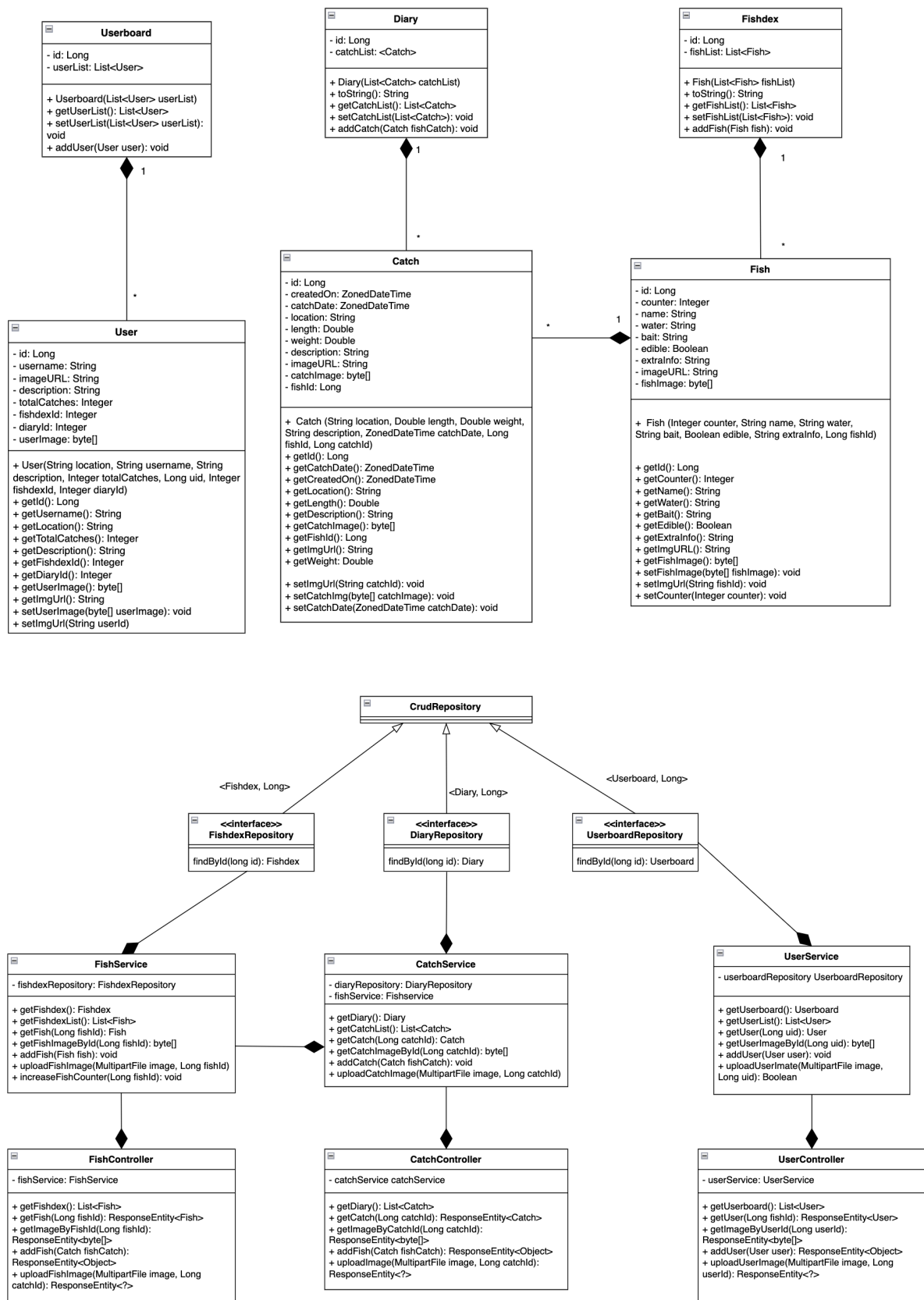
Mach auf GitHub einen Pull Request, um in den dev-Branch zu mergen.

### Code-Reviews

Es braucht mind. 1 Person, die sich deinen Pull Request anschaut und bestätigen muss, bevor es in den dev-Branch gemerged wird.

