

Dokumentation Masterprojekt

KML Vision – IKOSA Platform Subscription Management Interface

Melina Kampitsch, Anna Kramer, Verena Pichler

Inhalt

Zielsetzung 2

Mock-Up-Erstellung in Figma 3

Entscheidung des Entwicklungstools 7

Integration der Endpunkte im Backend 9

React-Admin Ergebnis..... 9

 Login-Screen 10

 Seitenansicht der Organisation 10

 Abonnements einer ausgewählten Organisation 11

 Formular Create..... 11

 Übersicht aller bestehenden Abonnementpläne 12

 Formular zur Erstellung eines neuen Abonnementplan..... 12

Reflexion 13

Referenzen 14

Abbildungsverzeichnis 15

Zielsetzung

Das Masterprojekt wurde in Zusammenarbeit mit KML Vision GmbH durchgeführt, welche mit ihrer IKOSA Plattform ein breites Portfolio von KI-basierten Algorithmen für unterschiedliche Bildanalysen im Life Science Bereich, cloud-basiert und ohne die Notwendigkeit lokaler Computer-Ressourcen anbietet. Das Geschäftsmodell der IKOSA Plattform basiert auf einem Abonnement-Modell für die verschiedenen IKOSA Services. Dadurch soll der Zugriff der Nutzerinnen und Nutzer auf die IKOSA Services verwaltet und kontrolliert werden. Ziel des Projektes ist die Entwicklung einer einfachen IKOSA Abonnement-Verwaltung um dem KML Vertriebspersonal die eigenständige Verwaltung von Abos zu ermöglichen. Dadurch sollen die intern bisher verwendeten SQL-Skripte ersetzt werden.

User-Stories

Main Requirements									
1. list organizations and list subscriptions and plans for a specific organization (master-detail)									
2. list custom subscription plans, create custom subscription plans and edit restrictions									
3. subscribe an organization to an existing subscription plan									
4. edit an existing subscription									
a. edit parameters: extend, auto-renewal, price, ...									
b. extend an existing subscription (copy subscription and set new period)									
c. cancel one or all subscriptions for an existing organization									
Tech Requirement									
1. Secure all operations for IKOSA users with a special global role (TBD, e.g. SALES)									
2. Tracking of changes by SALES users									

Abbildung 1 Vorlage für User-Stories

Wie in Abbildung 1 dargestellt, wurden die konkreten Anforderungen definiert. Um die Spezifikationen der Anforderungen für das Projekt zu bestimmen, wurden User Stories erstellt. Für die Erstellung einer einheitlichen Struktur, wurde folgende Vorlage verwendet: "As [ROLE] I want to [ACTION], so that [PURPOSE]". Hierbei wurde für jede Anforderung die benötigte Rolle, die Aktion sowie der Zweck eingefügt. Diese wurden ihrer jeweiligen Funktion (Login, View Organization details, Managing custom subscription plans and restrictions, Subscribe to an existing subscription plan und Manage an existing subscription) zugeordnet. Nachdem die User Stories erstellt waren, wurden sie anhand der Priorität in drei Releases (Minimum Viable Product (MVP), Version 2 und Version 3) aufgeteilt (siehe Abbildung 2). Der Fokus des Projektes lag auf der Umsetzung des MVPs. Alle weiteren Versionen könnten in einem zukünftigen Projekt oder intern von KML Vision umgesetzt werden.

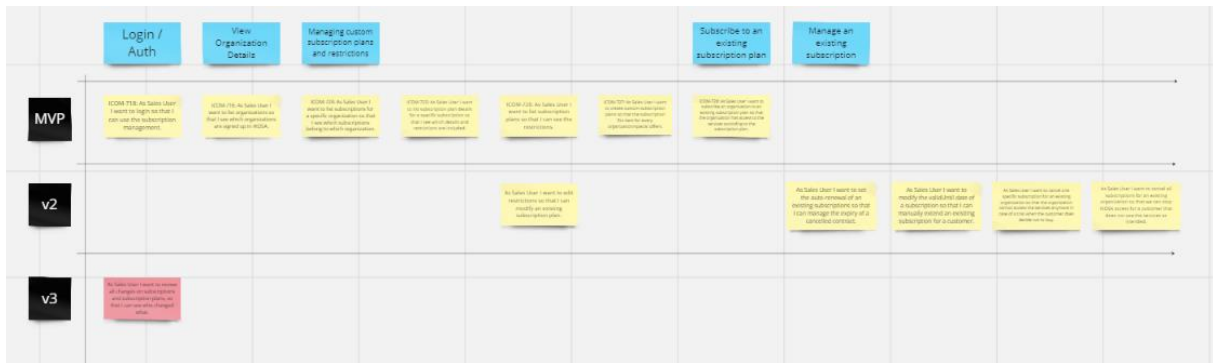


Abbildung 2 User-Stories im Miro

Nach Erstellung der User Stories wurden, wie in Abbildung 3 zu sehen, die dazugehörigen Akzeptanzkriterien erstellt. Für die Akzeptanzkriterien wurden jeweils die Voraussetzungen (given), die Aktion (when) und das Ergebnis (result) jeder Einzelnen User Story ergänzt.

