Prototyping Projektdokumentation

Name: Endrit Kukalaj

E-Mail: kukalend@students.zhaw.ch

URL der deployten Anwendung: <https://swissdentallineprototyping.netlify.app/>

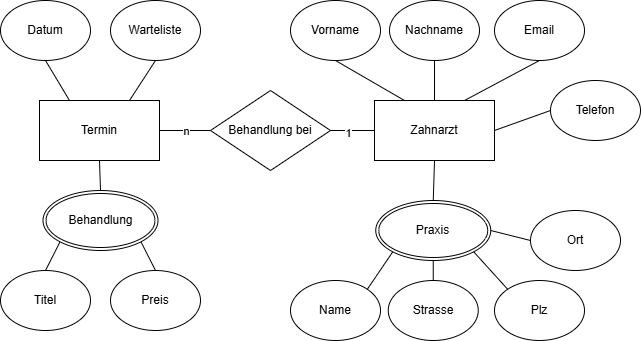
GitHub Repository: <https://github.com/EndritKukalaj/Prototyping_Abschlussprojekt>

# Einleitung

SwissDentalLine ist eine webbasierte Plattform zur digitalen Terminverwaltung im zahnärztlichen Umfeld. Die Anwendung ermöglicht es Patienten Behandlungen online zu buchen, die von Zahnarztpraxen eingestellt wurden. Dabei lag der Fokus insbesondere auf einer benutzerfreundlichen Oberfläche, einer unkomplizierten Terminwahl und auch der klaren Darstellung relevanter Behandlungsinformationen. Ziel des Abschlussprojektes war es, zentrale Funktionalitäten als Webanwendung mit SvelteKit und MongoDB umzusetzen.

Das Abschlussprojekt baut auf der SwissDentalLine-Projektidee aus der Prototyping-Gruppenarbeit auf. Im Rahmen dieser Einzelarbeit wurden bewusst nur die Kernfunktionalitäten der Patientenansicht realisiert, um den Projektumfang überschaubar zu halten. Der Fokus liegt auf der Terminbuchung und -verwaltung. Komplexere Features wie Flex-Termine oder die Zahnarzt-Ansicht wurden dabei ausgeklammert, um den Umfang der Arbeit nicht zu überschreiten. Das Datenset wurde ebenfalls vereinfacht, um eine praktische Umsetzung zu gewährleisten.

# Datenmodell

**

Die Applikation verwendet folgende MongoDB-Collections:

**Termine**: Speichert alle buchbaren Termine inkl. Datum, Behandlung, Preis und Wartelistenstatus

**Zahnärzte**: Enthält Stammdaten zu Zahnärzten und Praxisinformationen

Die Datenstruktur wurde gezielt vereinfacht, indem komplexere Attribute wie Behandlung und Praxis direkt in die jeweiligen Dokumente integriert wurden. Durch diese Denormalisierung konnte auf komplexe Joins oder Aggregationsabfragen verzichtet und stattdessen eine einfachere Datenabfrage im Frontend ermöglicht werden. In einem produktiven System wäre es jedoch empfehlenswert, Entitäten wie Behandlungen oder Praxen in eigene Collections auszulagern, um Redundanz zu vermeiden und die Datenpflege zu erleichtern.

**JSON Datenstruktur der Termine:**

{

    "\_id": "c155b900-93cd-419f-8b20-7e9a3ca657f2",

    "datum": "2025-03-25T14:45:00.000Z",

    "warteliste": true,

    "zahnarztId": "1581318d-8d80-4d97-ab50-649126f6fa8f",

    "behandlung": {

      "titel": "Dentalhygiene",

      "preis": 150

    }

  }

**JSON Datenstruktur der Zahnärzte:**

{

    "\_id": "5cec920e-d651-4ac5-ab5f-2a323720e383",

    "vorname": "Agata",

    "nachname": "Stutz",

    "email": "agata.stutz@zahnklinik-bellevue.ch",

    "telefon": "044 500 60 10",

    "praxis": {

      "name": "Zahnklinik Bellevue",

      "strasse": "Seestrasse 5",

      "plz": "8000",

      "ort": "Zürich"

    }

  }

# Beschreibung der Anwendung

## Startseite

**Route**: /

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

KI-generierte Inhalte können fehlerhaft sein.

