Projekt SWMT - eVisitenkarte uxitra GmbH
Projektdokumentation

Jonathan Müller, Tobias Wahl, Yannick Schilling

20. Oktober 2025

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Diese Projektdokumentation dient als Referenz für die Konzeption, Entwicklung und das Vorgehen des Projekts eVisitenkarte für die uxitra GmbH, im Rahmen des Studienprojekts Softwaretechnik und Medieninformatik. Sie schafft ein klares Verständnis der Problemstellung, des Funktionsumfangs, der architektonischen Entscheidungen sowie der organisatorischen Aspekte. Das Softwaresystem eVisitenkarte ermöglicht die Erstellung, Verwaltung und Gestaltung dynamischer elektronischer Visitenkarten für E-Paper-Displays. Eine intuitive Benutzeroberfläche (UI) verantwortet die Nutzerprofile und Visitenkarteninhalte, welche formatiert per POST an das Display gesendet werden. Das System richtet sich an Personen und Organisationen, die ihre professionellen Kontakte digital und stets aktuell präsentieren möchten.

Die vorliegende Dokumentation ist in mehrere Kapitel unterteilt. Sie ist ein lebendiges Dokument und wird im Laufe des Projektfortschritts kontinuierlich aktualisiert und erweitert. Für den Aufbau und die Struktur der Dokumentation haben wir uns an verschiedenen Vorlagen und Beispielen orientiert.

2 Problemstellung und Zielgruppe

Dieses Kapitel beleuchtet die Ausgangssituation und definiert die primäre Nutzerbasis.

2.1 Detaillierte Problemstellung

Die eVisitenkarte wird entwickelt, um die aktuell bestehende Herausforderung in Form von häufig nötigem Aktualisieren der Visitenkarte zu lösen. Diese Situation führt derzeit zu Papierverschwendung, Papiermüll und dadurch eventuell auftretende Verwirrung. Die Kernaspekte des Problems umfassen:

- **Aspekt 1:** Hohe Kosten durch Papiermüll
- **Aspekt 2:** Hoher Zeitaufwand durch häufiges erneuern der Visitenkarte
- **Aspekt 3:** Verwirrung beim Kunden durch den eventuellen Besitz mehrerer Visitenkarten

Das Ziel ist es, durch die Implementierung der eVisitenkarte eine signifikante Verbesserung in diesen Bereichen zu erzielen.

2.2 Definition der Zielgruppe

Die eVisitenkarte ist primär für folgende Zielgruppen konzipiert:

- **Primäre Benutzer:** Unternehmen benötigen das System, um weniger Geld und Zeitaufwand zu verwenden um ihre Infos an Kunden und interessierte weiterzugeben.
- **Sekundäre Benutzer:** Auch selbstständige Businesspersonen können durch das System profitieren, da es viele sonst aufwendigen Aufgaben vereinfacht und automatisiert.

Für jede dieser Zielgruppen wurden spezifische Bedürfnisse und Erwartungen identifiziert, die im weiteren Verlauf der Anforderungsanalyse berücksichtigt werden.

3 Geforderter Funktionsumfang

Dieses Kapitel beschreibt den gefordeten Funktionsumfangs, abgeleitet aus den Anforderungen der definierten Zielgruppen und der Problemstellung, wie sie von der uxitra GmbH vorab kommuniziert wurde. Es gliedert sich in Hauptmodule und deren spezifische Funktionalitäten.

3.1 Funktionale Anforderungen

- Die Daten wie Vor- und Zuname des Benutzers und Job-Titel sollen aus einem Identity and Access Management (IAM) System kommen (z.B. Keycloak).
- Ein Admin soll verschiedene Layouts erstellen können aus denen ein User auswählen kann.
- Ein Layout soll aus verschiedenen Elementen bestehen können (Text, Bild, QR-Code).
- Eine Vorschau des Layouts für den Admin soll mit Musterdaten angezeigt werden (z.B. Dr.Ing. Max Mustermann, Senior Software Developer).
- Ein Benutzer soll ein Layout auswählen können, eine Vorschau mit seinen befüllten Daten sehen können und diese auf die Visitenkarte schreiben können.
- Text Elemente sollen dabei verschiedenen Datenfeldern aus dem IAM System zugeordnet werden können, z.B. Job-Titel, Vorname, Nachname, Titel.

Aktuell befinden sich noch weitere Features und Funktionen in der Ausarbeitungsphase, welche zu einem späteren Zeitpunkt final festgelegt werden.
Einige Ideen sind:

- QR-Code Generierung um vom ePaper Display digitale Visitenkarten bzw. Kontaktdaten direkt in einem Smartphone zu speichern (vCard).
- QR-Code Generierung um direkt auf die Website des Erstellers zu kommen

3.2 Hauptmodule und deren Kernfunktionen

Das System wird folgende Hauptmodule umfassen:

1. **Modul A: Benutzerverwaltung**

   Funktion A.1, Authentifizierung.

   Funktion A.2, Autorisierung von Zugriffen auf Backend und Daten

2. **Modul B: Datenmanagement**

   Funktion B.1, Daten-CRUD für Visitenkartendaten/Layouts über Services und Datenbank

   Funktion B.2, Bereitstellung von Visitenkartendaten für das ePaper-Display

   Funktion B.3, Speicherung von erstellten Visitenkarten

3. **Modul C: Generierung & Bereitstellung**

   Funktion C.1, Generierung von Visitenkarten auf Basis entsprechender Anfragen

   Funktion C.2, Bereitstellung der generierten Bilder für Frontend über Backend

   Funktion C.2, Umwandlung der Visitenkarten für das ePaper-Display

4. **Modul D: Hardware-Interaktionen**

   Funktion D.1, Schnitstelle zur Kommunikation mit ePaper-Display

   Funktion D.2, Senden der Visitenkarte an das ePaper-Display

3.3 Nicht-funktionale Anforderungen

Neben den Kernfunktionen ergeben sich vorab folgende nicht-funktionale Anforderungen:

- **Usability:** Die Benutzeroberfläche muss intuitiv bedienbar sein und eine Einarbeitungszeit von maximal 1h erfordern.
- **Wartbarkeit:** Die Codebasis muss klar strukturiert und gut dokumentiert sein, um eine einfache Wartung und Erweiterung zu ermöglichen.
- **Plattformunabghängigkeit:** Die Software soll unabhängig vom Betriebssystem des Nutzers lauffähig sein.

4 UI Entwürfe

Dieses Kapitel präsentiert Konzepte und erste Sketches der Benutzeroberfläche. Die Entwürfe dienen als visuelle Leitlinie für die Implementierung und sollen ein klares Bild der Benutzerinteraktion und des Layouts vermitteln. Sie basieren auf den identifizierten Anforderungen und Usability-Prinzipien. Die Anwendung ist primär als Desktop-Webanwendung für gängige Webbrowser konzipiert.

4.1 Ziel

Ziel der erstellten Sketches ist es, das geplante Layout und die grundlegende Benutzerführung der Anwendung konzeptionell zu visualisieren. Die Entwürfe dienen als Grundlage für spätere Prototypen und Implementierungen.

4.2 Übersicht

Die Anwendung gliedert sich in mehrere zentrale Ansichten, die über ein einheitliches Navigationsmenü erreichbar sind. Durch das klare und minimalistische Design, soll eine intuitive Benutzerführung gewährleistet werden. Alle Ansichten folgen demselben Aufbau, wodurch sich sowohl Endnutzer als auch Administratoren schnell zurechtfinden können.

4.3 Wireframes und Mockups ausgewählter Oberflächen

In diesem Abschnitt werden die einzelnen Ansichten vorgestellt und erläutert. Jeder Entwurf dient dazu, den geplanten Aufbau, die Navigation und die Funktionsweise der Benutzeroberfläche konzeptionell zu visualisieren. Ziel der Entwürfe war es, vor der eigentlichen Implementierung eine klares Verständnis für die Struktur und den Navigationsablauf der Benutzeroberfläche zu gewinnen.