## 3. Diagramme

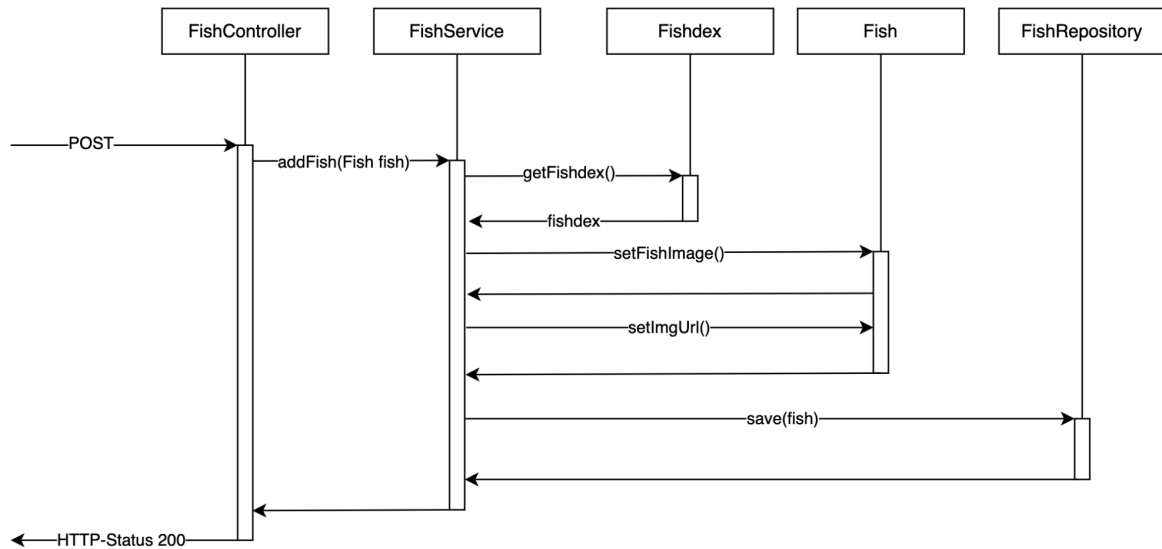
### 3.1. Klassendiagramme



Anmerkung: Die User-Klassen sind wie im Klassendiagramm implementiert. Allerdings werden ihre Funktionen bisher nicht genutzt.

### 3.2. Sequenzdiagramm

Ein neuer Fisch wird angelegt





## 4. Entwicklerdokumentation

Die Entwicklerdokumentation dient dazu, jeden, der sich für das Testen oder das Weiterentwickeln der App interessiert, die nötigen Informationen bereitzustellen.

### 4.1. Einrichtung der Entwicklungsumgebung

Um die Entwicklungsumgebung für das Projekt einzurichten, folgen Sie bitte den nachstehenden Anweisungen:

#### Allgemeine Einrichtung:

(Falls nötig) Git-Repository herunterladen mit:

```
git clone https://github.com/HonKLam/fishdex
```

Für die Entwicklungsumgebung wird folgendes empfohlen:

- VSCode (Frontend)
- IntelliJ IDEA (Backend)

#### Zur vollständigen Einrichtung der Umgebung

Innerhalb von “fd-frontend”:

- Node.js / npm benötigt
- yarn installieren:  

```
npm install --global yarn
```
- Alle benötigten Abhängigkeiten herunterladen mit  

```
yarn install
```
- Entwicklungsumgebung für das Frontend starten mit  

```
yarn run dev
```
- Das Frontend sollte unter [localhost:5173](http://localhost:5173) erreichbar sein.

Innerhalb von “fd-backend”:

- Java Development Kit (JDK) installieren (es wird Version 21 benötigt)

- innerhalb IntelliJ IDEA das Projekt “fd-backend” aus dem Repository importieren
- es sollten automatisch alle Abhängigkeiten durch die IDE heruntergeladen werden
- ggf. muss die richtige JDK-Version (21) ausgewählt werden
- Entwicklungsumgebung für das Backend starten mit  

```
./gradlew bootRun
```
- Das Backend sollte unter [localhost:8080](#) erreichbar sein.

## 4.2. Einrichtung der Laufzeitumgebung

Um das Testen der App zur Bewertung des Projekts zu erleichtern, wird das Einrichten der Laufzeitumgebung über Docker empfohlen.

