

ONES

Online-**NE**nnungs-System

Projektmanager

Josha Landsmann

Frontend-Team

Tim Vahlbrock & Sam Nowakowski

Backend-Team

Nikos Epping & Marcel Weirather

Softwaretechnik 2

Sommersemester 2021



**Westfälische
Hochschule**

Inhaltsverzeichnis

1	Einführung	1
1.1	Das Projekt	1
1.2	Kontext	1
1.3	Ist-Zustand	2
2	Anforderungen	3
2.1	Funktionale Anforderungen	3
2.1.1	Registration & Anmeldung	3
2.1.2	Veranstaltungssuche	3
2.1.3	Wettbewerbsanmeldung	3
2.1.4	Ergebnisübersicht	3
2.1.5	Verwaltung eigener Daten	4
2.1.6	Verwaltung von Pferden	4
2.1.7	Bezahlungssystem	4
2.2	Nicht-funktionale Anforderungen	4
2.2.1	Responsivität	4
2.2.2	Sicherheit	4
2.2.3	Deployment	4
3	Projektplanung	5
3.1	MVPs	5
3.1.1	Scaffolding	5
3.1.2	Autorisierung	5
3.1.3	Veranstaltungsübersicht	6
3.1.4	Anbindung des ECM	6
3.1.5	Anmeldung bei Wettbewerb	7
3.1.6	Anzeigen von Ergebnissen	7
3.1.7	Anzeigen von Benutzerdaten	8
3.1.8	Verwaltung von Pferden	8
3.1.9	Filtern von Veranstaltungen	8

4 Architektur	10
4.1 Backend	10
4.1.1 Framework	10
4.1.2 Aufbau	10
4.1.3 Erweiterung und Tools	11
4.1.4 Schnittstellen	12
4.1.5 Datenbank	12
4.2 Frontend	14
4.2.1 Erweiterungen und Tools	14
4.2.2 Komponenten-Aufteilung	21
4.3 Deployment	24
4.4 Sicherheit/Authentifizierung	26
4.4.1 Allgemein	26
4.4.2 Registrierung	26
4.4.3 Verifizierung der E-Mail Adresse	26
4.4.4 Login	27
4.4.5 Zurücksetzen des Passworts	27
5 Reflexion	28
5.1 Planungsphase	28
5.2 Umsetzungsphase	28
5.3 Schlussphase	28
5.4 Finaler Stand	29
5.4.1 Login	29
5.4.2 Passwort vergessen	30
5.4.3 Registration	31
5.4.4 Veranstaltungen	33
6 Ausblick	36

1 Einführung

1.1 Das Projekt

Der Verein Deutscher Distanzreiter und -fahrer e.V. (VDD) bietet seinen Mitgliedern und anderen Interessierten auf ihrer Webseite ein Portal für Veranstaltungen und Wettbewerbe. Dieses Portal ist jedoch lediglich eine Übersicht und bietet einem Interessierten nicht die Möglichkeit sich für einen Wettbewerb anzumelden. Dazu ist das Ausfüllen eines Formulars notwendig, welches als PDF-Datei vorliegt. Dieses Formular muss vorher vom Veranstalter manuell mit den Daten des Wettbewerbs angereichert werden. Das ausgefüllte Formular ist dann per Mail oder per Post an den Veranstalter zu verschicken. Der Veranstalter muss diese Anmeldungen dann händisch sammeln und in Teilnehmerlisten überführen. Diese Teilnehmerlisten werden mit Hilfe von Excel verwaltet. Nach einem Wettbewerb werden aus dieser Liste dann die Ergebnislisten erstellt und erneut händisch veröffentlicht.

1.2 Kontext

Das ONES (**O**nline-**N**ennungs-**S**ystem) ist ein Softwareprojekt für den VDD, welches das bisherige Verfahren digitalisieren soll. Dazu soll zunächst eine Benutzerverwaltung im ONES implementiert werden. Bei dieser können sich Benutzer registrieren, und anmelden. Außerdem soll es eine Übersicht von Veranstaltungen geben, die man durchsuchen und filtern kann. Zu jeder Veranstaltung existieren ein oder mehr Wettbewerbe. Ein Benutzer soll sich bei einem solchen Wettbewerb über eine entsprechende Maske anmelden können, ohne dass er ein händisch Formular ausfüllen muss. Desweiteren soll dem Benutzer Einsicht in seine bisherigen Ergebnisse bei vergangenen Veranstaltung gewährt werden. In dieser Übersicht sollen alle teilgenommenen Wettbewerbe des letzten Jahres aufgelistet werden. Das Verwalten der Benutzer- und Anmeldedaten soll dem Benutzer ebenfalls ermöglicht werden. Der Benutzer soll dadurch seine persönlichen Daten, sowie seine Anmeldedaten ändern können. Neben seinen eigenen Daten soll es dem Benutzer auch möglich sein, seine Pferde zu verwalten, mit denen er sich bei Wettbewerben anmelden kann.

1.3 Ist-Zustand

Zur Verwaltung von Mitgliedern, Veranstaltungen, Pferden und anderen Daten gibt es bereits ein bestehendes System: Das ECM. Dieses System bietet einem Veranstalter die Möglichkeit seine Veranstaltung in die Liste auf der Vereinswebseite einzutragen, und das vorbereitete Anmeldeformular hochzuladen. Die Vereinsmitglieder haben auf dieses System jedoch keinen Zugriff. Außerdem bietet das System zu Beginn des Projektes noch keine Schnittstelle die das ONES nutzen kann.

2 Anforderungen

2.1 Funktionale Anforderungen

2.1.1 Registration & Anmeldung

Es soll einem Vereinsmitglied des VDD möglich sein sich im neuen System mit seiner VDD-Nummer und seiner E-Mail-Adresse zu registrieren. Neue Mitglieder können sich durch Eingabe von persönlichen Daten auch ohne eine VDD-Nummer bei ONES registrieren. Beim letzteren Fall erstellt das ECM ein neues Mitglied. Nach der erfolgreichen Registration kann sich ein Benutzer beim ONES mit seiner E-Mail-Adresse und seinem Passwort anmelden. Die Benutzerverwaltung liegt seitens ONES. Abfragen an das ECM werden mit einer eindeutigen Kennung des Benutzers angereichert, um benutzerspezifische Daten abfragen zu können.

2.1.2 Veranstaltungssuche

Ein Benutzer soll Veranstaltungen einsehen und durchsuchen können. Neben einer Listenansicht über mehrere Veranstaltungen, gibt es auch eine Detailansicht für jede Veranstaltung. Auf dieser Detailansicht gibt es eine Übersicht von Wettbewerben, welche Teil der Veranstaltung sind. Das Bereitstellen der Daten und das Filtern geschieht seitens des ECM.

2.1.3 Wettbewerbsanmeldung

Ein Benutzer kann sich mit einem oder mehreren Pferden bei einem Wettbewerb anmelden. Dabei gibt es verschiedene Zulassungsregeln. Den eigentlichen Anmeldeprozess sowie die Zulassungsprüfung übernimmt das ECM.

2.1.4 Ergebnisübersicht

Ein Benutzer soll die Möglichkeit haben, seine eigenen Ergebnisse von vergangenen Wettbewerben aus dem letzten Jahr einzusehen. Dabei sollen dort Teilnahmedaten wie Rittdauer oder auch Geschwindigkeit angezeigt werden.

2.1.5 Verwaltung eigener Daten

Ein Benutzer soll seine eigenen Daten einsehen können. Dazu zählen seine VDD-Nummer, sein Name, seine Adresse, seine Telefonnummer, sowie seine E-Mail-Adresse.

2.1.6 Verwaltung von Pferden

Ein Benutzer soll Pferde verwalten können, auf die er Zugriff hat. Dabei ist das Einsehen eines Pferdes, sowie das Registrieren eines Pferdes erlaubt, weder das Löschen noch Bearbeiten eines Pferdes soll möglich sein.

2.1.7 Bezahlungssystem

Ein Benutzer soll nach der Wettbewerbsanmeldung direkt über ONES seine Zahlungsmethode auswählen, und die Zahlung abschließen können.

2.2 Nicht-funktionale Anforderungen

2.2.1 Responsivität

ONES soll sowohl von mobilen Endgeräten als auch von einem Desktop erreichbar und bedienbar sein. Außerdem soll ONES über eine Webseite erreichbar sein.

2.2.2 Sicherheit

ONES soll verschlüsselt mit dem Server kommunizieren, damit keine dritten Daten bei Anfragen abfangen können. Außerdem sollen Benutzerdaten verschlüsselt gespeichert werden. Zudem soll es für einen Benutzer nicht möglich sein, auf die Daten von anderen Benutzern zuzugreifen.