4.3.1 Login-Seite

Die Login-Seite, wie in Abb. 1 zu sehen ist, bildet den Einstiegspunk der Anwendung. In der Entwurfsphase wurde ein Einfacher Anmeldebildschirm gestaltet, über den sich der Nutzer mit E-Mail und Passwort anmelden können. Zusätzlich wurde über ein Link nachgedacht, der bei Vergessen oder Verlust des eigenen Passwortes eine Verbindung zum Admin herstellt, um eine Zurücksetzung zu beantragen. Nach Abschluss der Entwurfsphase wurde entschieden, die Authentifizierung über das externe System Keycloak zu realisieren. Dadurch entfällt die manuelle Implementierung des Login-Prozesses. Der erstellte Sketch dient somit als Konzeptionelle Grundlage und half, das Erscheinungsbild und die Benutzerführung des später durch Keycloak bereitgestellten Login-Dialogs frühzeitig zu planen und zu visualisieren. Dennoch ist die Darstellung hier unbedingt erforderlich, da der Login der Einstiegspunkt und somit essenziell für das Verständnis der Abläufe ist.

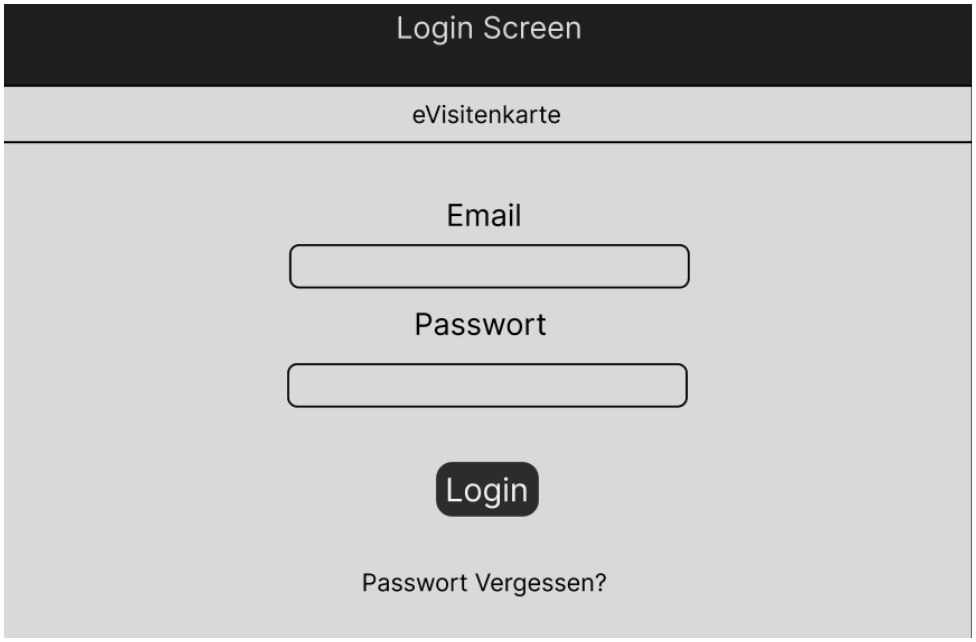

Abbildung 1: UI Entwurf: Login-Seite

4.3.2 Hauptseite

Die Hauptseite stellt die zentrale Übersicht der Anwendung dar und dient als Ausgangspunkt für alle weiteren Funktionen. Nach erfolgreicher Anmeldung gelangt der Nutzer standardmäßig auf diese Ansicht. Sie besteht aus einer klar gegliederten Navigation mit Reitern „Daten“ und „Layouts“, über die zwischen den Hauptfunktionen der Anwendung gewechselt werden kann, wie in Abb. 2 zu sehen ist. Im Benutzermodus werden ausschließlich persönliche Informationen angezeigt. Im Reiter „Daten“, der standardmäßig aufgerufen wird nach dem Einloggen, kann der Nutzer eigene Angaben wie Name, Titel, Adresse, Telefonnummer oder E-Mail-Adresse einsehen. Diese Informationen bilden die Grundlage für die spätere Erstellung der eVisitenkarte.

Administratoren verfügen über denselben Aufbau, können jedoch über einen Adminmodus-Schalter in der Kopfzeile, siehe Abb. 3, zwischen Benutzer- und Administratoransicht wechseln. Wird der Schalter aktiviert, erscheint im Navigationsbereich ein zusätzlicher Reiter mit „Nutzer“, der für reguläre Benutzer ausgeblendet bleibt. Die übrigen Funktionen, wie die Einsicht der eigenen Daten und Layouts bleiben auch im Adminmodus unverändert, wodurch die Bedienlogik konsistent bleibt.

Durch dieses Konzept wird eine rollenbasierte Benutzeroberfläche geschaffen, die mit einem einzigen Interface sowohl einfache Nutzerinteraktionen als auch administrative Aufgaben unterstützt. Der Wechsel zwischen den Modi erfolgt ohne Kontextwechsel, was die Übersichtlichkeit erhöht, und die Wartung vereinfacht.

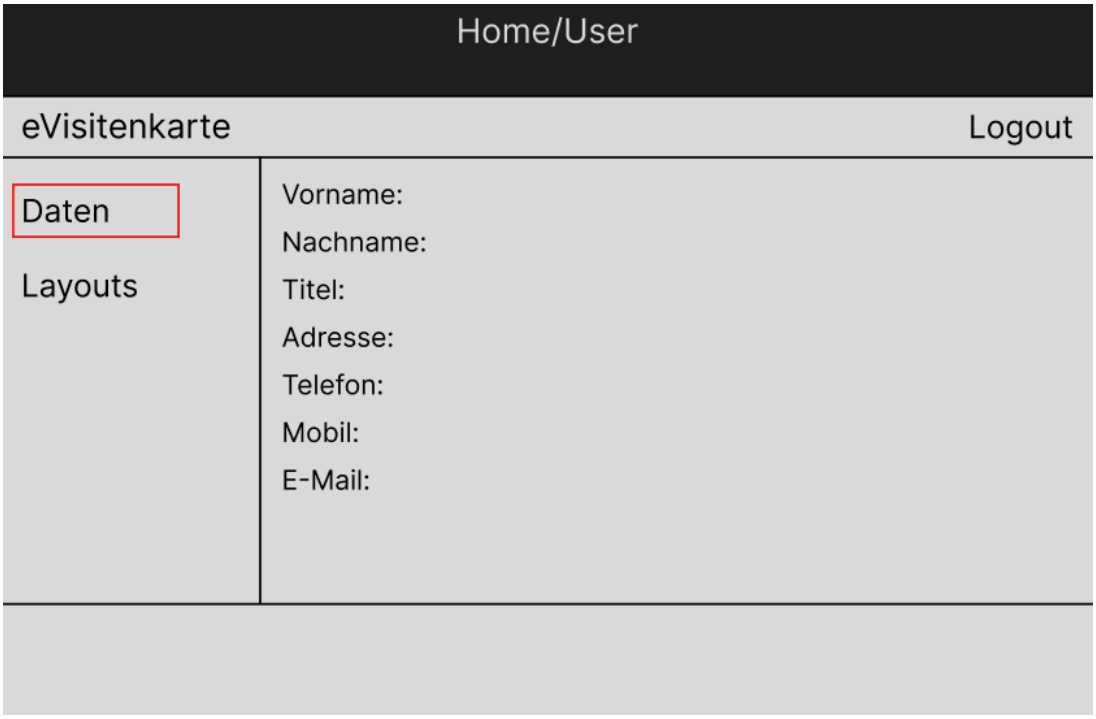

Abbildung 2: UI Entwurf: Hauptseite

4.3.3 Nutzerverwaltung

Die Nutzerverwaltung steht ausschließlich Administratoren zur Verfügung und dient der Verwaltung aller im System registrierten Benutzer. In diesem Bereich erhält der Administrator Zugriff auf eine Liste aller Registrierten Mitarbeiter und kann einzelne Datensätze auswählen, bearbeiten, hinzufügen oder löschen wie in Abb. 3 zu sehen ist. Wird ein Eintrag ausgewählt oder neu angelegt, öffnet sich die Bearbeitungsansicht, siehe Abb. 4, in der alle zugehörigen Felder editiert werden können. Diese Eingabemaske orientiert sich am Aufbau der Benutzeransicht und ermöglicht dem Administrator, dieselben Datenfelder wie ein regulärer Nutzer anzupassen – jedoch mit erweiterten Rechten, um auch fremde Konten zu ändern oder zu deaktivieren. Zum aktuellen Zeitpunkt ist noch nicht final geklärt ob eine Nutzerverwaltung implementiert werden muss oder ob, wie bereits in vorherigen Abschnitten erwähnt, die in Keycloak eingebettete Nutzerverwaltung genutzt wird. In diesem Fall würde die Nutzerverwaltung extern erfolgen und nicht mit in die UI integriert werden.