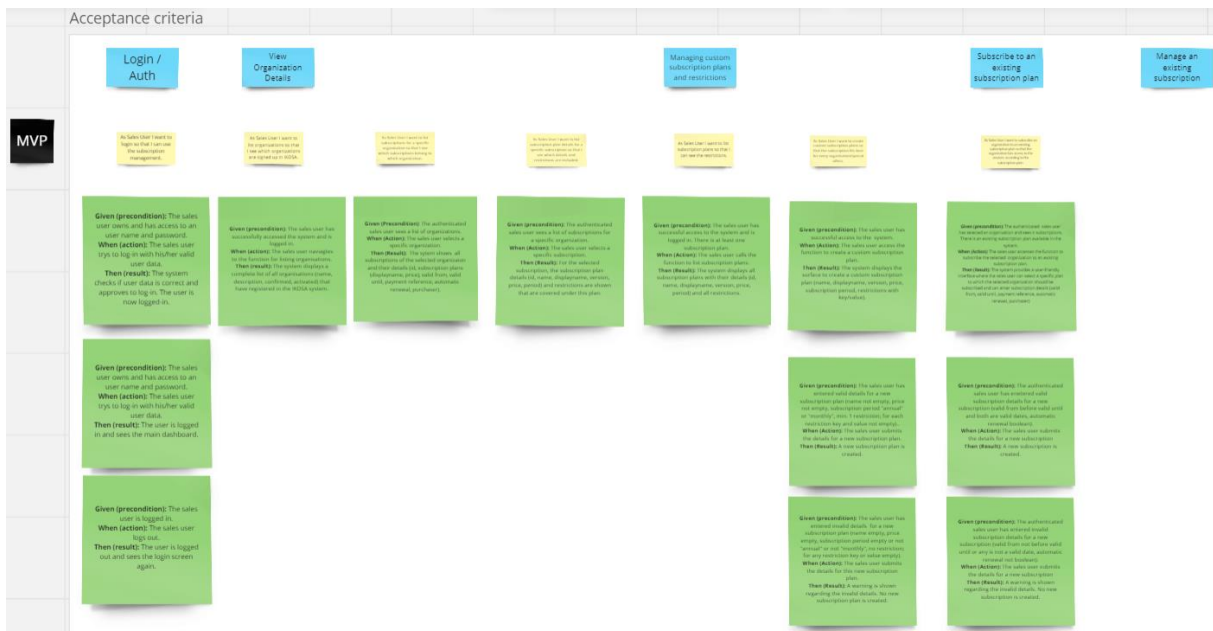


Abbildung 3 Akzeptanzkriterien im Miro

Mock-Up-Erstellung in Figma

Anhand der zuvor erstellten User Stories und Anforderungen wurde ein Mock-Up in Figma erstellt. Mittels dieses ersten Designs konnte man den Aufbau des zu entwickelnden Frontends graphisch abbilden und die Möglichkeiten der Darstellung sowie die Verknüpfung der einzelnen Seiten abwägen.

Wie mit KML Vision vereinbart, liegt der Fokus bei der Umsetzung des Projektes auf der Funktionalität. Das Design der Oberfläche wurde demnach nicht priorisiert.

Im Folgenden werden die Punkte des Mock-Ups dargestellt:

1. Um sich als autorisierte/r Nutzer/in auszuweisen, gilt es als essenziell sich vorerst einzuloggen. Demnach ist ein Login-Screen notwendig, bei welchem man sich mit den Benutzerdaten anmelden kann.
2. Nach der Anmeldung gelangt man zu einer Übersicht aller registrierten Organisationen. Diese Übersicht ist in Form einer Tabelle gestaltet, die Informationen wie ID, Namen, Beschreibung, Bestätigung und Aktivierung der Organisationen anzeigt. Zusätzlich besteht die Möglichkeit, durch Klicken auf "Subscription" weitere Details über die Organisation und ihre Abonnements zu erhalten. Eingefügte Links neben den Organisationen ermöglichen die Navigation zwischen den Organisationen und deren Abonnementplänen. Ein Suchfeld ermöglicht es, Organisationen anhand ihrer ID und/oder ihres Namens zu filtern.



