

INF202 SE

Krankenhaus Terminsystem Architekturspezifikation

40. Komando Alayı

Ahmet Oğuz Örsler 200503035

Kaan İmamoğlu 200503011

Architekturüberblick

- Die Schichten-Trennung ermöglicht eine saubere Strukturierung des Systems, bei der Datenzugriff, Geschäftslogik und Benutzeroberfläche voneinander getrennt sind. Die API wird vom Controller definiert, die Geschäftslogik wird vom Service ausgeführt und die Datensicherheit wird vom Repository gewährleistet.
- Dadurch können Änderungen an einer Schicht vorgenommen werden, ohne dass die anderen beeinträchtigt werden, und das System kann ohne Schwierigkeiten skaliert werden.

Architekturüberblick

- Unser Projekt wird eine mehrschichtige Architektur verwenden, um Datenzugriff, Geschäftslogik und Darstellungsschichten sauber zu trennen. Die folgenden Schnittstellen werden definiert:

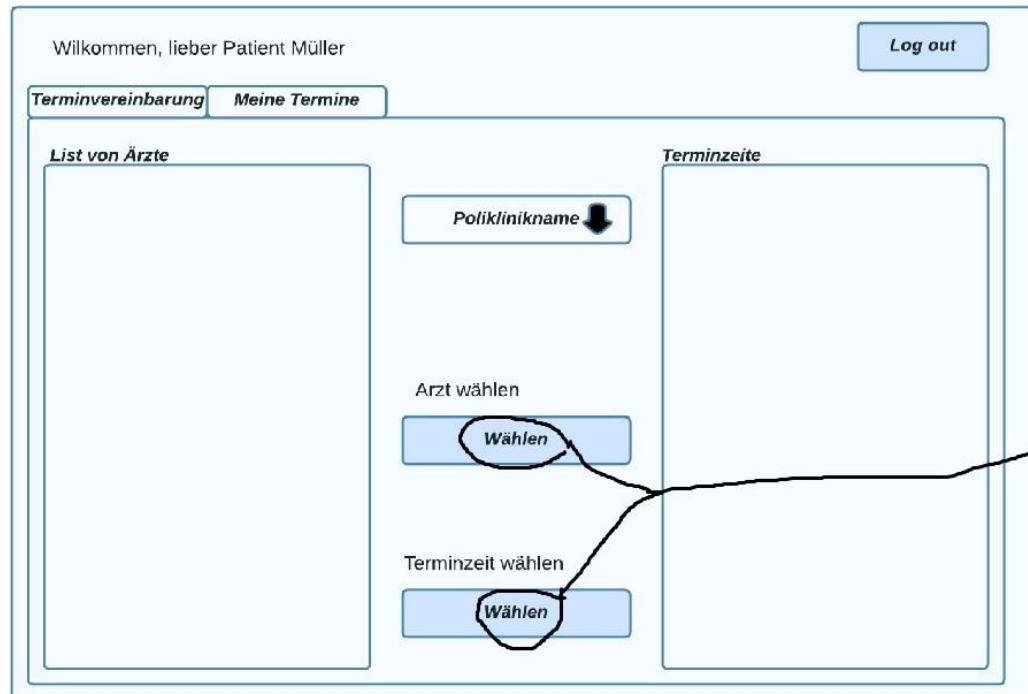




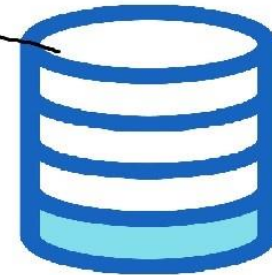
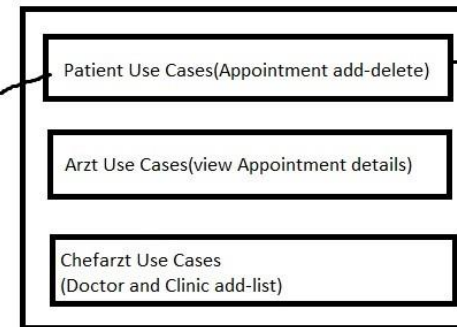
Architekturüberblick