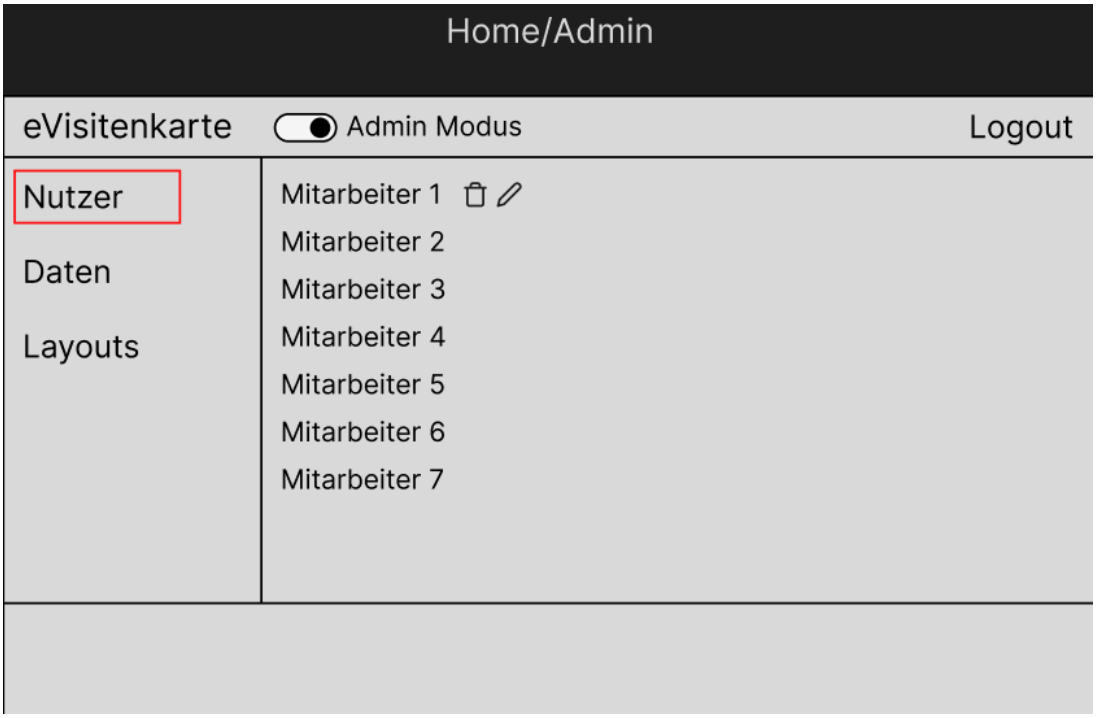

Abbildung 3: UI Entwurf: Nutzerverwaltung

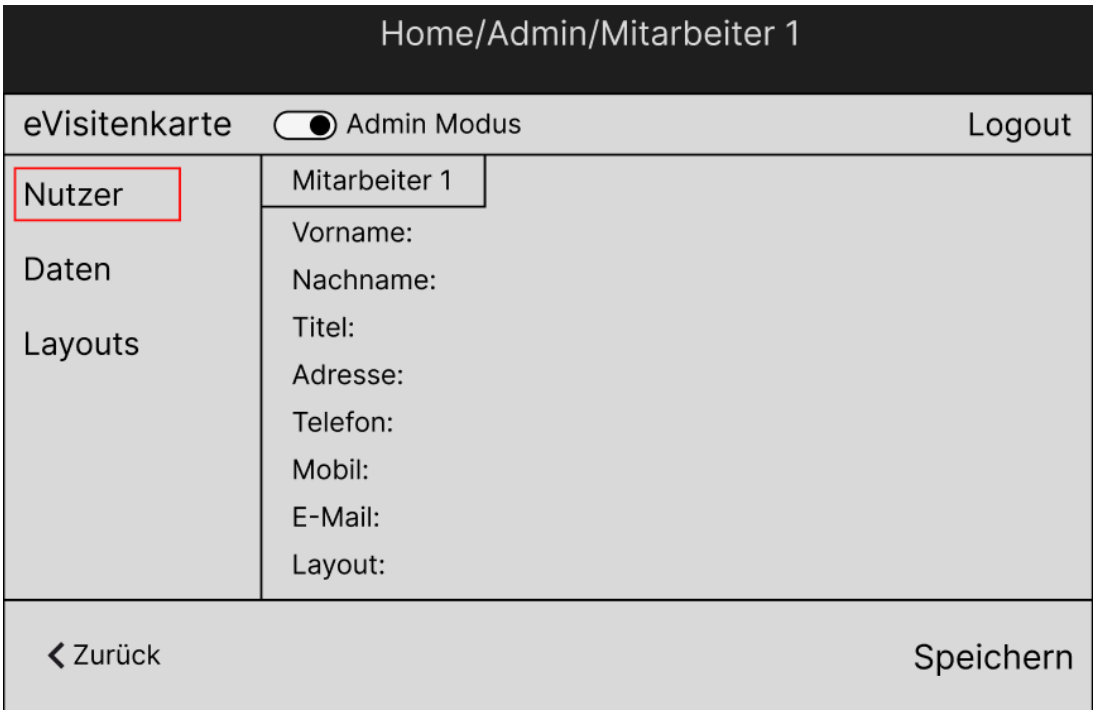

Abbildung 4: UI Entwurf: Nutzerbearbeitung

4.3.4 Layout Erstellung und Bearbeitung

Administratoren haben in der Anwendung die Möglichkeit, neue Layouts für die eVisitenkarten zu erstellen oder bestehende Vorlagen zu bearbeiten. Diese Funktion ist über den Reiter

„Layouts“ im Adminmodus zugänglich und erweitert die Layoutansicht der Benutzer um zusätzliche Bearbeitungsoptionen. Die Layout-Übersicht zeigt alle vorhandenen Designvorlagen in einer strukturierten Kachelansicht (siehe Abb. 5). Wird nun auf eines dieser Layouts gedrückt, so öffnet sich ein Popup das eine Vorschau des Layouts mit den eigenen Daten ermöglicht (siehe Abb. 6). Hier hat der Administrator die Möglichkeit das ausgewählte Layout zu bearbeiten oder es zu löschen. Wird ein Layout zur Bearbeitung geöffnet, wie in Abb. 7 zu sehen, wechselt die Anwendung in die Layout-Editor-Ansicht. Dort können zentrale Eigenschaften wie Farben, Schriftgrößen, Platzhalter für Textelemente oder die Positionierung von Logos angepasst werden. Der Editor ist so konzipiert, dass Änderungen unmittelbar in einer Vorschau sichtbar werden. Durch diesen Ansatz können Administratoren Layouts effizient an individuelle Anforderungen anpassen. Hierbei soll auch darauf geachtet werden, dass eingefügte Bilder oder Texte eine bestimmte Größe nicht übersteigen. Diese Begrenzungen dienen für eine klare Wiedergabe auf der eVisitenkarte. Der Bearbeitungsprozess ist bewusst einfach gehalten, um auch ohne tiefgehende Designkenntnisse nutzbar zu sein. Gleichzeitig stellt das konsistente Interface sicher, dass neue Layouts visuell mit bestehenden Vorlagen harmonieren. Änderungen werden erst nach expliziter Bestätigung gespeichert, um unbeabsichtigte Anpassungen zu vermeiden.