Abbildung 4 Login-Screen in Figma

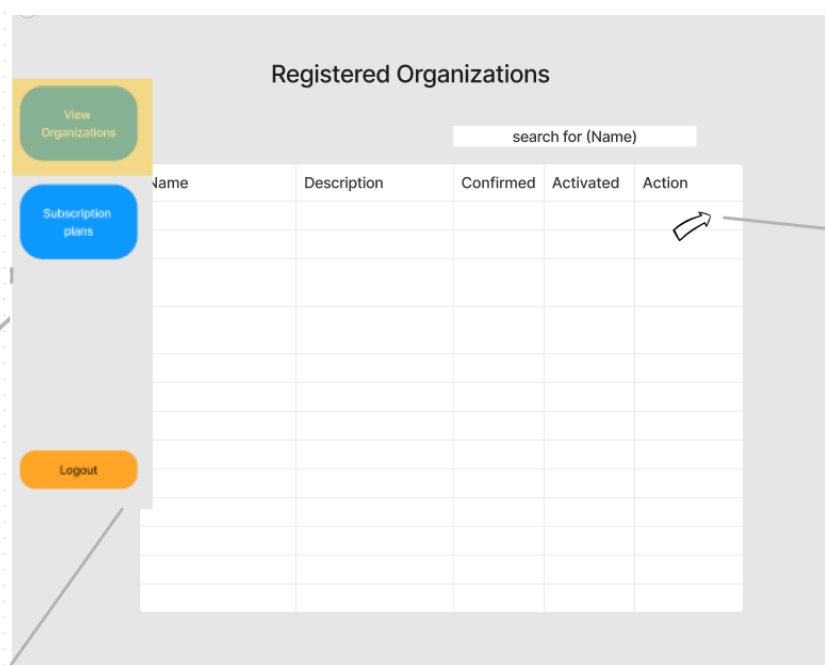


Abbildung 5 Übersicht mit allen registrierten Organizations in Figma

3. Auf der Seite der einzelnen Abonnements (siehe Abbildung 6) zu einer ausgewählten Organisation werden Informationen wie ID, Name, Preis, Gültigkeitsdatum von bis, Zahlungsreferenz, automatische Verlängerung und die E-Mail der jeweiligen Organisation angezeigt. Zudem besteht die Möglichkeit, die einzelnen Einschränkungen der ausgewählten Abonnements einzusehen.

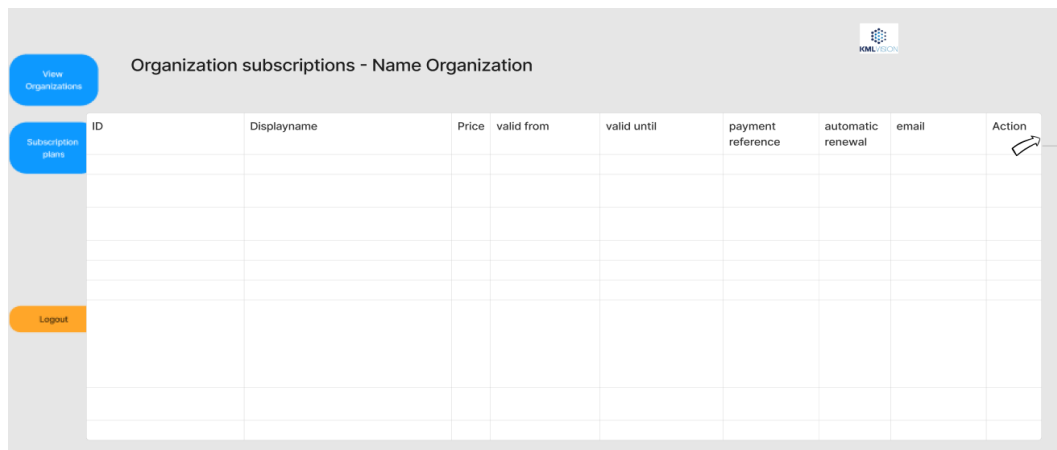


Abbildung 6 Auflistung der Subscriptions für eine ausgewählte Organization in Figma

4. Beim Klick auf die Schaltfläche "Details" gelangt man zu einer neuen Seite (siehe Abbildung 7), auf der die Details des Abonnementplans angezeigt werden. Dort werden Informationen wie ID, Name, Anzeigenname, Version, Nettopreis, Abonnementzeitraum und die Einschränkungen mit Schlüssel und Wert des ausgewählten Abonnementplans angezeigt.

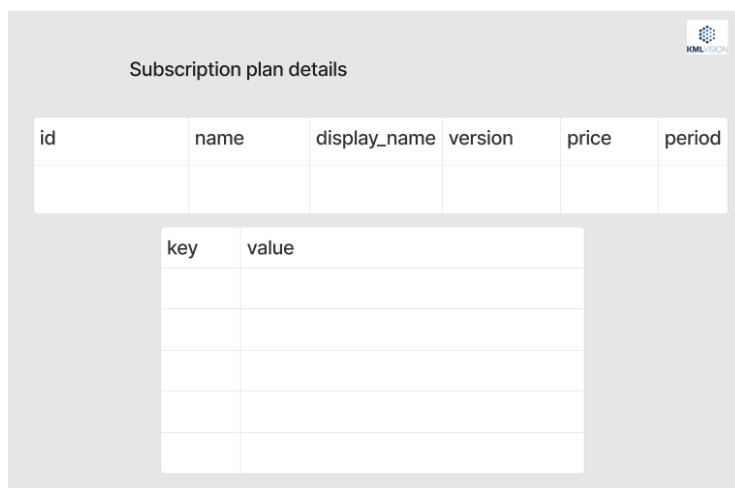


Abbildung 7 Übersicht über die Subscription Plan Details in Figma dargestellt

5. Durch die Auswahl der Option "Create" ist es möglich, ein neues Abonnement für die entsprechende Organisation anzulegen. Wie in Abbildung 8 zusehen, sind dabei folgende Angaben erforderlich: Der Name bzw. die ID der Organisation, die Abonnementplan-ID, das Gültigkeitsdatum von und bis, die Zahlungsreferenz und die Käuferin bzw. Der Käufer. Optional kann die Funktion "automatische Verlängerung" aktiviert werden.