- Laden Sie sich hierfür “Docker Desktop” für Ihr Betriebssystem herunter und folgen Sie den Installationshinweisen des Programms
- Navigieren Sie im Terminal zum entsprechenden Fishdex-Repository und führen Sie folgenden Befehl aus:

MacOS/Linux:

```
docker compose up -build
```

Windows:

```
docker-compose up --build
```

- In Ihrem Terminal werden nun sowohl Frontend als auch Backend als Image gebaut und anschließend in einem Docker-Container ausgeführt - dies kann eine bestimmte Zeit dauern
- Sie können nun in Ihrem Browser unter der Adresse “localhost” (localhost:80) die vollständige App mit vordefinierten Daten ausprobieren

## 4.3. Aufbau und Struktur des Projekts

Der Fishdex besteht aus zwei Teilen, die jeweils unterschiedliche Technologien verwenden.

## Frontend:

JavaScript mit ReactJS als Library

- Dient als Schnittstelle für User und App
- Bekommt seine Daten durch das Backend und kann Anfragen zur Änderung von Daten an das Backend senden

## Backend:

Java mit Spring Boot als Framework

- Stellt die API-Schnittstellen bereit
- verarbeitet Anfragen aus dem Frontend und sendet Antworten zurück
- Verwaltet und speichert in die eigene H2-Datenbank mittels der Java-Persistence-API (JPA)

Für weitere Informationen über den Stack selbst, kann im Pflichtenheft unter "Entwicklungsdokumentation" nachgeschaut werden.

Zusammenfassend arbeiten jedoch beide Teile nur in Kombination mit einer API (Application Programming Interface) zusammen.

Die einzelnen Entities/Models, Controller (API-Schnittstellen), Services, Repositories und die dazugehörigen Eigenschaften und Methoden wurden mit Hilfe von JavaDoc unter "fd-backend/docs" dokumentiert.

## 4.4. Versionsverwaltung

Für die Versionsverwaltung und Zusammenarbeit wird GitHub verwendet. Das Repository ist unter [diesem Link](#) zu finden. Innerhalb des Repositories stehen weitere Hinweise zur Arbeit innerhalb des Projekts.

### 4.4.1. Feature-Branch-Workflow

In Git wird der "Feature-Branch-Workflow" verwendet, um neue Funktionen zur App hinzuzufügen. Für mehr Informationen, kann [dieser Artikel](#) weiterhelfen. Im Pflichtenheft stehen weitere Informationen, wie man Git dafür richtig verwendet.

### 4.4.2. Tickets und Meilensteine

Es werden Tickets in Form von "GitHub Issues" geschrieben, die eine einzelne Aufgabe beschreiben. Diese werden in Meilensteine sortiert und bearbeitet, indem man neue Pull Requests erstellt und neue Funktionen dementsprechend vom Feature Branch in den Dev-Branch hinzufügt.

## 5. Vorgehensmodell

Für die Entwicklung unseres Projekts haben wir einen hybriden Ansatz gewählt, der sowohl Elemente iterativer als auch inkrementeller Verfahren enthält. Dieser Ansatz ermöglichte es uns, die Vorteile beider Modelle zu nutzen und gleichzeitig den spezifischen Anforderungen und Herausforderungen unseres Projekts gerecht zu werden.

### Mischung aus iterativem und Wasserfallmodell:

Unser Ansatz besteht darin, zu Beginn des Projekts einen umfassenden Plan zu erstellen, der alle Anforderungen und Ziele des Projekts umfasst. Dies entspricht dem klassischen Wasserfallmodell, das eine detaillierte Planung und Definition der Anforderungen vorsieht. Diese Vorgehensweise ist besonders wichtig, da unser Team zu Beginn mit der neuen Technologie im Backend unerfahren ist und aufgrund des kurzen Projektzeitraums eine klare Richtung benötigt, um sich zu orientieren.

### Iterativer Ansatz mit Meilensteinen:

Gleichzeitig erkannten wir jedoch, dass eine starre Anwendung des Wasserfallmodells möglicherweise nicht geeignet war, um die Herausforderungen zu bewältigen, die während der Entwicklung auftreten könnten. Daher entschieden wir uns für einen iterativen Ansatz, der es uns ermöglichte, flexibel auf Änderungen zu reagieren und schrittweise an der Entwicklung unserer Anwendung zu arbeiten. Wir definierten Meilensteine und erstellten Tickets für jeden Meilenstein, um den Fortschritt zu verfolgen und sicherzustellen, dass wir unsere Ziele erreichen.