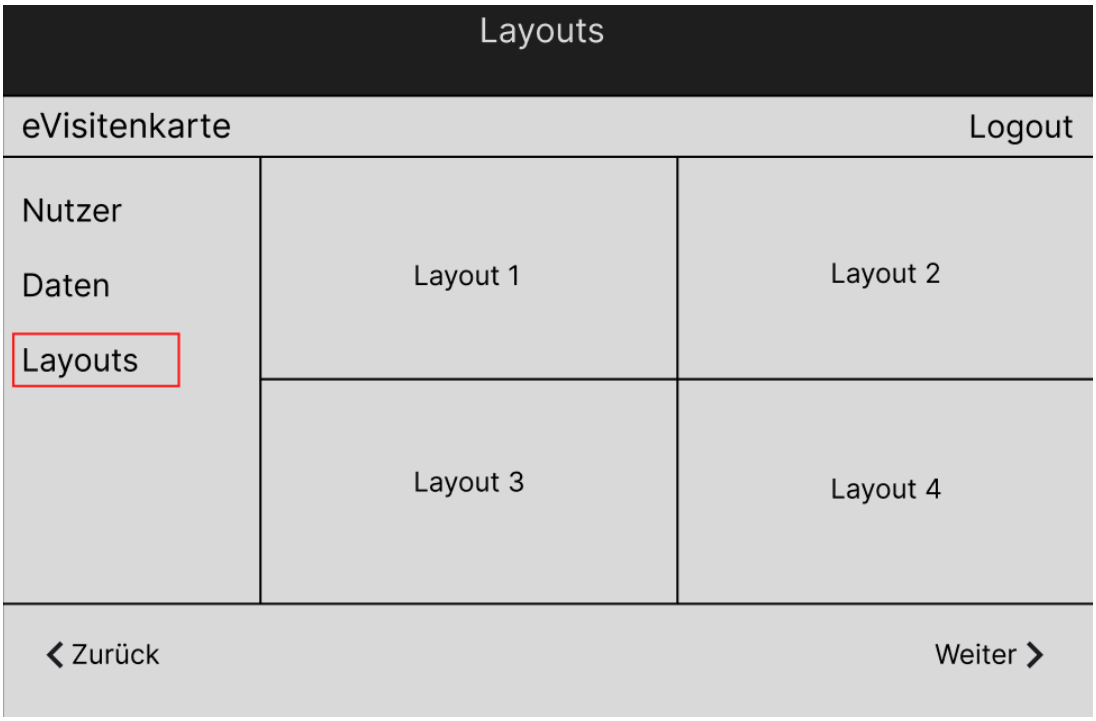

Abbildung 5: UI Entwurf: Layouts

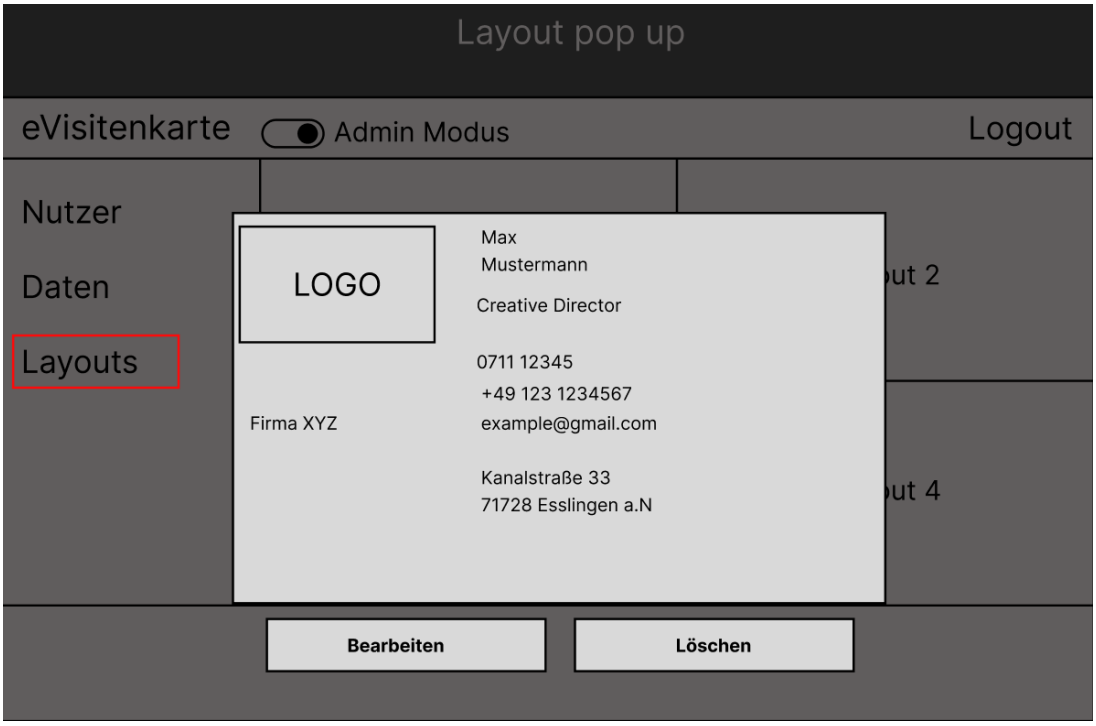

Abbildung 6: UI Entwurf: Layoutbearbeitung

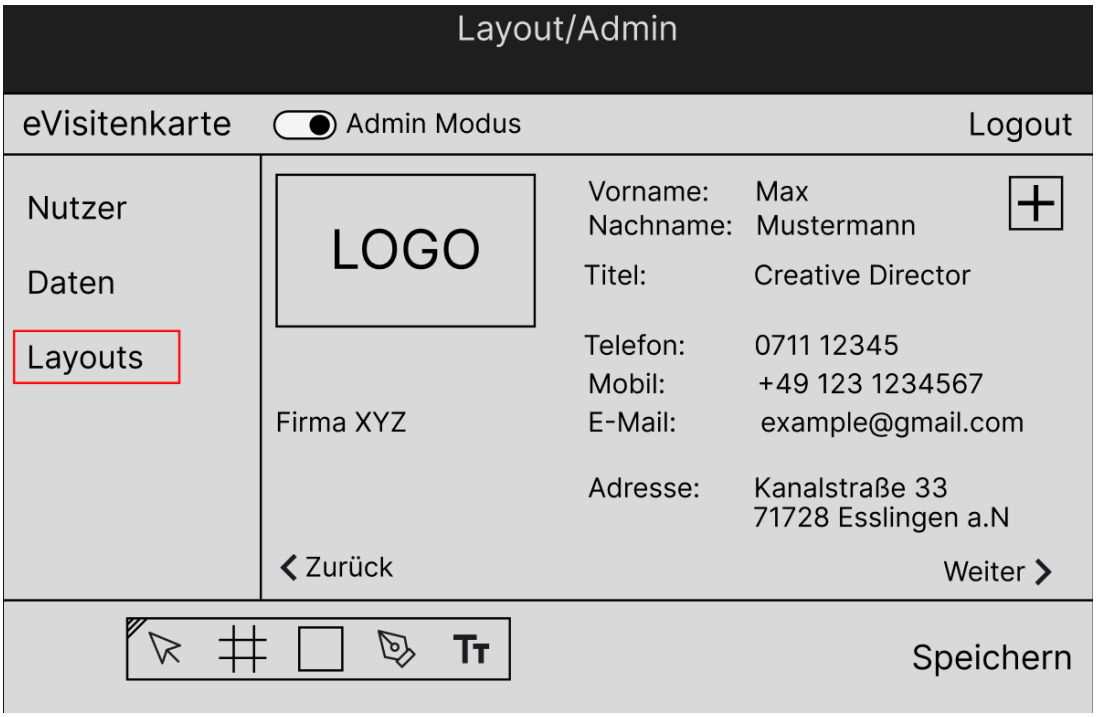

Abbildung 7: UI Entwurf: Layoutbearbeitung 2

4.3.5 Layout Auswahl durch Nutzer

Für reguläre Nutzer steht im Reiter „Layouts“ die Auswahl und Verwaltung der verfügbaren Visitenkarten-Designs im Vordergrund. Nach dem Login und Aufruf des Home-Screens kann

der Nutzer in dieser Ansicht aus verschiedenen, zuvor von Administratoren angelegten Layouts wählen (siehe Abb. 8). Die Layouts werden in einer übersichtlichen Kachel- oder Listenansicht dargestellt. Durch Anklicken eines Layouts öffnet sich eine Detailansicht, in der das gewählte Design in eVisitenkarten Darstellung und Dimensionen angezeigt wird. Dort kann der Nutzer, die zuvor unter „Daten“ einsehbaren Informationen, in dem Layout wiederfinden (siehe Abb. 9).