Create new subscription

Organization ID/Name

Subscription Plan Name

valid from

valid until

payment reference

automatic renewal

purchaser

create subscription

Warning!

New Subscription is created

Abbildung 8 Erstellung einer neuer Subscription in Figma

6. Durch Klicken auf "Subscription Plan" in der linken Menüleiste öffnet sich die Übersicht der Abonnementpläne (siehe Abbildung 9). Über den Suchbereich über der Tabelle können die Abonnementpläne anhand ihrer ID und/oder ihres Namens gesucht werden. Zusätzlich stehen die Funktionen "Erstellen" und "Export" zur Verfügung. Dadurch ist es möglich, neue Abonnementpläne zu erstellen und die gesamten Daten zu exportieren.

View Organizations

Subscription Plans

create new subscription plan

search for (Name, DisplayName)

ID	displayname	version	netPrice	subscriptionPeriod	Action

Logout

Abbildung 9 Übersicht aller Subscription Plans in Figma dargestellt

7. Der Button "Create new Subscription Plan" leitet zu einer neuen Seite weiter, auf der die Möglichkeit besteht, einen neuen Abonnementplan hinzuzufügen. Wie in Abbildung 10 zu sehen, müssen dort der Name des Plans, die Version, der Nettopreis, die Abonnementdauer, der Anzeigename und die entsprechenden Einschränkungen angegeben werden, um einen neuen Abonnementplan zu erstellen.

Abbildung 10 Erstellung eines neuen Subscription Plans in Figma

Entscheidung des Entwicklungstools

Um die Entscheidung für das passende Framework für die Erstellung der Benutzeroberfläche zu treffen, wurde ein Vergleich verschiedener Frameworks durchgeführt. Dabei wurden sowohl die Vor- als auch die Nachteile jedes Frameworks erörtert, um eine fundierte Entscheidung zu treffen.

Es wurde eine umfangreiche Recherche über Retool, Quickbase, Budibase, Angular und React-Admin durchgeführt (siehe Abbildung 11). Im Austausch mit KML Vision wurden alle Vor- und Nachteile dieser Optionen diskutiert. Die Entscheidung fiel auf das React-basierte React-Admin Framework, da es weit verbreitet ist und viele Möglichkeiten zur Entwicklung eröffnet. Es handelt sich um ein JavaScript-basiertes Framework, in dem Benutzeroberflächen aus wiederverwendbaren UI-Komponenten erstellt werden können. Die Flexibilität und Benutzerfreundlichkeit im Vergleich zu anderen Tools sowie die Wiederverwendbarkeit des Codes spielten ebenfalls eine Rolle bei der Entscheidung. React wurde im Jahr 2013 von Facebook entwickelt und ist kostenlos verfügbar.

Für die Projektziele war es wichtig, dass die entwickelte Benutzeroberfläche öffentlich zugänglich ist, und dieselbe Authentifizierung wie die bestehende IKOSA Webapplikation unterstützt. Dabei sollte sie das öffentliche IKOSA-API nutzen und als eigenständige Komponente fungieren, die nicht in die bereits bestehende IKOSA Web-Benutzeroberfläche integriert ist. Der Fokus liegt hierbei bei den funktionalen Aspekten und nicht bei dem Design der zu erstellenden Oberfläche. All diese Anforderungen werden durch React sowie React-admin erfüllt. Zudem hat das Projektteam Interesse gezeigt, sich im Laufe des Projekts mit React auseinanderzusetzen zu wollen, um Erfahrung mit dem Framework zu sammeln.

Um die Entscheidung abzusichern, wurde nach Absprache vorab ein kleines Testprojekt erstellt, in dem sich mit den Optionen des Tools vertraut gemacht wurde.