2.2.3 Deployment

ONES soll auf dem Server installiert werden, auf dem auch das ECM läuft.

3 Projektplanung

3.1 MVPs

3.1.1 Scaffolding

Zeitpunkt:

Geplant	Real
22. April 2021	22. April 2021

Inhalt:

- Die Anwendung soll eine leere aber strukturierte Anwendung sein.
- Die Anwendung kann schon gebaut und getestet werden.
- Anwendung kann mit Hilfe von statischer Codeanalyse auf Qualität überprüft werden.
- Die Anwendung wird bei Änderungen automatisiert getestet und gebaut.

3.1.2 Autorisierung

Zeitpunkt:

Geplant	Real
13. Mai 2021	06. Mai 2021

Inhalt:

- Die Anwendung bietet dem Benutzer die Möglichkeit sich anzumelden.
- Die Anwendung bietet dem Benutzer die Möglichkeit sich zu registrieren.
- Die Anwendung bietet dem Benutzer die Möglichkeit sein Passwort zurückzusetzen, falls er dies vergessen hat.
- Die Anwendung verbietet unautorisierten Benutzern den Zugriff auf bestimmte Teile.

3.1.3 Veranstaltungsübersicht

Zeitpunkt:

Geplant	Real
20. Mai 2021	07. Juli 2021

Inhalt:

- Der Benutzer hat eine Übersicht von Veranstaltungen.
- Der Benutzer kann sich Details zu einer einzelnen Veranstaltung anschauen.
- Der Benutzer kann Veranstaltungen sortieren.
- Der Benutzer kann Veranstaltungen paginieren.
- ~~Der Benutzer kann Veranstaltungen filtern.~~

Das Filtern von Veranstaltungen wurde in ein späteres MVP ausgelagert.

3.1.4 Anbindung des ECM

Zeitpunkt:

Geplant	Real
09. Juni 2021	30. Juni 2021

Inhalt:

- Der Benutzer bekommt Veranstaltungen aus dem ECM angezeigt
- Das System registriert neue Benutzer im ECM
- Das System registriert bestehende Benutzer über das ECM

Auf Grund von starken Komplikationen bei der Kommunikation mit dem ECM hat sich dieses MVP verschoben.

3.1.5 Anmeldung bei Wettbewerb

Zeitpunkt:

Geplant	Real
28. Mai 2021	-

Inhalt:

- Der Benutzer kann sich bei einem Wettbewerb anmelden
- Bei der Anmeldung kann der Benutzer die Pferde angeben mit denen er teilnehmen möchte

Auf Grund der späteren Änderungen von Anforderungen und Datenstrukturen hat sich dieses MVP soweit verschoben, dass es nicht mehr innerhalb des Projektzeitraums umgesetzt werden konnte.

3.1.6 Anzeigen von Ergebnissen

Zeitpunkt:

Geplant	Real
26. Mai 2021	-

Inhalt:

- Der Benutzer kann sich die Ergebnisse seiner bisherigen Wettbewerbe anschauen
- Der Benutzer kann diese Ergebnisse sortieren
- Der Benutzer sieht die Gesamtstrecke
- Der Benutzer sieht die durchschnittliche Durchschnittsgeschwindigkeit

Auf Grund der vergleichsweisen niedrigen Priorisierung hat sich dieses MVP soweit verschoben, dass es nicht mehr innerhalb des Projektzeitraums umgesetzt werden konnte.

3.1.7 Anzeigen von Benutzerdaten

Zeitpunkt:

Geplant	Real
09. Juni 2021	-

Inhalt:

- Der Benutzer kann Information zu seiner Person einsehen

Auf Grund der vergleichsweisen niedrigen Priorisierung hat sich dieses MVP soweit verschoben, dass es nicht mehr innerhalb des Projektzeitraums umgesetzt werden konnte.

3.1.8 Verwaltung von Pferden

Zeitpunkt:

Geplant	Real
23. Juni 2021	-

Inhalt:

- Der Benutzer kann seine Pferde einsehen
- Der Benutzer kann ein neues Pferd registrieren

Auf Grund der vergleichsweisen niedrigen Priorisierung hat sich dieses MVP soweit verschoben, dass es nicht mehr innerhalb des Projektzeitraums umgesetzt werden konnte.

3.1.9 Filtern von Veranstaltungen

Zeitpunkt:**Inhalt:**

- Der Benutzer kann Veranstaltungen nach Startdatum filtern

Geplant	Real
07. Juli 2021	07. Juli 2021

- Der Benutzer kann Veranstaltungen nach Enddatum filtern
- Der Benutzer kann Veranstaltungen nach Region filtern
- Der Benutzer kann Veranstaltungen nach Kategorie filtern
- Der Benutzer kann Veranstaltungen nach Kartenritt filtern
- Der Benutzer kann Veranstaltungen nach Internationaler Ritt filtern

4 Architektur

Das ONES ist eine Web-Anwendung nach Server-Client Konzept. Es besteht aus einem Spring-Boot¹ Backend und einem Vue.js² Frontend. Das Backend stellt sowohl das Frontend selbst als auch die REST API die dieses verwendet zur Verfügung.

4.1 Backend

4.1.1 Framework

Im Backend wird das Spring-Framework eingesetzt. Um die Konfiguration von Spring zu vereinfachen wird auf die „Konvention vor Konfigurationen“-Lösung Spring-Boot gesetzt. Im Gegenteil zum „klassischen“ Spring, wo die Konfiguration über XML-Dateien geschieht, kann eine Spring-Boot-Anwendung komplett im Java-Code konfiguriert werden.

Desweiteren bringt Spring-Boot von Haus aus eine vereinfachte Maven-Konfiguration mit und stellt mehrere nicht-funktionale Features bereit.

4.1.2 Aufbau

Jede Komponente auf die das Frontend zugreifen können soll, besitzt die Komponenten Controller, Services und Models.

- **Models**

Models werden dazu genutzt Daten der Komponente zu übergeben und zu verarbeiten. Da es bei Anfragen an das ECM vorkommen kann, dass mehr Daten oder Daten in einem anderen Format ankommen, werden nur die benötigten Daten strukturiert, verarbeitet und an die API ausgegeben. Models werden nach dem Entwurfsmuster „**Data Transfer Object**“ umgesetzt.

- **Services**

Der Service der Komponente stellt die Logik für den Controller, in Form eines Interfaces, bereit.

¹<https://spring.io/projects/spring-boot>

²<https://vuejs.org/>

- **Controller**

Der Controller nutzt den Service der Komponente und implementiert die Schnittstellen für das Frontend.

4.1.3 Erweiterung und Tools

- **Hibernate**

Die Library Hibernate wird in der Persistenzschicht eingesetzt. Es agiert hier als ORM (Object Relational Mapping) und abstrahiert den Zugriff auf die Datenbank, sodass unterschiedliche DBMS eingesetzt werden könnten. Beispielsweise wird bei der lokalen Entwicklung standardmäßig eine in-memory H2-Datenbank verwendet, während im Staging-Betrieb eine PostgreSQL-Datenbank eingesetzt wird.

Desweiteren übernimmt Hibernate die Datenvalidierung. Hibernate berücksichtigt `javax.validation` Annotations und prüft vor dem Persistieren der Daten, ob diese valide sind. Damit wird beispielsweise verhindert, dass ein Nutzer mit einer ungültigen E-Mail-Adresse gespeichert wird.

- **Mapstruct**

Zum einfacheren „mappen“ von Daten aus dem ECM auf die zuvor angesprochenen Models wird Mapstruct genutzt. Dabei muss der Entwickler nur ein Interface erstellen und angeben, welche Felder aus dem Quellobjekt in welche Felder im Zielobjekt übertragen werden müssen. Mapstruct verwendet nicht Reflection, wie andere Bibliotheken, sondern generiert zur Compile-Zeit den Code, der für das Mapping verwendet wird.

- **Liquibase**

Um das Datenbankschema nicht von Hand erstellen zu müssen, kommt Liquibase zum Einsatz. Liquibase vereinfacht die Migration der Datenbank und ist, wie Hibernate, mit verschiedenen DBMS nutzbar. Die Datenbankmigrationen (also die einzelnen Schritte, die nötig sind, um das Datenbank-Schema zu erstellen) können in XML- oder YML-Dateien angegeben werden. Dies hat den Vorteil, dass diese Dateien auch mit versioniert werden können.

- **SLF4J / Logback**

Im Java-Umfeld wird für die Implementierung von Logging meistens die SLF4J (Simple Logging Facade for Java) verwendet. Wie der Name schon vermuten lässt, handelt es sich dabei lediglich um eine API, für die verschiedenste Implementierungen ausgewählt werden können.