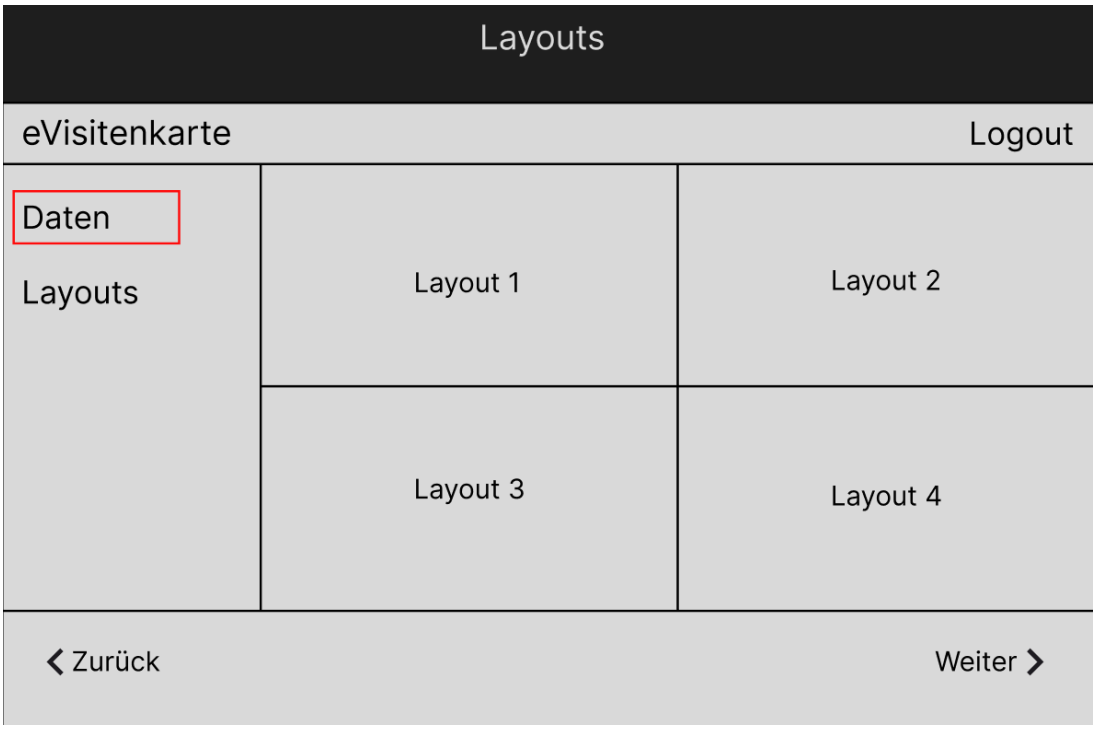

Abbildung 8: UI Entwurf: Layout-Asuwahl

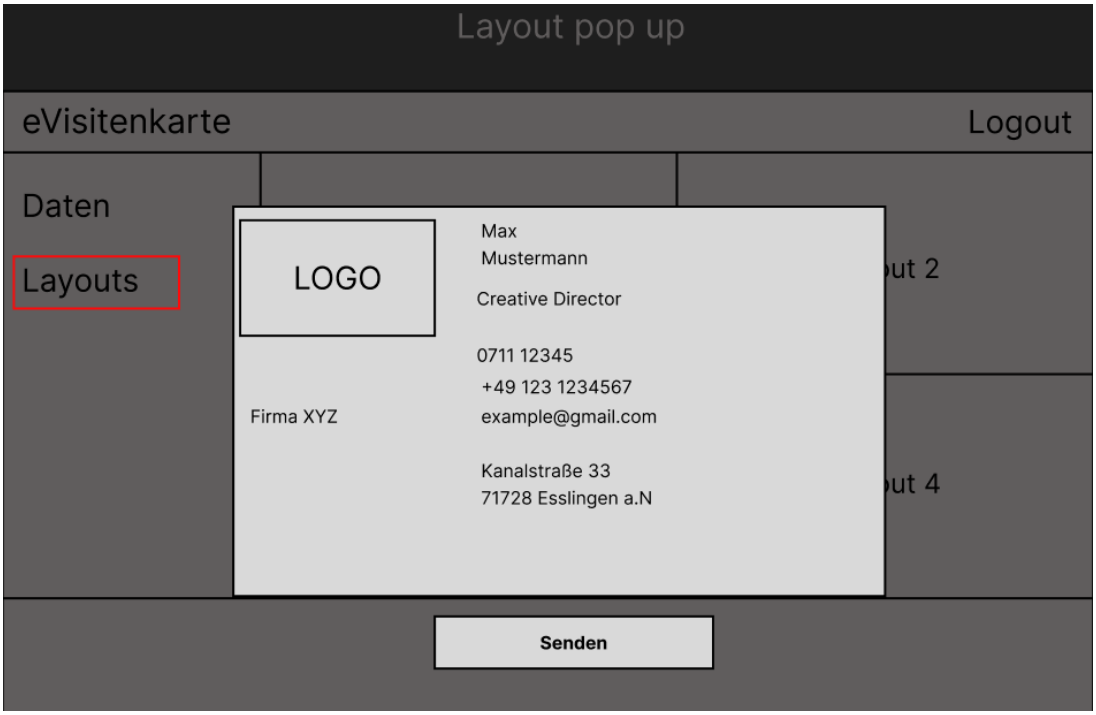

Abbildung 9: UI Entwurf: Layout senden

4.4 Zusammenfassung und Ausblick

Im nächsten Schritt soll auf Grundlage der erstellten Wireframes ein interaktiver Prototyp mit einem finalen Farbschema und detaillierteren grafischen Elementen entwickelt werden. Ziel ist es, die Benutzerführung und das Design realitätsnäher darzustellen. Zusätzlich ist die Erstellung eines Clickdummys geplant, um die Navigation zwischen den einzelnen Seiten zu simulieren und erste Usability-Tests durchführen zu können.

5 Architektur (konzeptionell und technologisch)

Dieses Kapitel soll einen Überblick zur Architektur geben, wobei sowohl konzeptionelle Modelle als auch die zugrunde liegenden technologischen Entscheidungen detailliert werden. Aus den Gegebenheiten, Gesprächen mit Herrn Michael Watzko und Herrn Alexander Pärsch, sowie der Bereitstellung des Raspberry-Pi Prototyps durch die uxitra GmbH, erschließt sich das folgende Konzept und ist das für uns Naheliegenste. Ein weiterer äußerst wichtiger Aspekt für uns bei der Entscheidung war, der zum Einen sehr begrenzte Zeitraum als auch die geringe Manpower. Auf diesem Hintergrund haben wir uns für die im folgenden beschriebnen Prinzipien und Designziele entschieden.

5.1 Architekturprinzipien und Designziele

Die Architekturgestaltung basiert auf folgenden Leitprinzipien:

- **Modulare Bauweise:** Reduzierung der Komplexität durch Zerlegung in unabhängige, wiederverwendbare Komponenten und zur Parallelisierung der Entwicklung, sowohl in Bezug auf Implementierung als auch Konzeption, Design und Planung.
- **Schichtenarchitektur:** Klare Trennung von Belangen zur Förderung von Wartbarkeit, Erweiterbarkeit und Nachvollziehbarkeit.
- **Plattformunabhängigkeit:** Die Anwednung kann auf die verschiedenen Betriebssystemen laufen, was die Wahl der Infrastruktur, Wartbarkeit, Migration und ggf. Weiterentwicklung deutlich erleichtert.

5.2 Konzeptionelle Architektur

Die konzeptionelle Architektur beschreibt das System aus einer einfacheren Perspektive und identifiziert die Hauptkomponenten und deren Interaktionen auf einer hohen Abstraktionsebene. Diese soll ein Grundverständnis für den Aufbau und Umfang schaffen, sowie die zugrundeliegenden Prinzipien veranschaulichen.

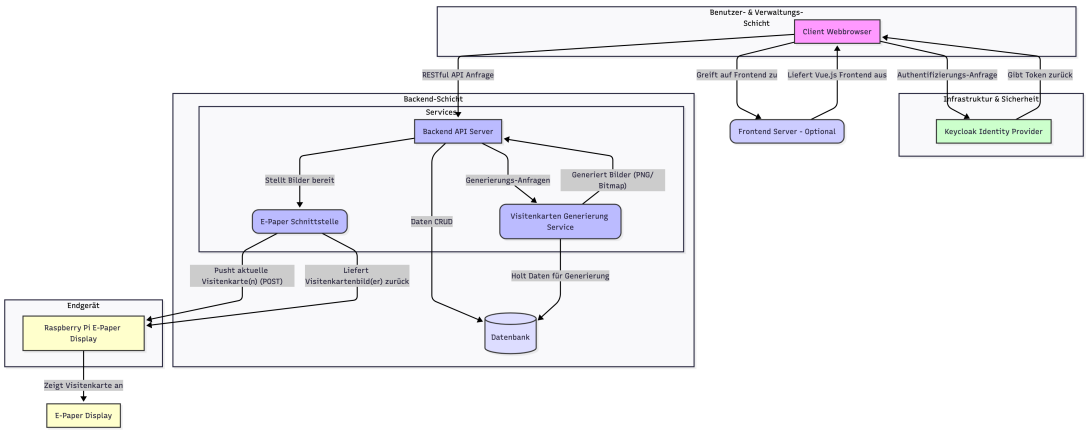

Abbildung 10: Konzeptionelle Ansicht des Gesamtsystems

Das System ist in drei Hauptschichten unterteilt: die Benutzer-/Verwaltungs-Schicht (Präsentation), die Backend-Schicht (Applikation und Domäne) und die Endgerät-Schicht (Infrastruktur).