	Option 1: angular	Option 2: react	Option 3: vue	Option 4: svelte	Option 5: flutter	Option 6: Vaadin	Option 7: Budibase	Option 8: Retool
Description	Build a SPA based on the angular framework using the generated typescript client.	Build a SPA based on the react framework using the generated typescript client.	Build a SPA based on the vue framework using the generated typescript client	Build a SPA based on the svelte framework using the generated typescript client	Build a SPA based on the flutter framework using the generated typescript client	Build a vaadin UI into the user-service	Build a UI with the no-code tool Budibase and connect to our API Use the self-hosted variant to run retool within our service cluster.	Build a UI with the no-code tool Retool and connect to our API. Use the self-hosted variant to run retool within our service cluster.
Pros and cons	<ul style="list-style-type: none"> + mighty framework + knowledge in the frontend team + used in other KML products - harder to start i opinionated with regards to architecture 	<ul style="list-style-type: none"> + steeper learning curve than angular + knowledge in the KML frontend team i non-opinionated with regards to architecture 	<ul style="list-style-type: none"> + similar to angular + more lightweight than angular - no knowledge in the frontend-team i opinionated with regards to architecture 	<ul style="list-style-type: none"> - no knowledge in the frontend-team 	<ul style="list-style-type: none"> + opinionated framework + enables mobile out of the box + no need to think about design + knowledge in the frontend team - prioritizes android / iOS, web supported but might not work that well - harder to start than react or vue 	<ul style="list-style-type: none"> + no need to setup a frontend project, everything can be coded in Java + provides default ui components (tables, buttons, dropdowns, ...) and default style - not used in KML products 	<ul style="list-style-type: none"> + self-hosted variant available + free tier for self-hosted variant available + self-hosted free-tier seems to be unlimited with regards to users + container image and kubernetes recipe available in the docs + wysiwyg editor for UI - requires to run a backend server and database - seems to not support OAuth authentication out of the box (but might be possible to implement via simple rest-calls) 	<ul style="list-style-type: none"> + self-hosted variant available + free tier for self-hosted variant available + docker-container and kubernetes recipes available in the docs + states to support OAuth authentication - requires to run a backend server and database - seems to be resource intense (2vCPU, 8GB RAM, 60GB storage) - free tier limited to 5 users
Estimated cost	LOW	LOW	LOW	LOW	LOW	LOW	LARGE	LARGE

Abbildung 11 Entscheidungstabelle Frontendframeworks (in Englisch aufgrund der Dokumentationsanforderungen für KML)

Integration der Endpunkte im Backend

Vor Erstellung des Frontends war es wichtig zu analysieren, welche Endpunkte intern bei KML Vision bereits existieren und welche noch ergänzt werden müssen. KML Vision entwickelt das öffentliche API von IKOSA mittels OpenAPI und benutzt für die öffentliche Dokumentation der Schnittstelle Swagger UI.

Anhand der Schnittstellendokumentation wurden jene Endpunkte identifiziert, die zusätzlich erstellt und integriert werden mussten. Im Folgenden sind alle notwendigen Endpunkte für das Projekt dargestellt, jene in der Farbe "gelb" wurden neu erstellt oder angepasst.

User

GET user details - whoAmI: /api/v1/users/whoami

GET list permissions of a user – data.roles.contains('SALES_USER')

Organization

GET all organizations (mit details) - listOrganizations: /api/v1/organizations

Subscription

GET list all subscriptions - listSubscription(s): /api/v1/{organizationId}/subscriptions

POST new subscription – createSubscription: /api/v1/{organizationId}/subscriptions

Subscription Plan

GET details of an available subscription plan - getSubscriptionPlan: /api/v1/subscriptionPlans/{subscriptionPlanId}

GET list available subscription plans - listSubscriptionPlans: /api/v1/subscriptionPlans

POST new subscription Plan - createSubscriptionPlan: /api/v1/subscriptionPlans

Die Struktur der Endpunkte wurde im Laufe des Projekts in OpenAPI definiert. Im Zuge der Projektdurchführung wurde entschieden, die Umsetzung der vom Projektteam definierten Endpunkte vom Projektteam zum KML Vision-Team zu übergeben, da eine Einführung in die Backend Entwicklung im Rahmen des Projektes nicht möglich gewesen wäre.

React-Admin Ergebnis

Im Anschluss an die Konzeptionierungsphase hat die konkrete Umsetzung des Projekts gestartet. Alle vorab erfassten Anforderungen und Ergebnisse wurden mit React-Admin umgesetzt.

Folgende Punkte sind Teil des Projekts:

Login-Screen

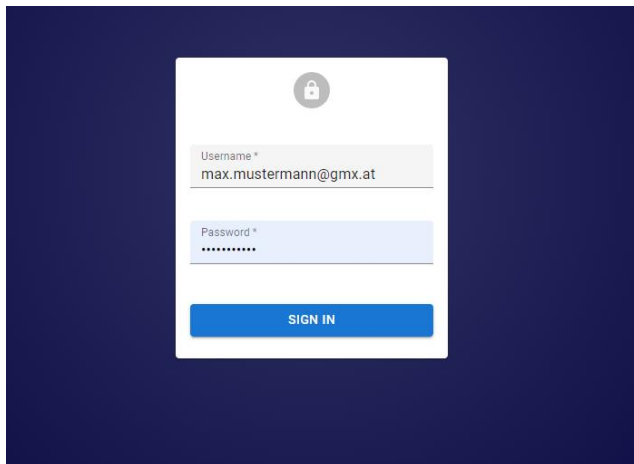


Abbildung 12 Login-Screen in React-Admin

Der Login-Screen ermöglicht die Anmeldung der Nutzerin und des Nutzers. Die Anmeldung beinhaltet im Hintergrund ebenso die Umsetzung des *OAuth2 refresh-token flow*, der sicherstellt, dass User eingeloggt bleiben, bis sie sich aktiv ausloggen.