Beim Aufrufen der Webanwendung gelangen Benutzer auf die Startseite. Auf der Startseite erhält der Benutzer eine direkte Übersicht über die wichtigsten Informationen. Diese dient als zentrales Dashboard und zeigt auf einen Blick die wichtigsten Informationen:

* **Gesamtanzahl der geplanten Termine**
* **Anzahl offener Wartelisteneinträge**
* **Details des nächsten anstehenden Termins** (Verwendung der erstellten Termin-Komponente)

Die Daten werden dynamisch aus der MongoDB geladen und stets aktuell dargestellt. Die Informationen sind kompakt in visuell hervorgehobenen Karten strukturiert und ermöglichen dem Benutzer eine schnelle Orientierung.

Im oberen Bereich befindet sich eine Navigationsleiste, die durch die Datei **+layout.svelte** bereitgestellt wird und auf allen Seiten verfügbar ist. Sie erlaubt einen direkten Zugriff auf die weiteren Unterseiten wie Terminübersicht, Zahnarztliste oder das Formular zur Erstellung eines neuen Termins.

**Dateien:**

* routes/+page.svelte
* routes/+layout.svelte
* routes/+page.server.js
* lib/components/Termin.svelte

## Terminübersicht

**Route**: /termine

Ein Bild, das Text, Screenshot, Software, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Beim Navigieren zur Seite «Termine» erhält der Benutzer eine strukturierte Übersicht aller erfassten Termine. Jeder Termin wird als eigenständige Karte dargestellt und mit der Behandlungsart, dem Termin-Datum und dem Preis der Behandlung zeigt:

Zusätzlich befindet sich oberhalb der Liste ein Button «+ Neuen Termin erfassen», mit dem die Seite zur Terminerfassung aufgerufen werden kann. Die Inhalte werden dynamisch aus der MongoDB-Datenbank geladen und im Frontend mithilfe von Svelte-Komponenten gerendert.

**Dateien:**

* routes/termine/+page.svelte
* routes/termine/+page.server.js
* lib/components/Termin.svelte

## Termin-Details

**Route**: /termine/[id]

Ein Bild, das Text, Menschliches Gesicht, Screenshot, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.

Beim Aufruf dieser Seite werden alle Informationen zum ausgewählten Termin angezeigt. Dazu gehören die gebuchte Behandlung, das Datum und die Uhrzeit, der Preis sowie der Status des Wartelisteneintrags. Zusätzlich wird das Profil des zugewiesenen Zahnarztes eingeblendet, inklusive Bild, Praxisname, Adresse und Kontaktdaten.

Die Terminseite enthält zwei Schaltflächen. Mit dem «Zurück»-Button gelangt man wieder zur Übersicht aller Termine. Über «Löschen» kann der angezeigte Termin vollständig entfernt werden.

Diese Seite kombiniert Informationen aus der termine- und zahnaerzte-Collection, um einen vollständigen Überblick zu geben.

**Dateien:**

* routes/termine/[id]/+page.svelte
* routes/termine/[id]/+page.server.js
* lib/components/ZahnarztDetails.svelte

## Neuen Termin erstellen

**Route**: /termine/create

Auf dieser Seite können Benutzer einen neuen Termin erfassen. Die Ansicht besteht aus einem strukturierten Formular mit folgenden Elementen:

* **Behandlungswahl:** Die Auswahl der Behandlung erfolgt über farblich hervorgehobene Buttons. Diese sind fest definiert. Je nach gewählter Behandlung wird der Preis automatisch berechnet und gespeichert.
* **Datum und Uhrzeit:** Hier wird ein Date-Picker verwendet, der auf dem Kalender-Element von Bootstrap basiert. Benutzer können das gewünschte Datum und die Uhrzeit auswählen.
* **Zahnarztwahl:** Ein Dropdown-Menü listet alle verfügbaren Zahnärzte aus der Datenbank. Sobald ein Zahnarzt gewählt wurde, wird automatisch dessen Profil mittels der wiederverwendbaren Zahnarzt-Komponente angezeigt.
* **Warteliste:** Optional kann über eine Checkbox festgelegt werden, ob der Termin als Wartelisteneintrag behandelt werden soll.

Ein Bild, das Text, Menschliches Gesicht, Screenshot, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.

Die Seite enthält zwei Schaltflächen. Mit dem «Zurück»-Button gelangt man wieder zur Übersicht aller Termine. Nach dem Ausfüllen des Formulars wird der Termin durch Klick auf „Termin vereinbaren“ erstellt und in der Datenbank gespeichert. Bei Knopfdruck werden die Eingabefelder des Formulars validiert und entsprechend wird der Benutzer auf Pflichtfelder hingewiesen.