In diesem Fall wurde als Implementierung Logback verwendet. Logback kann über eine XML-Datei flexibel abhängig vom laufenden Anwendungsprofil konfiguriert werden. Beispielsweise soll im Profil *development* nur auf die Standardausgabe geloggt werden, im Produktivbetrieb allerdings zusätzlich in eine Log-Datei mit täglichem rollover.

- **OpenAPI**

OpenAPI (früher: Swagger) ist ein weit verbreiteter Standard um Schnittstellen zu beschreiben. Wozu in diesem Fall genau OpenAPI verwendet wird, ist im Abschnitt 4.1.4 genauer erläutert.

4.1.4 Schnittstellen

Die Kommunikation zwischen Server und Client erfolgt über eine REST API.

Um sicherzustellen, dass die Endpunkte vom Frontend richtig bedient werden und die DTOs im Frontend zu denen im Backend passen, wird der Code für den Zugriff auf die API im Frontend automatisch generiert.

Im Backend wird automatisch eine OpenAPI-Spezifikation im JSON-Format generiert. Ein Maven-Plugin (*openapi-generator-maven-plugin*) liest diese Spezifikation ein und generiert Typescript-Code, der mit AXIOS³ auf die Endpunkte zugreift.

Ebenso wird der Java-Code für den Zugriff auf das ECM-System automatisiert aus einer OpenAPI-Spezifikation generiert.

4.1.5 Datenbank

Das ONES-Backend betreibt eine eigene Datenbank, um eine Verwaltung von Benutzern zu ermöglichen. In dieser Datenbank wird zum einen in der User Tabelle die Be-

³<https://github.com/axios/axios>

nutzer mit E-Mail-Adresse und Passwort gespeichert, sowie die erfolgreiche Registrierung mit dem dazugehörigen Registrations-Code. Zum anderen wird in der "Passwort vergessen"-Tabelle der Code zum Zurücksetzen des Passworts gespeichert und dessen Gültigkeitsdauer. Die restliche Verwaltung von Veranstaltungen, Pferden, etc. liegt im Zuständigkeitsbereich des ECM.

4.2 Frontend

Das Frontend des ONES ist ein Web-Client der auf dem Vue.js-Framework (Vue) basiert. Vue ist eines der aktuell meist verwendeten HTML, CSS und JavaScript Frameworks im Web-Bereich. Neben diesem Fakt war die geringe Bundle-Size von Vue-Projekten, gegenüber beispielsweise Angular-Projekten, für die Verwendung ausschlaggebend. Wie die meisten anderen Frameworks auch arbeitet Vue komponentenbasiert.

4.2.1 Erweiterungen und Tools

- **Vue CLI**

Die Vue CLI vereinfacht das Verwalten und insbesondere das Anlegen eines Vue-Projekts. Beim Anlegen eines Projekts wird dem Nutzer bereits angeboten in das Vue-Projekt weitere Libraries und Tools einzubinden. Auf diesem Wege wurde für ONES NPM, TypeScript, Vue Class Components, SASS, Vuex, Vuex Class Components, ESLint, ESLint Plugins für Vue, Jest und TS-Jest mit eingebunden.

- **Node Package Manager (NPM)**

NPM ermöglicht einfaches Dependency-Management für JavaScript Projekte. Die Abhängigkeiten werden strukturiert in der package.json gespeichert, sodass sie auch einfach versioniert werden können. Über die NPM Scripts (*npm run ...*) *serve*, *build*, *test:unit* und *lint* kann der HotReload-Server, der Production Build, die Unit Tests oder die statische Code-Analyse gestartet werden.

- **TypeScript & Vue Class Components**

TypeScript ist ein JavaScript Superscript, das bedeutet TypeScript erweitert die JavaScript Syntax. Durch die Verwendung von TypeScript kann nicht nur statisch typisiertes JavaScript entwickelt werden, sondern auch Polyfills zur Unterstützung älterer Browser durch den Compiler mit eingebunden werden. Mit der Verwendung von Vue Class Components geht einher, dass Klassen nach ES6 Class Syntax definiert werden.

- **ESLint**

ESLint ist ein Tool zu statischen Code Analyse von JavaScript. Wie bei Vue beinhaltet der Core nur das nötigste und wird über Plugins an den Anwendungsfall

angepasst. Für ONES wurde das Vue- und das TypeScript-Plugin verwendet. Das Linting kann über `npm run lint` gestartet werden. Einige Probleme werden dabei automatisch gefixt. Dies kann mit `-no-fix` unterbunden werden.

- **Jest & TS-Jest**

Jest ist ein Test-Framework, das ursprünglich für das React Framework von Facebook entwickelt wurde, sich aber ohne Probleme auch unabhängig von React verwenden lässt. Mit Jest können neben Unittests unter anderem auch Integrationstests geschrieben. Im Fall von ONES wurde sich jedoch auf ersteres beschränkt. Angelehnt an verschiedene andere Test-Frameworks können in Jest verschiedene Syntaxen zur Definition von Test-Suites und Test-Cases verwendet werden. Für ONES wurden die folgende Syntax verwendet:

```
1 describe("ClassUnderTest", () => {  
2     // Test Suite  
3  
4     beforeEach(() => {  
5         // setup  
6     });  
7  
8     afterEach(() => {  
9         // teardown  
10    });  
11  
12    it("a certain thing", () => {  
13        // test code  
14    });  
15 });  
16  
17
```

- **Syntactically Awesome Stylesheets (SASS, SCSS)**

SASS und SCSS sind zwei auf CSS basierende Styling Sprachen, mit dem das Styling in größeren Projekten deutlich vereinfacht wird. Beispielsweise können Styling Gruppen wie im folgenden Beispiel geschachtelt werden.

```
1 header {  
2     nav {  
3         background-color: magenta;  
4     }  
5 }
```

```
5     }  
6
```

Beide Sprachen werden im Build-Prozess zu CSS kompiliert und unterscheiden sich durch einzelne syntaktische Details, wobei SCSS CSS stärker ähnelt. Es wurde für ONES daher SCSS verwendet.

- **Tailwind CSS**

Tailwind CSS ist eine Bibliothek um Elemente durch das Setzen von CSS-Klassen gestalten zu können. Wird einem HTML-Element beispielsweise die Klasse `w-32` verliehen, erhält das Element die entsprechenden CSS-Regeln für eine Breite von 8rem. Darüber hinaus können für verschiedene Auflösungen unterschiedliche Regeln angegeben werden.

- **Axios & OpenAPI Axios Generator**

Axios ist ein promise-basierter HTTP Client, der die Kommunikation mit REST-Schnittstellen deutlich vereinfacht. Ein GET-Request an den Endpunkt `/users/` wird in Axios beispielsweise wie folgt formuliert.

```
1     async function fetchUsers() {  
2         await axios.get('/users/');  
3     }  
4
```

Um die Aufrufe der Endpunkte nicht selbst schreiben und typisieren zu müssen, wurde der OpenAPI Axios Generator verwendet, der, anhand der aus Swagger exportierten API-Beschreibung, entsprechenden TypeScript-Code generiert. Dazu muss das ONES-Webapp Backend gestartet und `/api/docs/v3` im Browser aufgerufen werden. Der Inhalt wird in der `api-docs.json` eingefügt. Durch `mvn install` im `ones-frontend` Ordner werden die entsprechenden Endpunkte generiert. Weil die Base-URL in der `base.ts` dabei jedes mal überschrieben wird, muss diese Datei nach der Generierung auf den versionierten Stand zurückgesetzt werden.

- **Vue-i18n** Weil die Benutzeroberfläche von ONES in Deutsch und Englisch verfügbar sein soll und um den angezeigten Text von den Komponenten zu separieren, wurde die Vue-i18n Bibliothek verwendet. Die Texte für die beiden Sprachen werden, mit passenden Schlüsselwörtern, jeweils in eine entsprechende JSON-Datei eingetragen. In der Komponente wird über `$t('key.to.use')` das Schlüssel-

wort *key.to.use* durch die entsprechende Übersetzung ersetzt. Muss in der Übersetzung ein Wert mit angezeigt werden, kann dieser an der passenden Stelle zwischen Unterstrichen eingetragen werden. In der Komponente wird als zweiter Parameter dann ein JSON-Objekt mit allen benötigten Parametern übergeben. Verwendet der Browser eine nicht-unterstützte Sprache, wird Englisch verwendet.

```

1 // de.json
2 {
3   "ronny": {
4     "cranePlaces": "Kranplätze müssen verdichtet sein.",
5     "tapeMeasure": "Haben die kein Bandmaß wat
6     _tapeMeasureLength_m lang ist?",
7     "peopleNotDoingTheirJob": "Die Leute kommen einfach ihrer
8     Arbeit nicht nach.",
9   }
10 }
```

```

1 // en.json
2 {
3   "ronny": {
4     "cranePlaces": "Crane places must be compacted.",
5     "tapeMeasure": "Don't they have a tape measure thats
6     _tapeMeasureLength_m long?",
7     "peopleNotDoingTheirJob": "People are simply not doing
8     their jobs.",
9   }
10 }
```

```

1 <span>{{ $t("ronny.cranePlaces") }}</span>
2 <span>{{ $t("ronny.peopleNotDoingTheirJob") }}</span>
3 <span style="text-transform: uppercase;">{{ $t("ronny.
4 tapeMeasure", { tapeMeasureLength: 8 }) }}</span>
```

• MomentJS

Weil das Rechnen mit Zeit und verschiedenen Zeit-Formaten meistens ziemlich

kompliziert ist, verwenden wir dazu die weit verbreitete Bibliothek MomentJS. Über *moment()* kann ein neues Moment Objekt des aktuellen Zeitpunkts erstellt werden. Wird dabei ein Parameter angegeben, wird dieser versucht als Zeitpunkt interpretieren zu werden. Über *moment.format()* kann das Moment Objekt in verschiedenen Formaten ausgegeben werden.