### Umgang mit Unsicherheit und neuen Technologien:

Gerade im Hinblick auf die Unerfahrenheit im Umgang mit der Technologie im Backend haben wir festgestellt, dass ein iteratives Vorgehen von Vorteil ist. Durch die schrittweise Herangehensweise an die Entwicklung mit Java und Spring Boot konnten wir uns langsam an die Technologie herantasten. So konnten wir Fehler erkennen und beheben und gleichzeitig das Verständnis und die Fähigkeiten unseres Teams im Umgang mit der neuen Technologie ausbauen.

### Fazit:

Insgesamt hat sich dieser hybride Ansatz als sehr erfolgreich erwiesen, da er es uns ermöglichte, sowohl strukturierte als auch flexible Ansätze zu kombinieren, um den Anforderungen unseres Projekts gerecht zu werden.

## 6. Projektdokumentation

### 6.1. Aufgabenverteilung

Aufgabe	Projektmitglied
Projektleitung	Hong Lam Nguyen
Frontend	Hong Lam Nguyen Nico Liebig
Backend	Hong Lam Nguyen Peter Strößner Finja Adam
Design	Finn Geneviève Hieber
Dokumentation	Finn Geneviève Hieber Hong Lam Nguyen Finja Adam
Präsentationsfolien	Peter Strößner

**Projektzeitraum:** 12.03.2024 bis 05.05.2024

Die exakte Aufgabenverteilung lässt sich anhand der Roadmap in Kapitel 6.4. erkennen.

### 6.2. Herangehensweise an das Projekt

Unser Projektansatz begann mit einem Brainstorming, um eine geeignete Idee zu entwickeln. Wir diskutierten verschiedene Ideen und wählten schließlich eine aus, die unseren Anforderungen und Zielen entsprach.

Nachdem wir uns für eine Idee entschieden hatten, arbeiteten wir diese weiter aus und identifizierten mögliche Funktionen und Technologien, die für die Umsetzung des Projekts erforderlich waren. Dies ermöglichte es uns, eine klare Vision für das Projekt zu entwickeln und den Entwicklungsprozess voranzutreiben.

Der nächste Schritt war das Aufsetzen des Projekts, was die Einrichtung der Entwicklungs- und Laufzeitumgebung beinhaltet. Außerdem haben wir den Git-Workflow definiert und Coding-Richtlinien festgelegt, um eine konsistente Entwicklung zu gewährleisten.

## Verlauf:

Während des Projekts haben wir mehrere Meilensteine definiert, um den Fortschritt zu überprüfen und sicherzustellen, dass wir unsere Ziele erreichen. Diese Meilensteine umfassten die Entwicklung von Prototypen mit den ersten Funktionen, die Erweiterung dieser Funktionen und das Hinzufügen neuer Funktionen zur App.

Während des gesamten Prozesses haben wir uns regelmäßig abgestimmt und Updates über den Fortschritt ausgetauscht. Dies wurde durch Protokolle dokumentiert und die Aufgabenverteilung durch Issues organisiert.

Bei Problemen oder Hindernissen haben wir sofort Maßnahmen ergriffen und diese gemeinsam bearbeitet, um den Entwicklungsprozess nicht zu verzögern.

Ende des Projekts:

Am Ende des Projekts haben wir eine App entwickelt, die zwar noch klein und einfach ist, aber Potential für zukünftige Erweiterungen und Verbesserungen bietet. Diese App kann als Grundlage für weitere Entwicklungen in ähnlichem Stil dienen und bietet eine solide Basis für zukünftige Projekte.

## 6.3. Entscheidungsfindungsprozess

Dieser Ablauf beschreibt unsere Herangehensweise an das Thema Problemlösung. In der ersten Projektsitzung wurde eine klare Struktur definiert, nach der man Probleme identifizieren, analysieren und lösen kann. Die Struktur ist wie folgt aufgebaut:

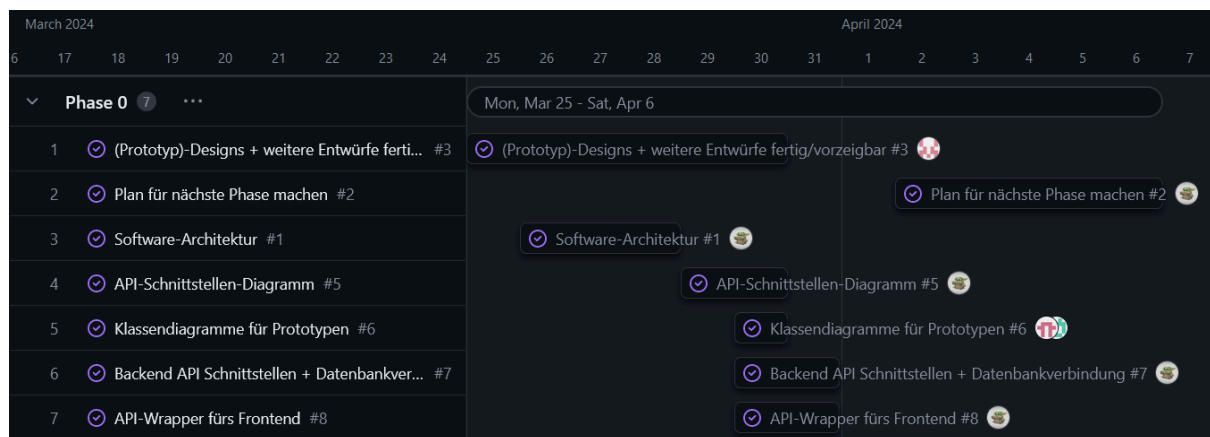
1. Ein Problem tritt auf: Ein Mitglied der Gruppe stößt bei seiner Arbeit auf ein Hindernis
2. Problem analysiert: Das gefundene Problem wird analysiert. Zu diesem Zweck stellt sich die Person drei Fragen:
  - Kann ich das Problem selbst lösen oder brauche ich Unterstützung von einer weiteren Person?
  - Ist das Problem ein kritisches Problem? Betrifft es die zentrale Infrastruktur? Bedroht es die rechtzeitige Abgabe des Projekts?
  - Ist eine Lösung des Problems dringend notwendig? Verhindert es, dass ich weiter an meiner Aufgabe arbeiten kann?
3. Wer kann mir helfen: Nachdem diese Fragen beantwortet wurden, war es nun von Relevanz herauszufinden, wer der Person beim Lösen des Problems helfen kann.

4. Zusammensetzen mit der entsprechenden Person
5. Szenarien durchspielen: Welche Lösung bietet das bestmögliche Ergebnis? Welche Lösung benötigt welchen Zeitaufwand? Abschließend muss eine Abwägung dieser beiden Fragen erfolgen.
6. Eine Entscheidung wird getroffen.

Diese Problemlösungsstrategie hat sich als hilfreich und effizient erwiesen und wurde im Verlauf des Projekts immer wieder angewendet.