**Dateien:**

* routes/termine/create/+page.svelte
* routes/termine/create/+page.server.js
* lib/components/ZahnarztDetails.svelte

## Zahnärzte-Übersicht

**Route**: /zahnaerzte  
Ein Bild, das Text, Screenshot, Software, Webseite enthält.

KI-generierte Inhalte können fehlerhaft sein.

Auf dieser Seite erhalten Benutzer eine vollständige Übersicht über alle verfügbaren Zahnärzte. Jeder Zahnarzt wird mittels der wiederverwendbaren Zahnarzt-Komponente angezeigt, die den Namen, die zugehörige Praxis und ein Portraitbild enthält. Die Zahnärzt-Daten werden dynamisch aus der MongoDB-Datenbank geladen und angezeigt. Die Bilddateien der jeweiligen Profile folgen einer einheitlichen Namenskonvention (VORNAME\_NACHNAME.png) und werden automatisch über den Namen geladen (z. B. images/Agata\_Stutz.png).

Ziel dieser Ansicht ist es, den Patient einen schnellen Überblick über alle verfügbaren Fachpersonen zu ermöglichen zur Orientierung vor der Terminbuchung

**Dateien**:

* routes/zahnaerzte /+page.svelte
* routes/zahnaerzte /+page.server.js
* lib/components/Zahnarzt.svelte

## Zahnarzt-Details

**Route**: /zahnaerzte/[id]

Ein Bild, das Text, Menschliches Gesicht, Screenshot, Mann enthält.

KI-generierte Inhalte können fehlerhaft sein.

Diese Seite zeigt die vollständigen Informationen zu einem ausgewählten Zahnarzt. Nachdem auf der Zahnarztübersicht ein Eintrag ausgewählt wurde, wird das zugehörige Zahnarztprofil auf einer eigenen Seite angezeigt. Die Seite enthält relevanten Informationen zum ausgewählten Zahnarzt. Dazu gehören ein Portraitbild, der vollständige Name, der Name der Praxis, die Adresse inklusive Strasse, Postleitzahl und Ort und auch die Telefonnummer und E-Mail-Adresse. Diese Detailseite dient zur besseren Orientierung bei der Zahnarztauswahl

**Dateien:**

* routes/zahnaerzte/[id]/+page.svelte
* routes/zahnaerzte/[id]/+page.server.js
* components/ZahnarztDetails.svelte

## Wiederverwendbare dynamische Komponenten

Neben den Komponenten der Hauptseiten (Startseite, Terminübersicht, Termin-Detailansicht, Terminerfassung, Zahnarztübersicht und Zahnarzt-Detailseite) wurden auch wiederverwendbare Komponenten entwickelt, um dynamische Inhalte übersichtlich darzustellen.

Dazu gehören unter anderem:

* **Termin.svelte**: Darstellung einzelner Termine innerhalb der Übersicht
* **Zahnarzt.svelte**: Darstellung einzelner Zahnärzte innerhalb der Listenansicht
* **ZahnarztDetails.svelte**: Detaillierte Informationen zum ausgewählten Zahnarztprofil

## Backend-Funktionalitäten

In der Datei lib/db.js befinden sich folgende Kernfunktionen:

* **getTermine():** Holt alle Termine
* **getTerminFullDetails(id):** Holt Detaildaten eines Termins inkl. Zahnarzt
* **createTermin(termin):** Speichert einen neuen Termin
* **deleteTermin(id):** Löscht einen Termin
* **getZahnaerzte():** Holt alle Zahnärzte

Neu hinzugefügt:

* **getTermineStats():** Gibt die Anzahl aller gespeicherten Termine zurück, für eine Zusammenfassung auf dem Dashboard
* **getWartelisteCount():** Gibt die Anzahl der Termine zurück, bei denen ein Wartelisteneintrag aktiv ist

# Erweiterungen

## Dashboard mit Statistiken

Auf der Startseite werden die Gesamtanzahl der Termine sowie die Anzahl der Termine mit Wartelistenstatus angezeigt. Die Werte werden dynamisch über neue Backendfunktionen aus MongoDB geladen. Ebenfalls wird auch der nächste Termin geladen und mittels Termin-Komponente auf der Satrtseite angezeigt:

**Implementierung:**

* lib/db.js: getTermineStats(), getWartelisteCount()
* routes/+page.server.js
* routes/+page.svelte:   
  Ersten Termin aus Termin-Result auslesen

const termin = data.termine && data.termine.length > 0 ? data.termine[0] : null;

## Dynamische Zahnarzt-Profilanzeige bei Auswahl im Formular

Bei der Terminbuchung wird nach Auswahl eines Zahnarztes im Dropdown automatisch das Profil des Zahnarztes unterhalb angezeigt. Dies geschieht ohne Neuladen der Seite. Dafür wurde $state und $derived verwendet, um bei veränderter Auswahl des Dropdowns das Zahnarzt-Profil neu zu laden.