- **FontAwesome & FortAwesome**

Um die Optik und die Nutzbarkeit der Benutzeroberfläche zu verbessern, haben wir über das FortAwesome Vue Plugin die FontAwesome Icons in die Anwendung eingebunden. Jedes Icon besitzt einen Namen und kann durch die Font Awesome Icon Komponente eingebunden werden. Im folgenden Auszug wird das Icon `signInAlt` auf dem Login Button eingebunden.

```
1      <v-button type="submit">
2          Login
3          <font-awesome-icon :icon="'signInAlt'"/>
4      </v-button>
5
```

- **ZXCVBN**

Mithilfe der ZXCVBN Library kann sichergestellt werden, dass ein Benutzer ein sicheres Passwort wählt, ohne konkrete Regeln, wie „min. 8 Zeichen, 1 Sonderzeichen“ angeben zu müssen. Diese haben häufig zur Folge, dass ein Benutzer schwer merkbare oder häufig das selbe Passwort verwendet. Die Bibliothek unterbindet auch die Verwendung von weit verbreiteten Passwörtern, wie „12345678“ oder sich wiederholende Zeichenketten wie „BloodyMaryBloodyMaryBloodyMary“. Die Bibliothek arbeitet lokal, aber dennoch innerhalb von Millisekunden. Bei der Analyse werden Übersetzungen zu den beanstandeten Problemen und Tipps zur Änderung generiert, die dem Benutzer angezeigt werden. Außerdem wird ein Passwort-Score von 0-4 ermittelt, der in Form einer farbigen Leiste angezeigt wird. Bei Passwörtern, die nicht den Maximal-Score erreichen, wird die Registrierung unterbunden.

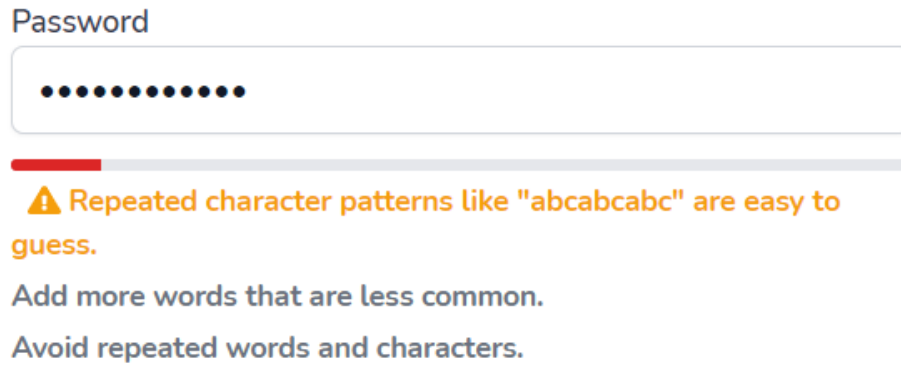


Abbildung 1: Password-Hinweise in ONES

- **Vuex**

Das Vue Plugin **Vuex** ermöglicht ein vereinfachtes State-Management in Vue Anwendungen. Die Daten werden dabei in sogenannten Stores abgelegt, die zwischen allen Komponenten geteilt werden. Jeder Store kann dabei einen State, Getter, Mutations, Actions und weitere Stores enthalten. Dabei bieten Getter die Möglichkeit computed Properties zu erstellen, Mutations verändern den State synchron und Actions denselbigen asynchron. In ONES wird State-Management für die Verwaltung der vom Server abgerufenen Daten verwendet. Außerdem werden auch einige komplexere Mechanismen, wie das korrekte Aufrufen der Endpunkte, oder der Zustand der Pagination mit in einen Store ausgelagert, um möglichst viel Geschäftslogik aus den Vue-Komponenten herauszuhalten.

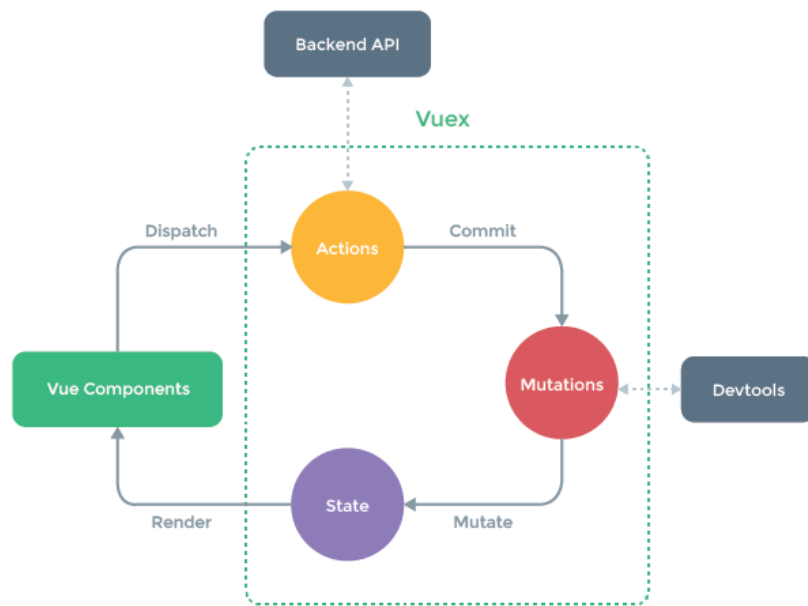


Abbildung 2: Vuex State Management (Quelle: Vuex Doku)

Standardmäßig verwendet Vuex - wie Vue Anwendung ohne vue class components auch - eine JSON basierte Syntax. Um wie in der restlichen Anwendung auch in den States mit der ES6 class Syntax arbeiten zu können, verwenden wir eine Wrapper-Bibliothek namens „vuex class components“. In der Ones Anwendung gibt es die folgenden Stores.

- **App**-Store zur Verwaltung des Anwendungsstatus (z.B. online/offline)
- **Event**-Store zur Verwaltung der Veranstaltungen und Wettbewerbe
- **Horse**-Store zur Verwaltung der Pferde
- **Result**-Store zur Verwaltung der Ergebnisse
- **User**-Store zur Verwaltung des Anmelde-Zustands und der Benutzer-Daten

4.2.2 Komponenten-Aufteilung

Im Sinne der Vermeidung von Duplicated Code sind sämtliche UI-Elemente abstrahiert und in einzelne Komponenten aufgeteilt. Tabellen sind beispielsweise als eigene Komponente definiert. Ein besonderer Vorteil dabei ist, dass diese responsiv sind. Im Fall von einem schmalen Bildschirm, sind Tabellen standardmäßig horizontal scrollbar, ohne das restliche Seitenlayout zu beeinflussen. Um die standardmäßigen HTML-Tags von den eigenen Komponenten zu unterscheiden, wird das v- Präfix verwendet.

```
1 <v-table>
2   ...
3 </v-table>
4
```

Listing 1: Aufruf von v-table

```
1 <template>
2   <div class="-my-2 overflow-x-auto sm:-mx-6 lg:-mx-8">
3     <div class="py-2 align-middle inline-block min-w-full
4 sm:px-6 lg:px-8">
5       <div class="shadow overflow-hidden sm:rounded-lg">
6         <table class="min-w-full divide-y divide-gray-200">
7           <slot></slot>
8         </table>
9
10        <div class="bg-white px-4 py-3 justify-between
11 border-t border-gray-200 sm:px-6" v-if="hasPagination()">
12          <slot name="pagination"></slot>
13        </div>
14      </div>
15    </div>
16  </template>
17
18 <script lang="ts">
19   import { Component, Vue } from "vue-property-decorator";
20
21   @Component
22   export default class VTable extends Vue {
23     hasPagination(): boolean {
24       return Boolean(this.$slots.pagination);
25     }
26   }
27 </script>
```



```
24     }  
25   }  
26   </script>  
27
```

Listing 2: Aufbau von v-table

Ein weiterer Vorteil ist, dass Funktionen ohne Redundanz hinzugefügt werden können. Im Fall der Tabelle kann eine Pagination optional hinzugefügt werden; Andere Komponenten, wie ein Passwort-Input `<v-password></v-password>`, zeigen bei der Eingabe die Komplexität des Passworts an, ohne dass diese bei jeder Verwendung explizit angegeben werden muss.