Seitenansicht der Organisation

ID	Name	Description	Confirmed	Activated	
0df619945e99			✓	✗	SUBSCRIPTIONS + CREATE SUBSCRIPTION
9e5431e4d024			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
a50c70d8a724			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
6c3a1d77e2c4			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
5ee842ba9396			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
4d639a62f6fc			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
87072bacb7c7			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
474c0a0bc634			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
4068f0f54f38			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION
0b5fe8d5c488			✗	✓	SUBSCRIPTIONS + CREATE SUBSCRIPTION

Abbildung 13 Übersicht aller Organisationen in React-Admin

Nach erfolgreichem Login wird man direkt zur Seitenansicht der Organisation weitergeleitet. Hier sind alle verfügbaren Organisationen aufgelistet, und es besteht die Möglichkeit, sämtliche Daten zu exportieren. Zusätzlich kann man durch einen Klick auf "SUBSCRIPTIONS" die entsprechenden Abonnements der Organisation einsehen und mit "+ CREATE SUBSCRIPTION" ein neues Abonnement für die Organisation anlegen. Die Suchfelder für ID und Name sind deutlich erkennbar platziert. Außerdem ist es jederzeit möglich, zwischen den Organisationen und den Abonnementplänen zu navigieren.

Abonnements einer ausgewählten Organisation

ID	Name	Description	Confirmed	Activated
0df619945e99			✓	✗
9e5431e4d024			✗	✓
a5bc70d8a724			✗	✓
6c3a1d77e2c4			✗	✓
5ee842ba8396			✗	✓
4d639a62f6fc			✗	✓
87072bacb7c7			✗	✓
474c3a0bc834			✗	✓
4068f054f38			✗	✓
0b5fe8d5c488			✗	✓

Abbildung 14 Abonnements einer ausgewählten Organisation in React-Admin

Auf dieser Seite werden die Abonnements einer ausgewählten Organisation angezeigt. Bei dem Exportieren von den Daten, werden diese in ein Excel-File heruntergeladen. Durch Klicken auf "DETAILS" können genauere Informationen zu dem jeweiligen Abonnement eingesehen werden.

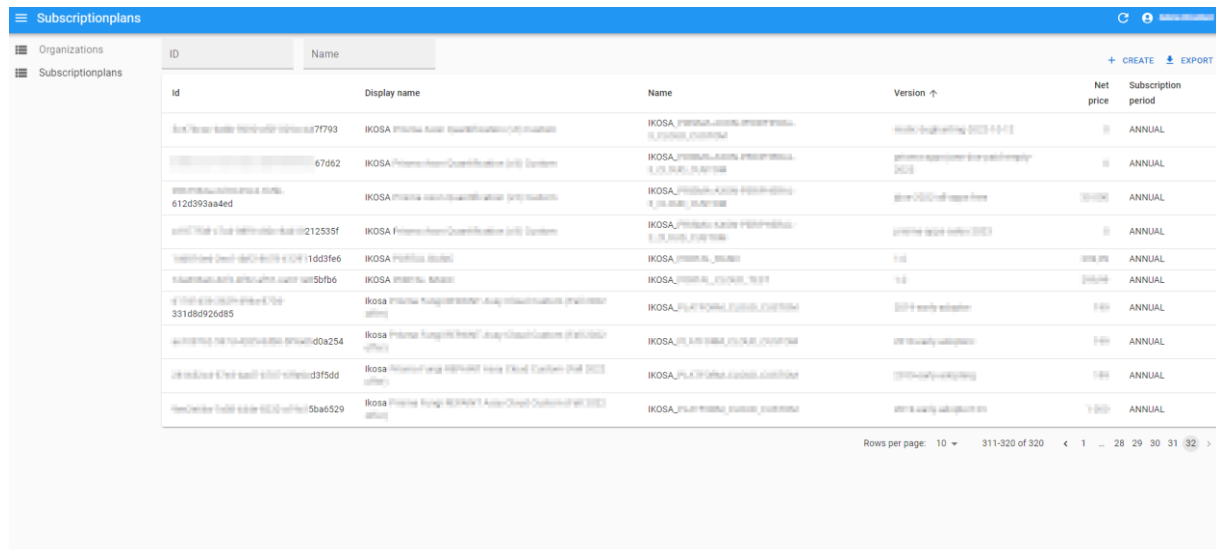
Formular Create

Abbildung 15 Formular "Create" in React-Admin

Auf diesem Bildschirm wird das "Create"-Fenster dargestellt. Hier sieht man, dass die Organisation bereits vorausgewählt ist und man mittels eines Dropdowns die jeweilige "Subscription Plan ID" auswählen kann. Bei dem Feld "Valid from" und "Valid until" wird eingestellt, von wann bis wann das Abonnement gültig ist. Die Leiste "Payment reference" wird benutzt, um die Zahlung korrekt zuordnen zu können. Durch die Aktivierung von "Automatic renewal", wird das Abonnement

automatisch verlängert, nachdem es abgelaufen ist. Die Leiste "Purchaser" bezieht sich auf die Käuferin bzw. den Käufer.