5.2.1 Benutzer-/Verwaltungs-Schicht (Präsentation)

Diese Schicht ist für die Interaktion mit dem Endbenutzer verantwortlich.

- **Client Webbrowser**: Der zentrale Bestandteil dieser Schicht. Er initiiert `RESTful API` Anfragen, greift auf das Frontend zu, stellt Authentifizierungsanfragen und empfängt `OAuth` Token.
- **Frontend Server (Optional)**: Liefert `JavaScript` basierte Frontends an den *Client Webbrowser* aus.
- **Keycloak Identity Provider**: Verantwortlich für Authentifizierung und Sicherheit. Er verarbeitet Authentifizierungsanfragen und gibt Token zurück.

5.2.2 Backend-Schicht (Applikation & Domäne)

Diese Schicht implementiert die Kernlogik des Systems.

- **Backend API Server**: Fungiert als zentrale Schnittstelle im Backend. Er empfängt `RESTful API` Anfragen, interagiert mit der *E-Paper Schnittstelle* und dem *Visitenkarten Generierung Service*. Er stellt zudem generierte Bilder bereit und verarbeitet Generierungsanfragen.
- **E-Paper Schnittstelle**: Verantwortlich für die Bereitstellung von Bildern, das Pushen aktueller Visitenkarten (`POST`) und das Zurückliefern von Visitenkartenbild(ern).
- **Visitenkarten Generierung Service**: Generiert Visitenkarten als Bilder (`PNG`/`Bitmap`) basierend auf Generierungsanfragen und verwaltet die für die Generierung notwendigen Daten.

5.2.3 Endgerät-Schicht (Infrastruktur)

Diese Schicht repräsentiert das physische Ausgabegerät.

- **Raspberry Pi E-Paper Display**: Empfängt Visitenkartenbilder von der *E-Paper Schnittstelle* und zeigt diese auf dem E-Paper Display an.

5.3 Interaktionen

Die Komponenten interagieren wie folgt:

- Der *Client Webbrowser* sendet Anfragen an den *Backend API Server* und kommuniziert optional mit dem *Frontend Server*.
- Authentifizierungsanfragen werden vom *Client Webbrowser* an den *Keycloak Identity Provider* gerichtet.
- Der *Backend API Server* leitet Bildgenerierungsanfragen an den *Visitenkarten Generierung Service* weiter und empfängt die generierten Bilder.
- Für die Verwaltung der E-Paper Displays interagiert der *Backend API Server* mit der *E-Paper Schnittstelle*, welche die Bilder an das *Raspberry Pi E-Paper Display* übermittelt.
- Eine **Datenbank** dient als persistenter Speicher für den *Backend API Server* (für `CRUD`-Operationen) und für den *Visitenkarten Generierung Service* (für generierungsrelevante Daten).
- Das *Raspberry Pi E-Paper Display* ist das Endgerät, das die finalen Visitenkarten darstellt.

5.4 Technologien

Die Technologien die zur Umsetzung der konzeptionellen Architektur verwendet werden. Hier ist anzumerken, dass zum aktuellen Zeitpunkt einige Details noch nicht bzw. nicht ausreichend überblickt werden können und es vor Beginn der Implementierung der ersten Features noch zu Änderungen kommen kann. Aktuell führen wir kleinere Tests auf lokaler Ebene durch, um uns mit den angeführten Technologien vertraut zu machen.

Es sind folgende Technologien zur Umsetzung der oben beschriebenen Komponenten angedacht:

- **Frontend:** Implementiert mit *vue.js*.
- **Backend-Services:** Entwickelt mit *Node.js*.
- **Datenbank:** *PostgreSQL* als primäre relationale Datenbank.
- **Authentifizierung:** *Keycloak* mit eigener PostgresSQL Datenbank für die Nutzerverwaltung

Zusätzlich ist Containerisierung mit *Docker* geplant.

6 Projektmanagement

Dieses Kapitel beschreibt die Methodik und die organisatorischen Aspekte, die für die Planung, Durchführung, Überwachung und Steuerung des Projekts zum Einsatz kommen.

6.1 Projektmethodik

Aus dem Projektrahmen ging bereits hervor, dass wir im Team agil arbeiten sollen. Zudem entschieden wir uns für Scrum, da sich der Projektaufbau mit Meilensteinen gut für Sprints eignet. Da zu Projektbeginn auch nicht alle Requirements bekannt sind und die Aufwandschätzung nur ungenau erfolgt, ist eine iterative und inkrementelle Arbeitsweise sinnvoll.

- **Sprints:** Die Entwicklung erfolgt in wöchentlichen Sprints.
- **Rollen:** Folgende Rollen sind definiert: Projektleiter, Product Owner, Scrum Master, Entwicklungsteam.

6.2 Meetings

In einem Sprint führen wir zuerst ein Sprint Planning durch, bei welchem wir Aufgaben verteilen. Diese werden in einem Weekly dann erneut abgeglichen, bei Fehleinschätzung des Aufwands eventuell nochmal umverteilt und letztendlich fertig bearbeitet. Zuletzt führen wir nach jedem Sprint eine kurze Retrospektive durch, welche dann in das nächste Sprint Planning übergeht. Abseits des Sprintkreises halten wir jede Woche (bei Bedarf) ein Treffen mit unserem Betreuer Michael Watzko ab, mit dem wir eventuelle Fragen abklären. Wir verzichten somit auf Daily Meetings, da für sie der Aufgabenumfang zu gering ist und wir sie im Rahmen eines Studienprojekts zeitlich für nicht angemessen erachten.

6.3 Teamorganisation und Verantwortlichkeiten

Das Projektteam besteht aus 3 Mitgliedern mit folgenden Hauptverantwortlichkeiten:

- **Projektleiter:** Tobias Wahl, verantwortlich für die Gesamtkoordination und Stakeholder-Kommunikation.
- **Product Owner:** Yannick Schilling, verantwortlich für die Produktvision und die Priorisierung der Anforderungen.
- **Scrum Master:** Jonathan Müller, verantwortlich für die Einhaltung des Scrum-Prozesses und die Beseitigung von Hindernissen.
- **Entwicklungsteam:** Bestehend aus 3 Entwicklern, verantwortlich für die Implementierung der Funktionalitäten.

Die Rollenverteilung wurde im Team besprochen und entspricht den Stärken die wir zu Beginn des Projektes mitbringen. Zwar sind wir primär alle Entwickler, eine grobe Aufgabenverteilung hilft jedoch uns bei eventuell auftretenden Problemen schneller zu Orientieren.

6.4 Tools und Prozesse

Für das Projektmanagement und die Entwicklung werden folgende Tools und Prozesse eingesetzt:

- **Anforderungsmanagement:** *Jira* zur Erfassung und Verwaltung von User Stories, Requirements und Tasks.

- **Versionskontrolle:** *Git* (mit *GitHub*) für die Quellcodeverwaltung.
- **Kommunikation:** *Discord* für die interne Teamkommunikation, *WebEx* und Email zur Kommunikation mit Stakeholdern.

Mit den von uns gewählten Tools sind wir teils schon aus vergangenen Projekten vertraut, so fällt uns ihre Benutzung leichter. Die Funktionalitäten sind für unsere Anforderungen mehr als ausreichend.

6.5 Jira

Auf Jira wollen wir die einzelnen Aufgaben in Sprints einteilen und sie über das Board als „To-Do“, „In Progress“, „in Review“ (für Code) oder „Done“ kennzeichnen. Die Aufgaben im aktuellen Sprint werden einer oder mehreren Personen zugeteilt und es wird eine Gewichtung vorgenommen. So ist schnell einsehbar, wer woran arbeitet bzw. arbeiten sollte.