Event Delegation

In einigen Fällen ist es notwendig, Events von verschachtelten Komponenten zu delegieren. Der einfachste Fall ist v-input und zeigt im Folgenden das @input-Event.

```
1   <input class="..." v-bind:value="value" @input="handleInput" :  
   disabled="disabled" ...>  
2
```

Listing 3: Verwendung von v-input

```
1   handleInput(e: InputEvent): void {  
2     this.$emit("input", e.target.value);  
3   }  
4
```

Listing 4: Eventhandler für v-input

Verfügbare Komponenten

- dropdown/
 - dropdown
 - dropdown-link
- forms/
 - v-checkbox
 - v-hint

- v-label
 - v-password
 - v-radio
- pagination/
 - pagination
- table/
 - v-table
 - v-td
 - v-th
- auth-card
- badge
- card
- date-range
- feedback
- v-button
- v-link

4.3 Deployment

Es gibt vielfältige Möglichkeiten, die finale Anwendung auszuliefern. Das „klassische“ Setup wäre es, mit maven eine .war Datei generieren zu lassen, die dann in einem Tomcat oder anderen Java-Webserver ausgeführt werden kann.

Für die Entwicklung und Testphase von ONES wurde allerdings eine andere Variante gewählt. Dabei wurden Maven und Spring so konfiguriert, dass lediglich eine .jar Datei generiert wird. Diese kann einfach mit dem Befehl `java -jar ones-webapp.jar` gestartet werden, da sie einen eingebetteten Tomcat beinhaltet. Bei der Verwendung eines Linux-Servers wäre ein übliches Setup beispielsweise, eine systemd-Unit zu erstellen, die dann die Anwendung als Service auf dem System ausführt.

```
/etc/systemd/system/unit.d/ones-webapp.conf:
```

```
1 [Service]
2 ExecStart=java -jar /opt/ones-webapp.jar
```

Listing 5: Deploy von ones als systemd-Unit

Während der Entwicklung von ONES wurde für das Staging-Deployment allerdings ein anderes Setup gewählt. In die CI-Pipeline wurde ein Schritt integriert, der bei einem Merge einer Merge-Request in den master-Branch ein Docker-Image baut. Zuerst wird die finale .jar Datei in das Image kopiert. Anschließend wird der Befehl spezifiziert, der verwendet werden soll, um das Image zu starten:

```
CMD [„java“, „-jar“, „-Dspring.profiles.active=production“,  
„-Dspring.config.location=classpath:/,/config/“,„/opt/ones/ones-webapp.jar“]
```

Hier wird einerseits das aktive Profil auf „production“ gesetzt, außerdem wird angegeben, außerdem wird angegeben, dass zusätzlich zum Classpath (Dateien innerhalb der .jar) auch im Verzeichnis /config/ nach Konfigurations-Dateien gesucht werden soll. So kann eine Konfigurationsdatei erstellt werden, die z.B. die Platzhalter-Werte für die Zugangsdaten im Produktivbetrieb überschreibt.

Die ONES-Webanwendung selbst kommuniziert nur unverschlüsselt über HTTP. Hier ist also eine weitere Komponente notwendig. Da es allerdings eine Anforderung war, dass ONES später auf dem gleichen Server laufen soll, wie das ECM-System, war klar, dass hier ein Reverse-Proxy notwendig sein wird. Daher wurde es direkt so eingeplant,

dass dieser Reverse-Proxy auch die HTTPS-Zertifikate ausliefert und den TLS-Traffic terminiert und unverschlüsselt an ONES weitergibt. Im Staging-System wurde dabei der Reverse-Proxy „Traefik“ verwendet, der speziell für das Docker-Ökosystem ausgelegt ist.

4.4 Sicherheit/Authentifizierung

4.4.1 Allgemein

Die Authentifizierung eines Benutzers erfolgt über eine Kombination aus E-Mail-Adresse und Passwort. Die Benutzerdaten eines Bestandsmitgliedes des VDD sind im ECM bereits hinterlegt, die Benutzerverwaltung erfolgt jedoch vollständig über ONES. Die Verknüpfung der Daten wird über die Übereinstimmung aus E-Mail-Adresse, und zukünftig auch über die VDD Nummer hergestellt. Die Kernfunktionen des Systems sind nur für angemeldete Benutzer zugänglich. Benutzer, die nicht angemeldet sind, werden auf die Login Seite weitergeleitet.

4.4.2 Registrierung

Die Seite zur Registrierung kann durch das Klicken auf den entsprechenden Link beim Login, oder über die URL /register erreicht werden. Der Nutzer wird darauf hingewiesen, wenn vorhanden die bereits im ECM verwendete E-Mail-Adresse zu verwenden. Passwort-Regeln wie „mindestens 8 Buchstaben“, „mindestens 1 Sonderzeichen“ erzeugen nicht nur unbequeme Passwörter, sondern sind auch nicht in der Lage häufig verwendete Passwörter zu unterbinden. Die Passwort-Stärke wird daher mithilfe der zxcvbn-Bibliothek ermittelt, die neben der Passwort-Komplexität auch weitere Parameter, wie das Auftreten von Wiederholungen, oder die Verwendung von häufig verwendeten deutschen und englischen Passwörtern überprüft. Erhält das Passwort nicht den maximalen Score, wird die Registrierung unterbunden, aber entsprechende Tipps zur Verbesserung des Passworts angezeigt. Nach der Registrierung erhält der Benutzer eine E-Mail, mit welcher er seine E-Mail-Adresse verifizieren, und sein Benutzerkonto freischalten kann. Während der Registrierung kann der Benutzer jederzeit über einen Button zurück zum Login wechseln.

4.4.3 Verifizierung der E-Mail Adresse

Über seine E-Mail Adresse erhält der Benutzer nach der Registrierung einen Link, über welchen er sein Benutzerkonto freischalten kann. Im Anschluss kann er über einen Button die Login Maske aufrufen.

4.4.4 Login

Der Login erfolgt mit den bei der Registrierung festgelegten Anmeldedaten, sofern diese nicht vom Benutzer geändert wurden. Der Benutzer hat außerdem die Möglichkeit zu wählen, ob die Sitzung dauerhaft erhalten bleiben, oder beim Schließen des Tabs verworfen werden soll. Letzteres ist dabei der Standard.

Das Backend generiert bei erfolgreichem Login ein JSON Web Tokens (JWT)⁴ und schickt es als Antwort an das Frontend. Das Frontend speichert dieses Token entweder im local-storage oder im session-storage des Browsers, je nachdem ob der Benutzer angegeben hat, dass die Sitzung gespeichert werden soll. Bei allen anderen API-Anfragen wird dieses Token als *Authentication*-Header mitgeschickt. Das Backend kann das Token dann wieder dekodieren, auf Gültigkeit prüfen und die Anfrage einem eingeloggten Benutzer zuordnen.

4.4.5 Zurücksetzen des Passworts

Für den Fall dass der Benutzer sein Passwort vergessen hat, kann er sich einen Link schicken lassen, über den er ein neues Passwort vergeben kann. Dazu kann er in der Login-Maske auf „Passwort vergessen“ klicken und im Anschluss seine E-Mail eingeben. Bei der neuen Vergabe des Passworts gelten die selben Anforderungen wie bei der Registrierung.

⁴<https://jwt.io/>

5 Reflexion

5.1 Planungsphase

Nach Bekanntmachung des Projektes und der Gruppeneinteilung begann die Planungsphase für ONES. Die anfängliche Präsentation durch Frau Niemann führte zu einer grundsätzlichen Planung des Systems. Da die benötigte Schnittstelle weder vorhanden noch definiert war, wurde mit der Implementation der Registration begonnen, die unabhängig vom ECM geschieht. Parallel dazu entstand eine erste Übersicht von Veranstaltungen sowie eine dazugehörige Detailseite.