Übersicht aller bestehenden Abonnementpläne

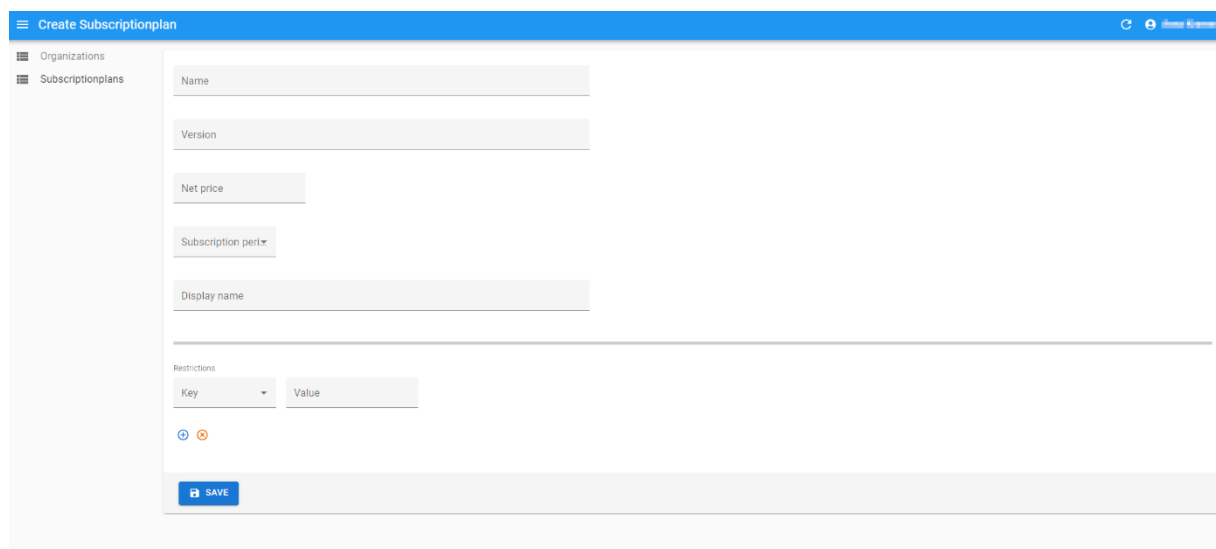


ID	Name	Version	Net price	Subscription period
67793	IKOSA Premium	1.0.0	1000	ANNUAL
67962	IKOSA Premium	1.0.0	1000	ANNUAL
612d393a4ed	IKOSA Premium	1.0.0	1000	ANNUAL
212535f	IKOSA Premium	1.0.0	1000	ANNUAL
1dd3fe6	IKOSA Premium	1.0.0	1000	ANNUAL
331d8d92ed85	IKOSA Premium	1.0.0	1000	ANNUAL
5ba6529	IKOSA Premium	1.0.0	1000	ANNUAL

Abbildung 16 Alle Abonnementpläne in React-Admin

Im Fenster "Subscription Plans" werden alle bestehenden Abonnementpläne angezeigt. Über die Suchleiste kann man die Abonnementpläne anhand ihrer ID oder ihres Namens durchsuchen. Rechts oben auf der Seite hat man die Möglichkeit, ebenso wie beim Fenster der Organisationen, die Daten zu exportieren. Durch Klicken auf "Create" kann ein neuer Abonnementplan erstellt werden.

Formular zur Erstellung eines neuen Abonnementplan



Name

Version

Net price

Subscription period

Display name

Restrictions

Key	Value

SAVE

Abbildung 17 Formular zur Erstellung eines neuen Abonnementplans in React-Admin

In Abbildung 17 wird das Formular für das Anlegen eines neuen "Subscription Plans" angezeigt. Zuerst werden der "Name", die "Version" und der "Net Price" eingegeben. Die "Subscription Period" kann mithilfe eines Dropdown-Menüs ausgewählt und der "Display Name" kann manuell eingegeben

werden. Bei den "Restrictions" kann der "Key" aus einem bereits befüllten Dropdown-Menü ausgewählt und der "Value" hinzugefügt werden. Mit dem blauen Plus können mehrere "Restrictions" angelegt werden.

Reflexion

Das vorliegend dokumentierte Masterprojekt kann als abgeschlossen betrachtet werden, da das MVP zur Nutzung vorliegt. Alle Anforderungen konnten erfolgreich in React-Admin umgesetzt werden.