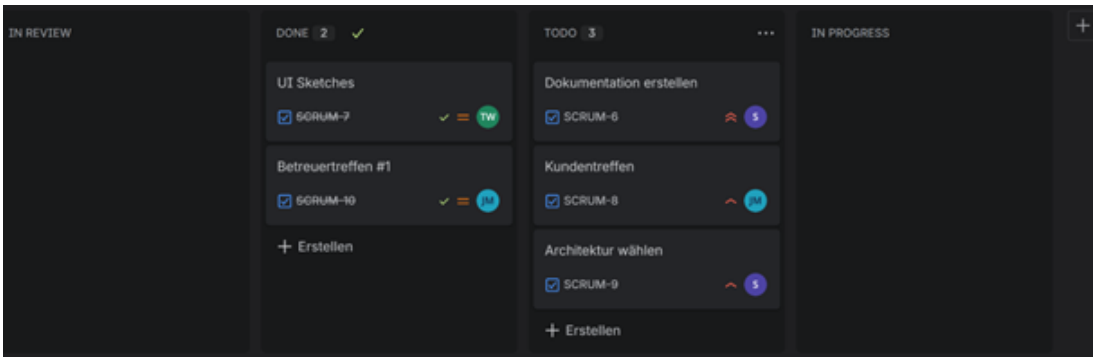

Abbildung 11: Jira Board

Zudem werden alle uns vorgegebenen Meilensteine auf Jira eingetragen und mit den uns bekannten Aufgaben versehen. So können wir schon vor Erstellung des aktuellen/nächsten Sprints abschätzen, wie viel Aufwand vor uns liegt und stellen sicher, dass zukünftige Augaben nicht vergessen werden.

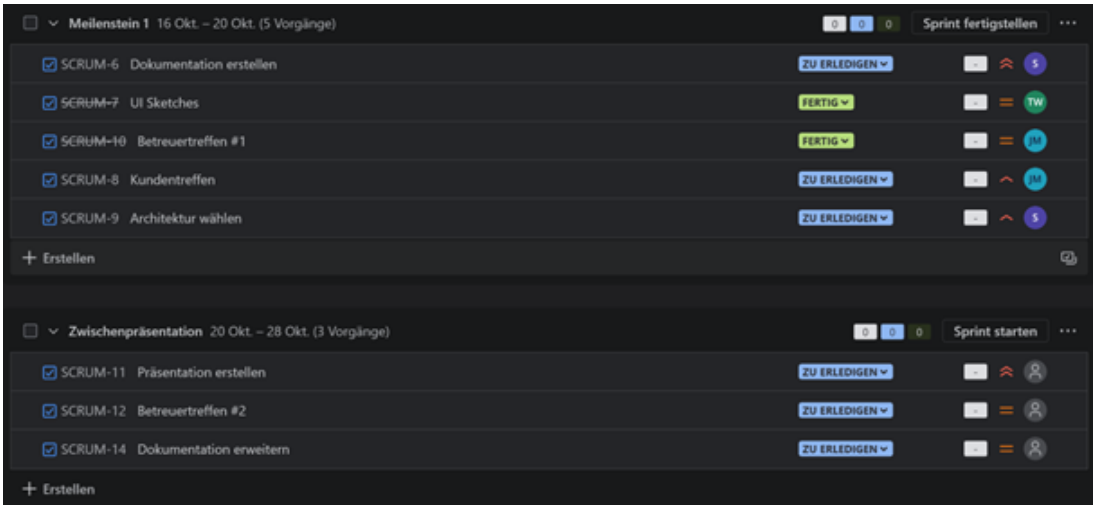

Abbildung 12: Meilensteine

Jede in Jira eingetragene Aufgabe versehen wir mit Acceptance Criteria, welche vor Fertigstellung der Aufgabe abgehakt werden müssen. Dies geschieht um die Qualität der Features zu garantieren und uns vor dem Vergessen eventueller Unterpunkte zu schützen. Ein Feautre gilt als abgeschlossen, sobald alle Acceptance Criteria erfüllt sind.

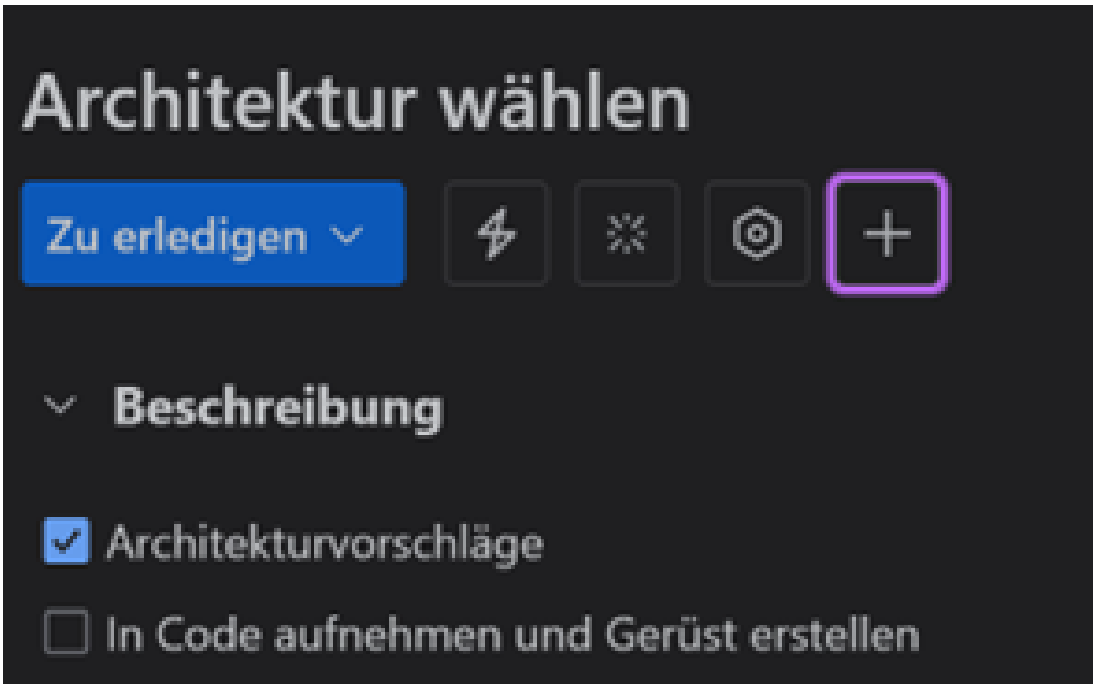

Abbildung 13: Acceptance Criteria

Zusätzlich ist noch geplant die Traceability (Nachvollziehbarkeit) im Entwicklungsprozess zu verbessern, indem eine Verknüpfung zwischen GitHub (Versionskontrolle) und Jira (Ticket-System) hergestellt wird. Konkret wurde vorgeschlagen, Jira-Tickets direkt über Commit- Nachrichten in GitHub zu schließen, um die Nachverfolgung von Änderungen und den Fortschritt von Aufgaben transparent zu machen.

7 Aufwandsschätzung

Dieses Kapitel präsentiert die detaillierte Aufwandsschätzung für die Entwicklung. Die Schätzung basiert auf der Analyse des geforderten Funktionsumfangs und berücksichtigt die Komplexität, Herausforderungen und Risiken.

7.1 Schätzmethodik

Die Aufwandsschätzung wurde mittels der *Planning Poker*-Methode durchgeführt. Wir haben uns aufgrund des begrenzten Zeitfensters entschieden in Stunden zu schätzen. Sie bewegen sich auch im ungefähren Rahmen des bereits vorgegebenen Zeitaufwands.

7.2 Detaillierte Schätzung pro Modul/Funktionalität

Die folgende Tabelle gibt einen Überblick über die geschätzten Aufwände.