5.2 Umsetzungsphase

Nach einigen Wochen wurden die ersten Endpunkte der ECM-API veröffentlicht, sodass mit der Anbindung des ONES an das ECM begonnen werden konnte. Jedoch stellten sich die Anforderungen die anfangs gestellt wurden, als nicht finalisiert heraus, wodurch es immer wieder nach eigentlicher Fertigstellung von Views zu Änderungen dieser kam. Ebenfalls wurden Anpassungen oder neue Features der ECM-API teilweise widersprüchlich zum Besprochenen umgesetzt, wodurch eine zeitliche Verzögerung, durch den Bedarf einer erneuten Absprache und Anpassung seitens ONES, unvermeidlich war. Dadurch verzögerten sich viele Features immer wieder und verdrängten die Möglichkeit alle gewünschten Features im Projektzeitraum umzusetzen.

5.3 Schlussphase

Gegen Ende des Projektes wurde mit den Anfordernden eine Aufgabenliste erstellt, die die finalen Anforderungen an die besprochenen Teile der Software beinhalten sollte. Diese hatte zum Ziel eine klare Reihenfolge der Umsetzung vorzugeben, damit weitere Anforderungsänderungen nicht weiter aufhalten. Jedoch stellten sich seitens ECM erneut Komplikationen heraus, die zu erneuten Änderungen der Anforderungen führten. Daher wurde sich dazu entschieden, die Änderungen erneut anzupassen, um eine Anbindung an das ECM weiterhin zu ermöglichen. Dadurch kam es jedoch zu zeitlichen Verzögerungen die letztendlich dazu führten, dass die Nennung bei einem

Wettbewerb, eine der Hauptfunktionen von ONES nicht mehr innerhalb des Projektzeitraums umgesetzt werden konnte.

5.4 Finaler Stand

Im folgenden wird der finale Stand von ONES anhand von Screenshots dargestellt.

5.4.1 Login

Beim Aufrufen von ONES wird man zunächst zur Login-Maske geleitet.

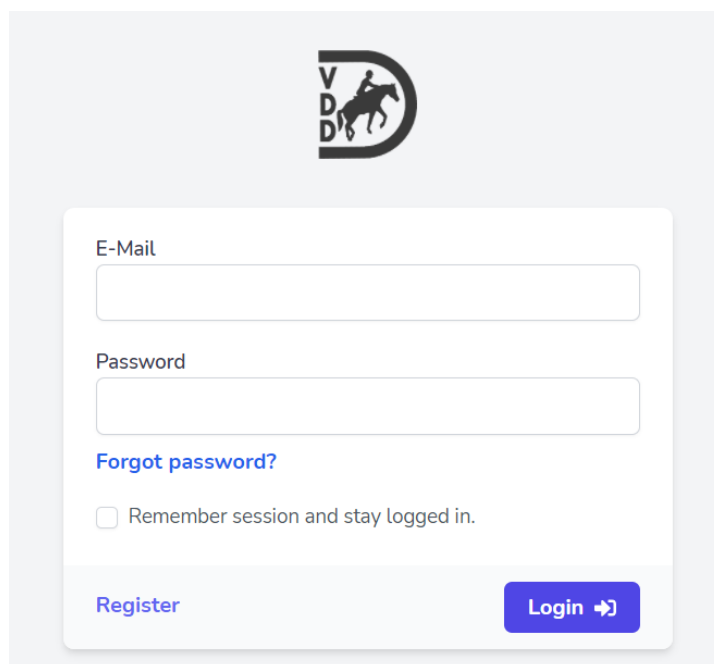


Abbildung 3: Login von ONES

Hat der Benutzer bereits ein Benutzerkonto, so kann er sich mit seiner E-Mail-Adresse anmelden. Falls der Benutzer sein Passwort vergessen haben sollte, kann er über die „Passwort vergessen“-Funktion ein neues Passwort festlegen. Hat der Benutzer kein Benutzerkonto kann dieser sich Registrieren und ein Benutzerkonto anlegen.

5.4.2 Passwort vergessen

Auf der „Passwort vergessen“-Maske kann der Benutzer die E-Mail-Adresse seines Benutzerkontos eingeben, um einen Link zum Zurücksetzen seines Passwortes zu erhalten.

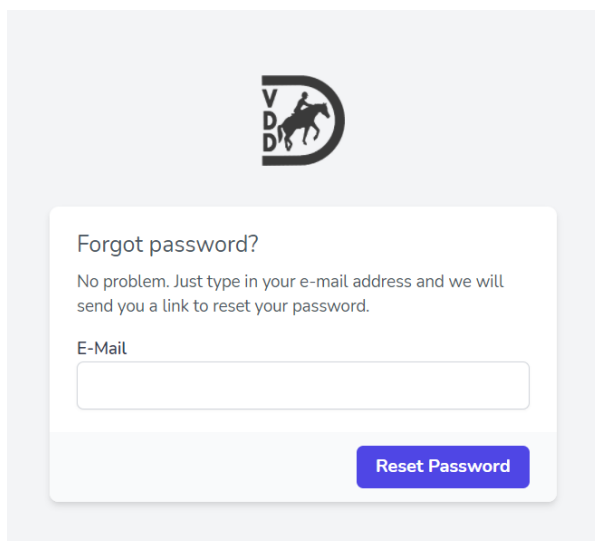
The screenshot shows a web interface for password recovery. At the top center is a logo consisting of a stylized 'D' with a horse silhouette inside, and the letters 'V', 'D', and 'D' stacked vertically to its left. Below the logo is a white rectangular form with a light gray border. Inside the form, the text 'Forgot password?' is displayed. Below this, a message reads: 'No problem. Just type in your e-mail address and we will send you a link to reset your password.' Underneath the message is a text input field labeled 'E-Mail'. At the bottom right of the form is a blue button with the text 'Reset Password' in white.

Abbildung 4: „Passwort vergessen“-Maske

Auch wenn die E-Mail-Adresse dem System nicht bekannt ist, wird der Benutzer trotzdem darüber informiert, dass eine E-Mail abgesendet wurde, damit nicht herausgefunden werden kann, ob eine E-Mail-Adresse bereits für ein Benutzerkonto verwendet wird.

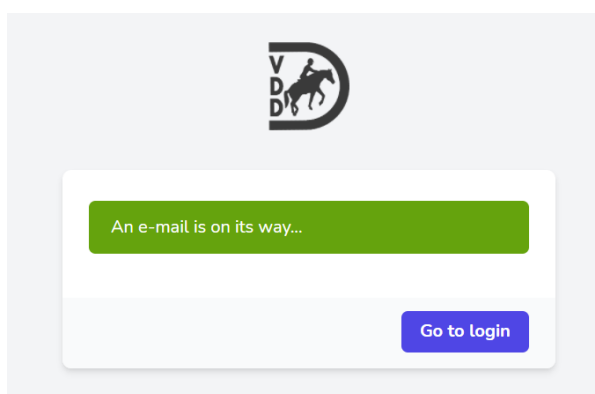
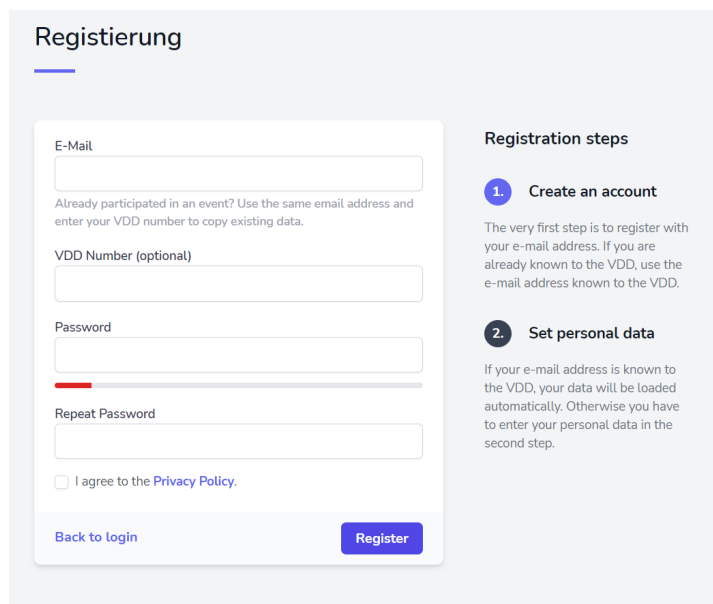
The screenshot shows the same web interface as before, but the form now displays a confirmation message. A green rectangular box with rounded corners contains the text 'An e-mail is on its way...'. Below this box is a blue button with the text 'Go to login' in white. The logo at the top remains the same.