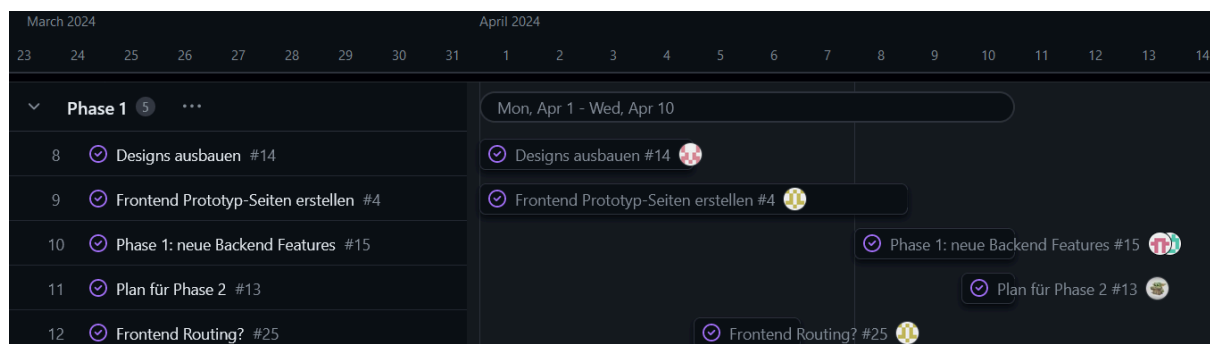
## 6.4. Roadmap

### Phase 0



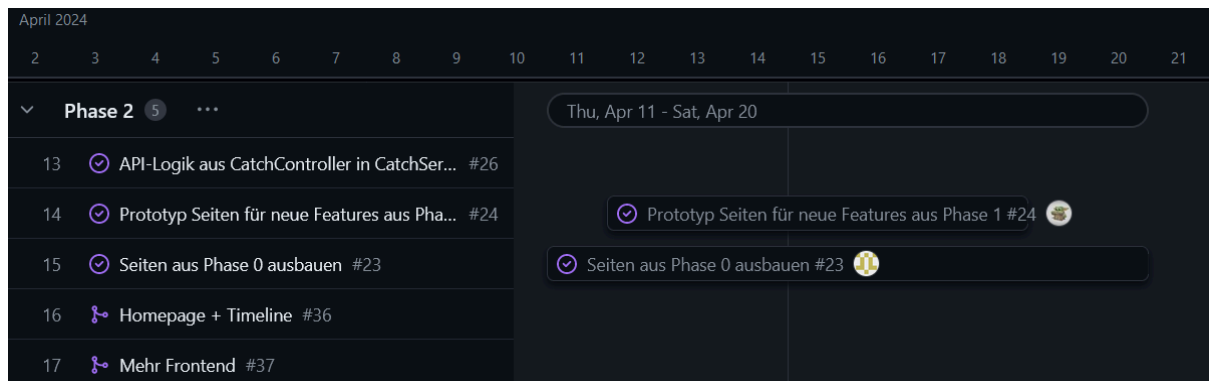
In Phase 0 wurden Design-Prototypen erstellt, die Software-Architektur geplant und wichtige API-Schnittstellen entworfen. Der Schwerpunkt lag auf der Konzeptualisierung und der Festlegung eines klaren Plans für die folgenden Entwicklungsphasen.

### Phase 1



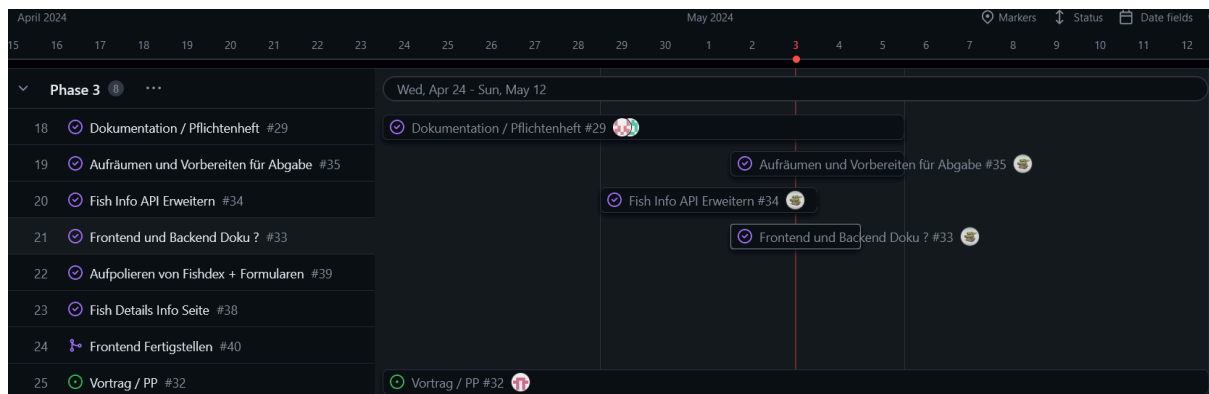
In Phase 1 wurde das Design erweitert, erste Frontend-Prototypen erstellt und neue Funktionen im Backend implementiert. Ziel war es, Frontend und Backend synchron zu entwickeln und die Routing-Struktur festzulegen.

## Phase 2



In Phase 2 wurde die API-Logik optimiert, Prototypen für neue Features erstellt und bestehende Seiten weiterentwickelt. Der Fokus lag auf der Verbesserung der User-Experience der Anwendung.

## Phase 3



Die letzte Phase konzentrierte sich auf die Dokumentation und die Vorbereitung für die Auslieferung. Die Fish Info API wurde erweitert, die Dokumentation verbessert und die Präsentation für die Abschlusspräsentation vorbereitet.



## 7. Backend und Frontend

### Frontend

Ziel ist es hier, die Daten aus dem Backend über APIs anzufragen und in der Website einzuarbeiten / Anfragen über API machen, dass bestimmte Daten gespeichert oder verändert werden im Backend.