Um das Projekt organisiert durchführen zu können, wurde auf gelernte Anwendungen des Studiums zurückgegriffen, welche den Ablauf erleichterten. Dementsprechend war es sehr sinnvoll und hilfreich einen Projektplan mit Meilensteinen zu erstellen, da dies bei der Umsetzung der Ziele geholfen hat. Die Meilensteine wurden vorab gemeinsam von KML Vision und dem Projektteam erstellt.

Während der Entwicklung stieß das Projektteam auf verschiedene Herausforderungen, insbesondere bei der Implementierung des Codes in React-Admin. Das Durchlaufen des Testbeispiels mit React-Admin war hierbei hilfreich für die Umsetzung, um sich mit dem neuen Tool zurechtzufinden. Die am Anfang des Projekts durchgeführte Erstellung der Anforderungen und Funktionen gemäß den Meilensteinen erwies sich als zeitaufwendig. Für den weiteren Projektverlauf galten sie aber als gute Richtlinie, an der man sich orientieren konnte.

Die Zusammenarbeit innerhalb des Teams war sowohl eine Herausforderung als auch eine Bereicherung. Während einige Aspekte der Entwicklung durch gemeinsame Arbeit vor Ort erleichtert wurden, erforderte die Natur des Projekts auch die Fähigkeit, unabhängig und verteilt zu arbeiten. Dies setzte klare Kommunikation und regelmäßige Abstimmungen voraus, um sicherzustellen, dass alle Teammitglieder auf dem gleichen Stand waren und effizient zusammenarbeiten konnten.

Im Generellen konnte das Projekt nach Zeitplan ablaufen. Während des gesamten Projekts haben laufend Meetings mit KML Vision stattgefunden. Dies unterstützte den gesamten Prozess, wodurch eine strukturierte Arbeitsweise möglich geworden ist. KML Vision hat immer schnell auf etwaige Anfragen reagiert und Hilfe angeboten. Da ebenso laufend klar war, wann das nächste Meeting stattfindet, war eine eindeutige Struktur möglich. Die Zusammenarbeit mit KML Vision kann demnach als sehr hilfreich und lehrreich angesehen werden.

Aus Sicht des Projektteams wurde das Masterprojekt erfolgreich abgeschlossen.

Referenzen

<https://www.kmlvision.com/our-offerings/ikosa-ai/>, eingesehen am 29.01.2024

<https://react.dev/>, eingesehen am 29.01.2024

<https://dev.api.ikosa.ai/api/v1/docs/#/>, eingesehen am 17.02.2024

Miro | The Visual Workspace for Innovation (<https://miro.com/index/>), eingesehen am 09.02.2024

Figma: The Collaborative Interface Design Tool (<https://www.figma.com/>), eingesehen am 17.02.2024

OpenAPI Specification v3.0.2 (<https://spec.openapis.org/oas/v3.0.2>), eingesehen am 17.02.2024

The OAuth 2.0 Authorization Framework (<https://www.rfc-editor.org/rfc/rfc6749>), eingesehen am 17.02.2024

*marmelab/react-admin: A frontend Framework for building data-driven applications
(<https://github.com/marmelab/react-admin>), eingesehen am 12.02.2024*

KML Vision: Your Partner In Microscopy Image Analysis (<https://www.kmlvision.com/>), eingesehen am 12.02.2024

<https://angular.io>, eingesehen am 20.11.2023

<https://vuejs.org/>, eingesehen am 20.11.2023

<https://svelte.dev>, eingesehen am 20.11.2023

<https://flutter.dev>, eingesehen am 20.11.2023

<https://vaadin.com>, eingesehen am 20.11.2023

<https://budibase.com>, eingesehen am 20.11.2023

<https://retool.com>, eingesehen am 20.11.2023

Abbildungsverzeichnis

Abbildung 1 Vorlage für User-Stories	2
Abbildung 2 User-Stories im Miro	3
Abbildung 3 Akzeptanzkriterien im Miro.....	3
Abbildung 4 Login-Screen in Figma	
Abbildung 5 Übersicht mit allen registrierten Organizations in Figma.....	4
Abbildung 6 Auflistung der Subscriptions für eine ausgewählte Organization in Figma.....	5
Abbildung 7 Übersicht über die Subscription Plan Details in Figma dargestellt	5
Abbildung 8 Erstellung einer neuer Subscription in Figma	6
Abbildung 9 Übersicht aller Subscription Plans in Figma dargestellt	6
Abbildung 10 Erstellung eines neuen Subscription Plans in Figma	7
Abbildung 11 Entscheidungstabelle Frontendframeworks (in Englisch aufgrund der Dokumentationsanforderungen für KML)	8
Abbildung 12 Login-Screen in React-Admin	10
Abbildung 13 Übersicht aller Organisationen in React-Admin	10
Abbildung 14 Abonnements einer ausgewählten Organisation in React-Admin	11
Abbildung 15 Formular "Create" in React-Admin.....	11
Abbildung 16 Alle Abonnementspläne in React-Admin	12
Abbildung 17 Formular zur Erstellung eines neuen Abonnementplans in React-Admin	12