Abbildung 5: „Passwort vergessen“-Email versendet

5.4.3 Registration

Um ein neues Benutzerkonto anlegen zu können, muss der Benutzer eine E-Mail-Adresse und ein Passwort eingeben. Optional kann er auch seine VDD-Nummer eingeben. Beim Registrationsprozess wird dem Benutzer zusätzlich eine Hilfestellung, in Form einer Rückmeldung zur Passwort-Stärke, gegeben. Außerdem wird er darauf hingewiesen, dass wenn er dem VDD schon bekannt ist, keine weiteren Daten eingeben muss. Ist der Benutzer dem VDD nicht bekannt oder seine Daten wurden noch nicht digital erfasst, muss er das nach der Registration nachholen. Um die Registration fortzusetzen, muss der Benutzer noch die Datenschutzbestimmungen akzeptieren.



The image shows a registration interface titled "Registrierung". On the left is a form with the following fields: "E-Mail" (with a note: "Already participated in an event? Use the same email address and enter your VDD number to copy existing data."), "VDD Number (optional)", "Password" (with a strength indicator bar), and "Repeat Password". Below the fields is a checkbox for "I agree to the Privacy Policy." and two buttons: "Back to login" and "Register". On the right, under the heading "Registration steps", there are two numbered steps: 1. "Create an account" (describing the first step of registering with an email address) and 2. "Set personal data" (describing the second step of entering personal data if the email is not known).

Abbildung 6: Registration

Nach den Eingaben wird dem Benutzer eine E-Mail zugeschickt, in welcher ein Link enthalten ist, um verifizieren zu können, dass er auch der Inhaber der E-Mail-Adresse ist.

Hello!

You are receiving this message because you signed up to ONES.

[To confirm your E-Mail address click the following link.](#)

If you did not sign up to ONES, just delete this message.

Abbildung 7: Bestätigungs E-Mail

Durch das Aufrufen des Links ist die Registration abgeschlossen.

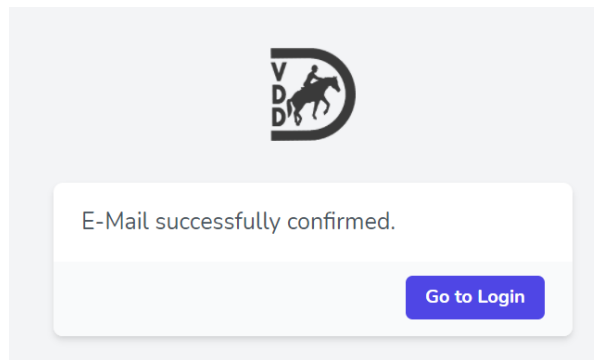
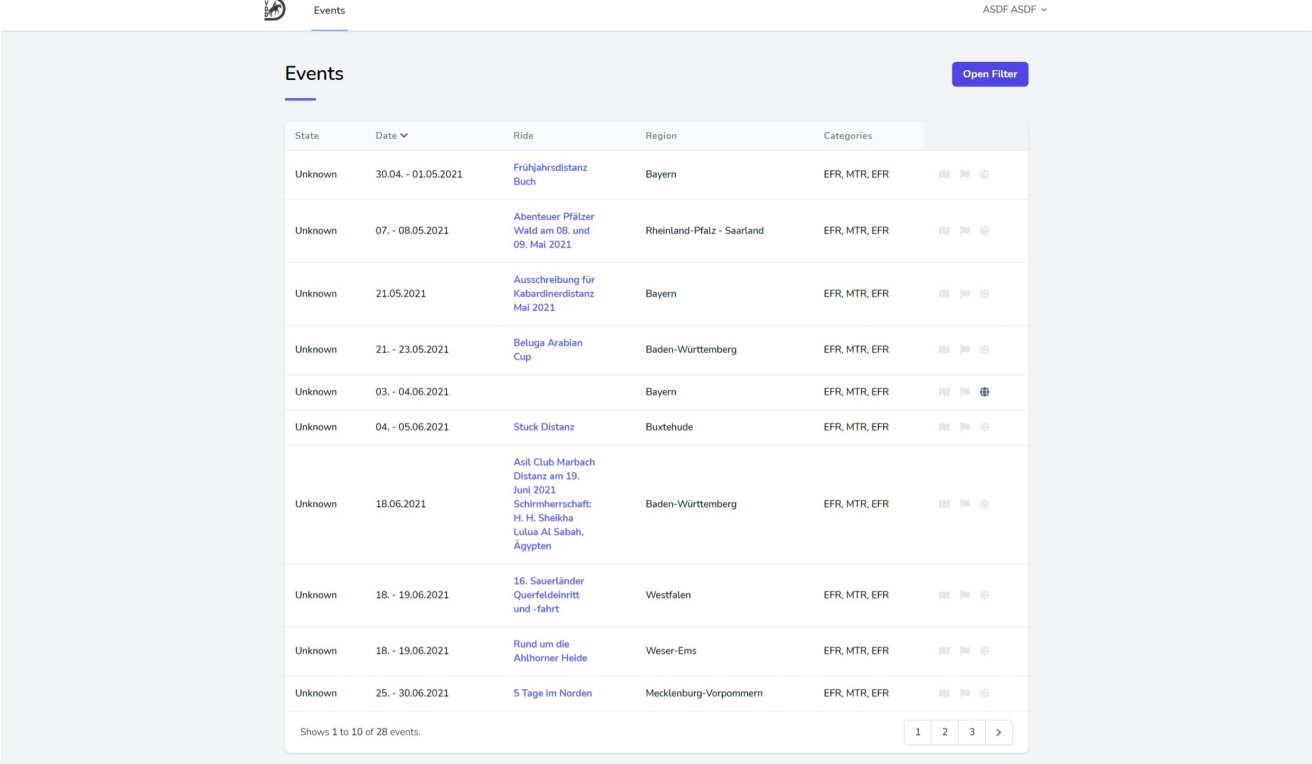


Abbildung 8: Registrations Abschluss

5.4.4 Veranstaltungen

Wenn der Benutzer sich erfolgreich angemeldet oder registriert hat, kann er nun alle verfügbaren Veranstaltungen sehen. Die verfügbaren Veranstaltungen werden tabellarisch aufgelistet.



Events

ASDF ASDF

Open Filter

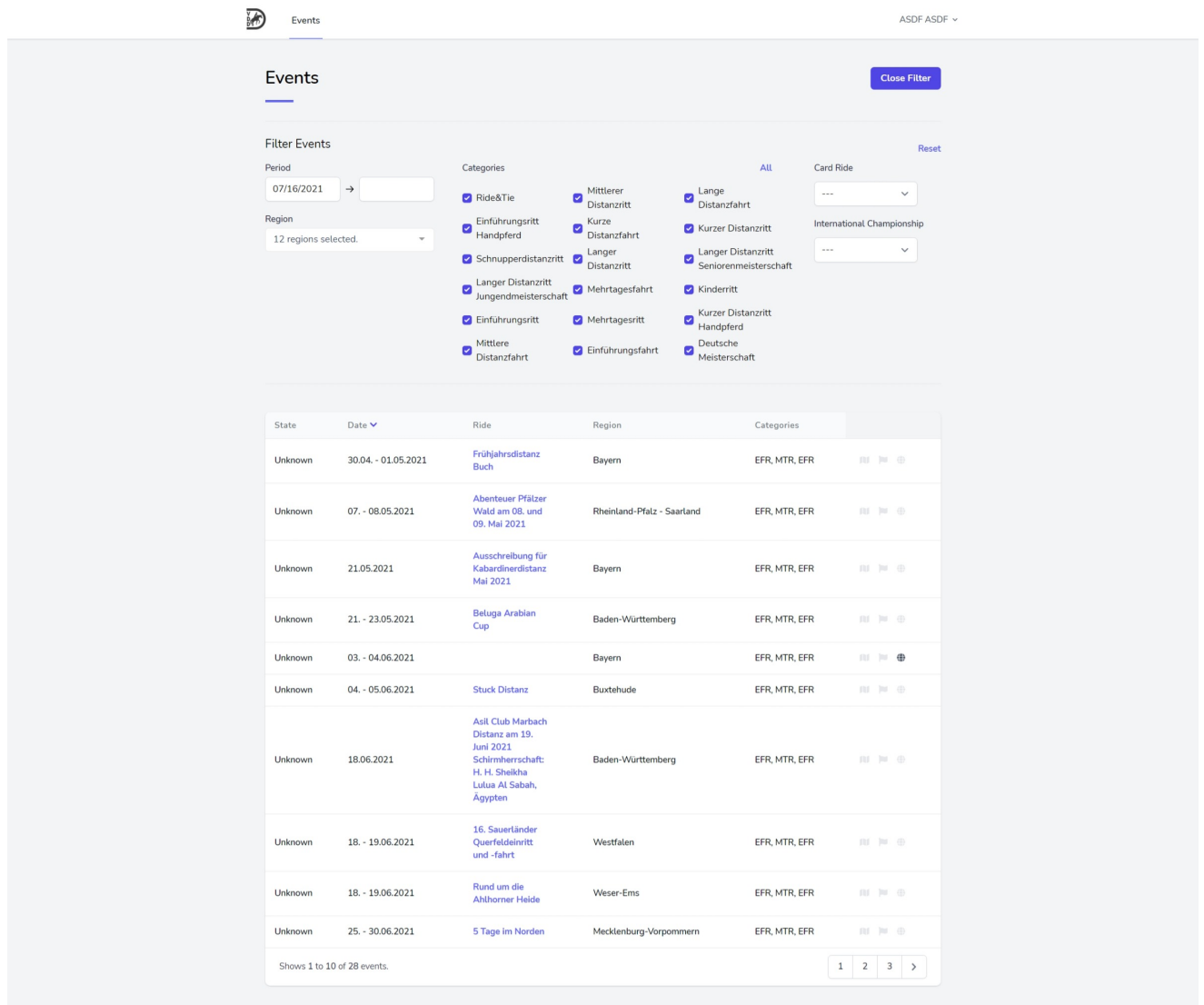
State	Date	Ride	Region	Categories
Unknown	30.04. - 01.05.2021	Frühjahrsdistanz Buch	Bayern	EFR, MTR, EFR
Unknown	07. - 08.05.2021	Abenteuer Pfälzer Wald am 08. und 09. Mai 2021	Rheinland-Pfalz - Saarland	EFR, MTR, EFR
Unknown	21.05.2021	Ausschreibung für Kabardinerdistanz Mai 2021	Bayern	EFR, MTR, EFR
Unknown	21. - 23.05.2021	Beluga Arabian Cup	Baden-Württemberg	EFR, MTR, EFR
Unknown	03. - 04.06.2021		Bayern	EFR, MTR, EFR
Unknown	04. - 05.06.2021	Stuck Distanz	Buxtehude	EFR, MTR, EFR
Unknown	18.06.2021	Asil Club Marbach Distanz am 19. Juni 2021 Schirmherrschaft: H. H. Sheikhha Lulua Al Sabah, Ägypten	Baden-Württemberg	EFR, MTR, EFR
Unknown	18. - 19.06.2021	16. Sauerländer Querfeldeinritt und -fahrt	Westfalen	EFR, MTR, EFR
Unknown	18. - 19.06.2021	Rund um die Althorner Heide	Weser-Ems	EFR, MTR, EFR
Unknown	25. - 30.06.2021	5 Tage im Norden	Mecklenburg-Vorpommern	EFR, MTR, EFR