### Backend

#### 1. API-Schnittstellen anbieten (Aufnehmen und Senden von Daten)

Die Anwendung verwendet APIs, um zwischen dem Frontend und dem Backend kommunizieren zu können. Das Ziel im Backend ist es, API-Schnittstellen zu erstellen, auf die das Frontend zugreifen kann, wenn es etwas will (sei es, um Daten anzufordern, um neue Daten zu speichern, um Daten zu verändern, ...).

#### 2. Klassen und Erstellung von Objekten (Verarbeiten von Daten)

Neben den APIs braucht die Anwendung auch die eigentlichen Klassen, von denen Objekte instanziiert werden. Jede Klasse hat dabei Attribute/Properties (Name, Datum, Fischart, ...) und Methoden (füge Fisch hinzu, ändere Namen), die aufgerufen werden können.

Die Informationen und Methoden dieser Objekte werden dann durch die API-Schnittstellen angeboten (gebe dem Frontend diese Daten, Ändere diese Daten im Backend, ...).

#### 3. Arbeit mit einer Datenbank (Speichern von Daten)

Daten müssen irgendwo gespeichert werden, um sie zum Beispiel beim nächsten Mal wieder auszulesen und anzuzeigen oder um zwischen unterschiedlichen Benutzern wechseln zu können. Im Falle von unserem Projekt verwenden wir JPA (Java Persistence API), um einfach mit einer H2 Datenbank zu arbeiten.

## 8. Projektergebnis

Nach Abschluss des Projektes stellt sich Fishdex als "Minimum Viable Product" (MVP) dar, das bereits als funktionierende Basisanwendung eingesetzt werden kann. Diese Version erfüllt die grundlegenden Anforderungen und bietet Hobby-Anglern die Möglichkeit, ihre Fänge zu dokumentieren und zu archivieren. Trotz des begrenzten Zeitrahmens kann festgestellt werden, dass die Applikation erfolgreich entwickelt wurde.

### Erweiterungsmöglichkeiten

Es gibt eine Reihe von Erweiterungen, die Fishdex weiter verbessern könnten. Dazu gehören Funktionen wie das Editieren und Löschen von Einträgen und Fischarten, die Implementierung eines Mehrbenutzersystems mit Registrierung und Login, eine vollständige Datenbank-Speicherung für Benutzerdaten sowie die Integration von Eingabevalidierung und Ortsangaben über Google Maps oder Koordinaten. Zusätzlich können vordefinierte Fischarten aus dem Internet bezogen werden, um die Vielfalt der verfügbaren Optionen zu erweitern.

### Projektverlauf

Im Laufe des Projektes ergaben sich einige Herausforderungen aufgrund des engen Zeitrahmens, insbesondere in Bezug auf die Einhaltung von Fristen und Zeitplänen. In der Endphase des Projekts musste zunehmend unter Zeitdruck gearbeitet werden, was die Integration weiterer Funktionen erschwerte. Die strukturierte Planung ging gegen Ende des Projektes leider verloren, aber die Herausforderungen wurden gemeistert. Probleme wurden gemeinsam erkannt und gelöst.

### Verbesserungsvorschläge für die Zukunft

Um zukünftige Projekte zu optimieren, ist eine verstärkte Kommunikation innerhalb des Teams unerlässlich. Bei auftretenden Problemen oder Fragen sollte ein sofortiger Austausch stattfinden, um Engpässe zu vermeiden und Termine einzuhalten. Des Weiteren sollte die Aufgabenverteilung durch den Teamleiter verbessert werden, um eine gerechtere Verteilung der Arbeitslast zu gewährleisten und das Team effizienter arbeiten zu lassen.

## 9. Anhang

- Anhang Designentwürfe unter /docs/Protokolle/Design

## 10. Unterzeichnung

Dresden, 03.05.24

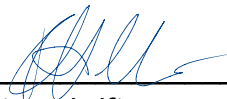
Ort, Datum



Unterschrift

Dresden, 03.05.24

Ort, Datum



Unterschrift

Dresden, 03.05.24

Ort, Datum



Unterschrift

Dresden, 03.05.24

Ort, Datum



Unterschrift

Dresden, 03.05.24

Ort, Datum



Unterschrift