Tabelle 1: Aufwandsschätzung pro Modul

Modul/Funktionalität	**Geschätzter Aufwand**	**Kommentar**
Benutzerverwaltung	70h	KeyCloak Einbindung, Profilverwaltung.
Datenmanagement	70h	Datenbank, KeyCloak, Schnittstellen.
UI Entwicklung (gesamt)	160h	Basierend auf den Mockups und Usability-Anforderungen.
Berichterstattung	90h	Erstellung von Dokumentation.
Architektur/Infrastruktur	30h	Initiales Setup.
Testing (gesamt)	40h	Unit-, Integrations- und Akzeptanztests.
Projektmanagement	130h	Overhead für Meetings, Koordination, Seminare.
Gesamtaufwand (geschätzt)	**590h** (ohne Puffer)	

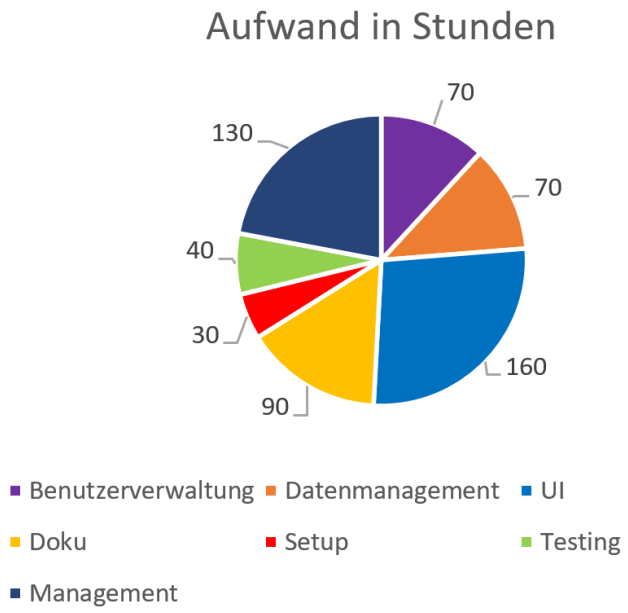

Abbildung 14: Stundenverteilung

7.3 Risiken und Puffer

Bei der Aufwandsschätzung wurden folgende potenzielle Risiken berücksichtigt und zusätzlich ein entsprechender Puffer eingeplant:

- **Integration mit externen Systemen:** Unvorhergesehene Komplexitäten bei der Einbindung des SSO-Systems Keycloak. 10%
- **Anforderungsänderungen:** Agiles Vorgehen reduziert dieses Risiko, aber ein kleiner Puffer für unvorhergesehene Änderungen ist eingeplant, da die Erfahrung mit den gneutzen Technologien begrenzt ist. 5%
- **Einarbeitung:** Ein gewisser Puffer für Einarbeitungszeiten in neue Technologien. 10%

8 Gesprächsprotokoll Kundentermin 1

- **Datum:** 16. Oktober 2025
- **Uhrzeit:** 16:00 Uhr
- **Ort:** WebEX

Anwesende:

- **Entwicklungsteam:**
  - Jonathan Müller
  - Tobias Wahl
  - Yannick Schilling
- **Betreuer:**
  - Michael Watzko (uxitra GmbH)
- **Kunde:**
  - Alexander Pärsch (uxitra GmbH)

Thema des Gesprächs:

Vorstellung der geplanten Inhalte des Projekts eVisitenkarte der uxitra GmbH im Rahmen des Studienprojekts SWMT.

Ablauf des Gesprächs:

Nach einer kurzen Vorstellungsrunde stellte das Entwicklungsteam Herrn Pärsch die geplante Agenda und im Anschluss die ersten Inhalte des Projekts mittels einer Präsentation vor. Herr Pärsch hatte jederzeit die Möglichkeit, die Präsentation für Anmerkungen, Fragen und Feedback zu unterbrechen.

Vorgestellte Inhalte und Anmerkungen des Kunden:

1. Vorstellungsrunde und Agenda

- **Vorgestellte Inhalte:**
  - Kurze gegenseitige Vorstellung der Anwesenden.
  - Aufzeigen der heutigen Agenda:
    - Umfang und Anforderungen: User Stories, funktionale Anforderungen
    - Visualisierung: UI-Sketches
    - Prozess: Tools, Kommunikation
    - Zeitplan und Meilensteine: Grobe Zeitplanung
- **Anmerkungen/Fragen/Feedback des Kunden:**
  - Keine spezifischen Anmerkungen zur Agenda.

2. Umfang und Anforderungen: User Stories, funktionale Anforderungen

- **Vorgestellte Inhalte :**
  - User Stories
  - Funktionale Anforderungen
- **Anmerkungen/Fragen/Feedback des Kunden:**
  - **Datenschutz (DSGVO):** Es wurde angeregt, die Aspekte der DSGVO bei der Verarbeitung und Speicherung von User-Daten zu berücksichtigen.
  - **Userverwaltung:** Der Kunde schlug vor, die Userverwaltung möglicherweise direkt in **KeyCloak** zu implementieren/integrieren, um vorhandene Lösungen zu nutzen und den Aufwand zu minimieren.
  - **Prüfung von User-Daten:** Die Möglichkeit eines Background-Tasks zur Prüfung von User-Daten (z.B. auf Aktualität oder Vollständigkeit) wurde als Idee eingebracht.
  - **QR-Code für vCard:** Der Kunde erwähnte die Idee eines QR-Codes, der direkt eine vCard-Datei generiert oder beinhaltet, um den Austausch von Kontaktdaten zu erleichtern.
  - **Tipps für die Zwischenpräsentation:** Herr Pärsch gab den Hinweis, dass bei der nächsten Präsentation der Fokus vor allem auf die **technische Umsetzbarkeit** gelegt werden sollte und man sich nicht zu viel versprechen sollte, was eventuell technisch nicht realisierbar ist.

3. Visualisierung: UI-Sketches

- **Vorgestellte Inhalte:**
- UI-Sketches
- Ideen für Designs und deren Umsetzung
- **Anmerkungen/Fragen/Feedback des Kunden:**
  - **Begrenzte Feldgrößen:** Herr Pärsch schlug vor, die Begrenzung von Feldgrößen (z.B. Textfelder für Namen, Titel) in den UI-Sketches zu berücksichtigen, um Probleme mit zu langen Eingaben zu vermeiden.
  - **Impersonierung (Admin als User):** Es wurde angeregt, die Möglichkeit zu schaffen, dass ein Administrator sich als normaler User ausgeben (impersonieren) kann, um Tests durchzuführen oder Support zu leisten.
  - **Help-/Support-Button:** Die Integration eines klar sichtbaren Help- oder Support-Buttons im UI wurde vorgeschlagen, um Usern bei Fragen oder Problemen schnell Unterstützung zu bieten.
  - **Systemkonzeption (Desktop vs. Mobile):** Eine wichtige Frage des Kunden war, für welches Endgerät das System primär konzipiert ist (Desktop oder Mobile), da dies Auswirkungen auf das UI-Design und die Usability hat.

4. Prozess: Tools, Kommunikation

- **Vorgestellte Inhalte:**
- Projektmanagement: Jira
- Versionskontrolle: github
- Kommunikationsmittel
- **Anmerkungen/Fragen/Feedback des Kunden:**
  - **Traceability mit Jira:** Der Kunde gab den Tipp, die Traceability (Nachvollziehbarkeit) im Entwicklungsprozess zu verbessern, indem eine Verknüpfung zwischen GitHub (Versionskontrolle) und Jira (Ticket-System) hergestellt wird. Konkret wurde vorgeschlagen, Jira-Tickets direkt über Commit-Nachrichten in GitHub zu schließen, um die Nachverfolgung von Änderungen und den Fortschritt von Aufgaben transparent zu machen.

5. Zeitplan und Meilensteine: Grobe Zeitplanung

- **Vorgestellte Inhalte:**
- Vorbestimmte Meilenstein des Projekts
- **Anmerkungen/Fragen/Feedback des Kunden:**
  - **Visuelle Timeline:** Der Kunde schlug vor, für die Ziwschenpräsentation eine visuelle Zeitleiste (Timeline) zu erstellen. Diese sollte übersichtlich darstellen, was bisher im Projekt geschehen ist und welche Meilensteine und Schritte für die Zukunft geplant sind.

Offene Punkte/Nächste Schritte:

- Das Entwicklungsteam prüft die technische Umsetzbarkeit der angesprochenen Punkte, insbesondere die Integration von KeyCloak und die DSGVO-Konformität.
- Das Team wird die Anmerkungen zu den User Stories, funktionalen Anforderungen, UI-Sketches (Feldgrößen, Impersonierung, Help-Button, Desktop/Mobile-Konzeption) und dem Prozess (Traceability Jira/GitHub) in die weitere Planung und Gestaltung einbeziehen.
- Für die nächste Präsentation: Fokus auf technische Machbarkeit und realistische Versprechen legen. Klärung der primären Plattform (Desktop/Mobile) für das System. Prüfung der Implementierung der Jira/GitHub-Verknüpfung. **Erstellung einer visuellen Timeline, die den bisherigen Fortschritt und die zukünftigen Meilensteine darstellt.**

Fazit des Gesprächs:

Das Gespräch bot einen guten ersten Einblick in die geplanten Inhalte des Projekts und ermöglichte dem Kunden, frühzeitig wertvolles Feedback und wichtige Anregungen zu geben, die in die weitere Entwicklung einfließen werden.

Protokolliert von:

Yannick Schilling