Shows 1 to 10 of 28 events.

1 2 3 >

Abbildung 9: Übersicht aller Veranstaltungen

Da es bei einer Vielzahl von Veranstaltungen umständlich sein kann nach der persönlich passendsten Veranstaltung zu suchen, gibt es die Möglichkeit des Filterns von Veranstaltungen.




The screenshot shows a web application interface for filtering events. At the top, there's a navigation bar with a logo and the text 'Events'. A 'Close Filter' button is visible in the top right. Below the navigation bar, the 'Filter Events' section is active. It includes a 'Period' dropdown set to '07/16/2021', a 'Region' dropdown showing '12 regions selected', and a 'Categories' section with multiple checkboxes for different event types. A 'Reset' button is located to the right of the categories. Below the filters, a table lists events with columns for State, Date, Ride, Region, and Categories. The table contains 10 rows of event data. At the bottom, a pagination bar shows 'Shows 1 to 10 of 28 events.' and a set of navigation buttons (1, 2, 3, >).

State	Date	Ride	Region	Categories
Unknown	30.04. - 01.05.2021	Frühjahrsdistanz Buch	Bayern	EFR, MTR, EFR
Unknown	07. - 08.05.2021	Abenteuer Pfälzer Wald am 08. und 09. Mai 2021	Rheinland-Pfalz - Saarland	EFR, MTR, EFR
Unknown	21.05.2021	Ausschreibung für Kabardinerdistanz Mai 2021	Bayern	EFR, MTR, EFR
Unknown	21. - 23.05.2021	Beluga Arabian Cup	Baden-Württemberg	EFR, MTR, EFR
Unknown	03. - 04.06.2021		Bayern	EFR, MTR, EFR
Unknown	04. - 05.06.2021	Stuck Distanz	Buxtehude	EFR, MTR, EFR
Unknown	18.06.2021	Asil Club Marbach Distanz am 19. Juni 2021 Schirmherrschaft: H. H. Sheikh Lulus Al Sabah, Ägypten	Baden-Württemberg	EFR, MTR, EFR
Unknown	18. - 19.06.2021	16. Sauerländer Querfeldeinritt und -fahrt	Westfalen	EFR, MTR, EFR
Unknown	18. - 19.06.2021	Rund um die Ahlthorner Heide	Weser-Ems	EFR, MTR, EFR
Unknown	25. - 30.06.2021	5 Tage im Norden	Mecklenburg-Vorpommern	EFR, MTR, EFR

Abbildung 10: Filtern von Veranstaltungen

Wurde die passende Veranstaltung gefunden, kann dieses geöffnet werden, um mehr Details über die Veranstaltung zu erfahren.

 Events
 ASDF ASDF ▾

21. - 23.05.2021
0/60
Participants
Min. 20

←
Beluga Arabian Cup

Place and Time

Time Range 21. - 23.05.2021

Country Deutschland

Starting "Im Schloßle"

Place 89192 Rammingen/Lindenau
Deutschland

Coordinates 48°32'28.6"N 10°10'14.8"E
[📍](#)

Details

Sign up Deadline 07.05.2021

Sign Up After Deadline Possible No

Vaccination Requirements Unknown

Helmet Requirements Unknown

Host Website [Sign Up On Paper](#)

Notes

Tagesetappen bei MTR (A): 84 km / 62 km / 54 km es besteht 2-3 x pro Tag die Möglichkeit, den Ritt in der Wertung zu beenden.

Ansonsten vorzeitiges Beenden an • Tag 1 bei 32 km, 54 km, 67 km • Tag 2 bei 45 km, 55 km • Tag 3 bei 27 kmEs können auch einzelne Etappen (B/C/D) genannt werden, ohne die Möglichkeit vorzeitig in der Wertung zu beenden!

Zielort am Samstag: Fam. Allgaier,
Erschneuren Zielort am Sonntag: Fam. Weber

Contacts

Event Host
Baader Ina
Edelsteinstraße - 69198 Schriesheim (Deutschland)

Info and Sign Up
Baader Ina
Edelsteinstraße - 69198 Schriesheim (Deutschland)

📞 0049 177 8989891

@ ina.baader@web.de

Contests

CON	Distance	Date	Qualification-Stage	Age Horse	Age Participant	Stages	Markings	(max.) Participants	
LDR	84 km	21.05.	Q22	From 7		0		0 / ∞	Sign Up
MDR	62 km	22.05.	Q1	From 6		0		0 / ∞	Sign Up
KDR	54 km	23.05.	Q0	From 6		0		0 / ∞	Sign Up
MTR	200 km	21.05.	Q2	From 7		3		0 / ∞	Sign Up
KDR	54 km	21.05.	Q0	From 6		0		0 / ∞	Sign Up

Abbildung 11: Detail Ansicht einer Veranstaltung

6 Ausblick

Das Produkt wurde inklusive des zugehörigen Quellcodes am 15. Juli 2021 an die Anforderer übergeben und von diesen abgenommen. Da noch einige funktionale Anforderungen offen sind, die nicht im Projektzeitraum umgesetzt werden konnten, werden diese wohl als nächstes umgesetzt, damit ONES dann für die Mitglieder des VDD veröffentlicht werden kann. So folgt die Nennung bei einem Wettbewerb, eine Ergebnisübersicht, sowie eine Datenverwaltung der benutzereigenen Daten. Auch die Anbindung eines Bezahlsystems zur Bezahlung von Nenngeldern ist ursprünglich geplant gewesen, und könnte in Zukunft Teil von ONES werden.

Zur Umsetzung der noch fehlenden Funktionalitäten steht die Durchführung eines Modules „SWT 3“ im Wintersemester 2021/2022 in Aussicht.

Abbildungsverzeichnis

1	Password-Hinweise in ONES	19
2	Vuex State Management (Quelle: Vuex Doku)	20
3	Login von ONES	29
4	„Passwort vergessen“-Maske	30
5	„Passwort vergessen“-Email versendet	30
6	Registration	31
7	Bestätigungs E-Mail	32
8	Registrations Abschluss	32
9	Übersicht aller Veranstaltungen	33
10	Filtern von Veranstaltungen	34
11	Detail Ansicht einer Veranstaltung	35