**Implementierung**:

* lib/components/ZahnarztDetails.svelte
* routes/termine/create/+page.svelte:   
  Binding von SelectedId mit Dropdown-Value und das Setzen des ersten Zahnarztes als Wert

selectedId = $state(data.zahnaerzte.length > 0 ? data.zahnaerzte[0].\_id : null)

Zahnarzt-Profil wird bei Änderung des selektierten Zahnarztes aktualisiert

selectedZahnarzt = $derived(data.zahnaerzte.find((d) => d.\_id === selectedId),

## Preisermittlung anhand Behandlungsauswahl

Die Behandlungsbuttons im Formular sind fest vorgegeben. Bei Auswahl wird automatisch der passende Preis aus einem Mapping gesetzt und mit dem Termin gespeichert. Dabei wird zur Umsetzung davon ausgegangen, dass es bei den Zahnärzten keinen Preisunterschied bei den Behandlungen gibt.

**Implementierung:**

* routes/termine/create/+page.server.js

Preis wird durch die ausgewählte Behandlungsart im switch-Block ermittelt

## Dynamisches Zahnarzt-Bildladen per Namenskonvention

Portraitbilder der Zahnärzte werden automatisch anhand von Vor- und Nachnamen geladen (z. B. images/Jil\_Müller.png). Dies reduziert ermöglicht einfache Skalierung der Bilder.

**Implementierung:**

* lib/components/ZahnarztDetails.svelte
* lib/components/Zahnarzt.svelte  
  Aufruf durch <img src=images/{zahnarzt.vorname}\_{zahnarzt.nachname}.png

## Warteliste als Checkbox im Termin-Formular

Die Möglichkeit, sich auf die Warteliste zu setzen, wurde mittels eines integrierten Bootstrap-Checkbox-Elements umgesetzt. Der Wert wird gespeichert und auf dem Dashboard ausgewertet.

**Implementierung:**

* routes/termine/new/+page.svelte
* Bootstrap Checkbox

## Warteliste als Checkbox im Termin-Formular

Die Möglichkeit, sich auf die Warteliste zu setzen, wurde mittels eines integrierten Bootstrap-Checkbox-Elements umgesetzt. Der Wert wird gespeichert und auf dem Dashboard ausgewertet.

**Implementierung:**

* routes/termine/new/+page.svelte
* Bootstrap Checkbox

## Wiederverwendbare Komponenten für klare Struktur

Mehrere Komponenten wurden modular aufgebaut, um Wiederverwendung sicherzustellen.

* **Termin.svelte**: Wiederverwendung auf der Startseite und Terminübersicht
* **Zahnarzt.svelte**: Darstellung einzelner Zahnärzte innerhalb der Listenansicht
* **ZahnarztDetails.svelte**: Die detaillierte Informationen zum ausgewählten Zahnarztprofil wird bereits vermehrt aufgerufen auf der Termin-Detailansicht, der Zahnarzt-Detailansicht und der Terminerfassung

## Erweiterte Termin-Detailansicht mit Zahnarztdetails

Die Detailansicht eines Termins wurde um ein eingebettetes Zahnarztprofil erweitert. Dadurch erhalten Benutzer direkt beim Öffnen eines Termins auch relevante Daten des Zahnarztes. Angezeigt werden Portraitbild, Vor- und Nachname, Praxisname sowie Kontaktdaten.

Diese Informationen werden über die Komponente ZahnarztDetails.svelte eingebunden und basieren auf einer datenbankübergreifenden Fremdschlüssel-Verknüpfung per zahnarztId. Die Verknüpfung und Kombination der Daten findet in der Funktion getTerminFullDetails(id) in lib/db.js statt.

## Verwendung von Bootstrap-Icons

Zur besseren Visualisierung wurden Icons aus dem Bootstrap-Icons-Set verwendet, z. B. für Datum, Uhrzeit, Zahnarzt oder Behandlung. Die Icons wurden direkt über Klassen wie «bi bi-calendar-event» eingebunden.