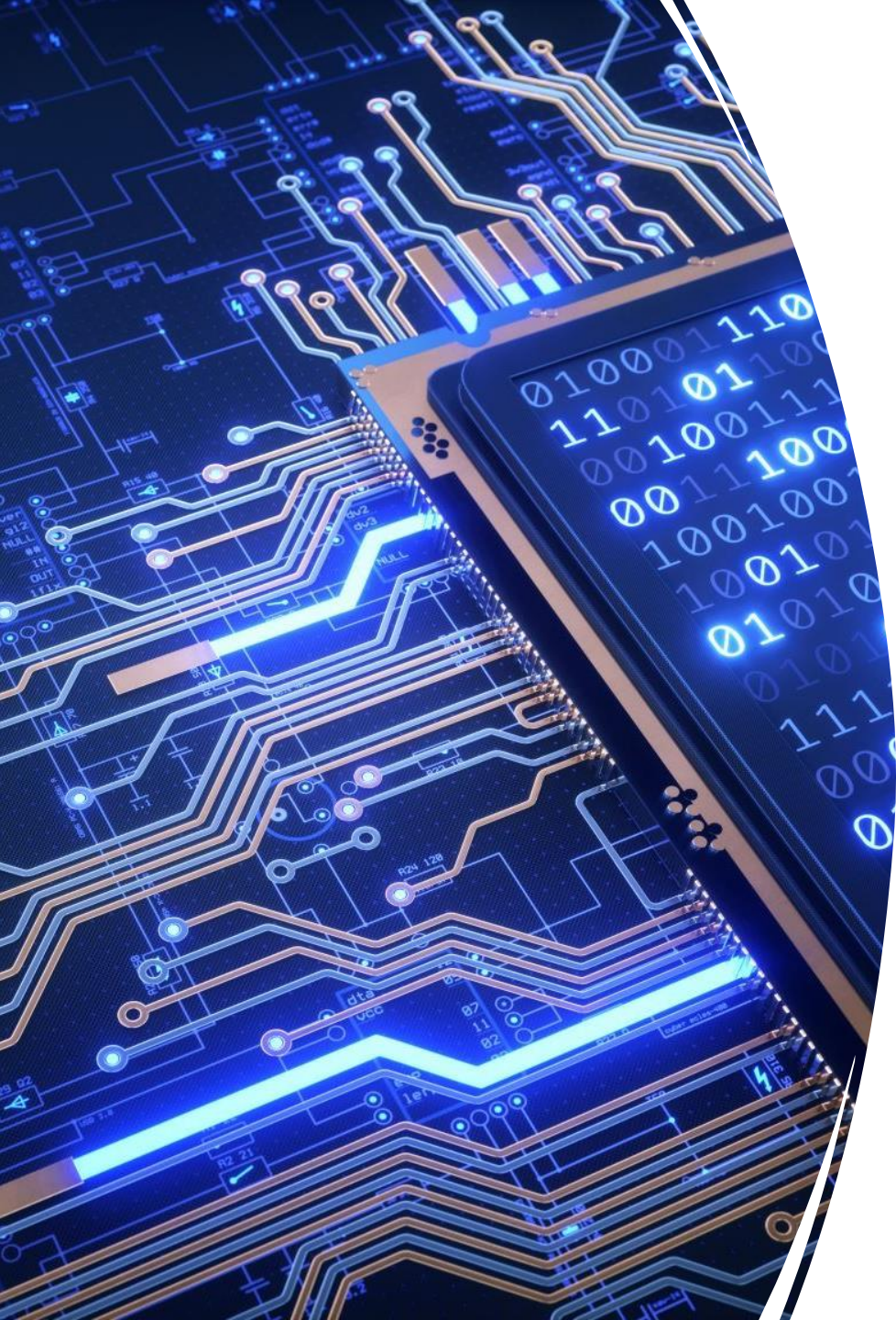
1. Wir werden die APIs und Webservices in der Controller-Komponente definieren, die für die Kommunikation zwischen der Benutzeroberfläche und der Geschäftslogik verwendet werden.
2. Wir werden die Geschäftslogik in der Service-Komponente implementieren, um die Anforderungen der Use Cases zu erfüllen. Jeder Schritt oder Use Case kann eine spezifische Servicekomponente haben.
3. Wir werden den Datenzugriff und die Datenpersistenz in der Repository-Komponente implementieren.

Architekturüberblick



USE CASES





Architekturüberblick

- Die Komponenten kommunizieren über bestimmte Schnittstellen. Wir werden die folgenden Frameworks und Technologien verwenden:
- Die Backend-Anwendung wird in Java entwickelt und verwendet das Java Framework für die Service-Komponente. RESTful APIs werden verwendet, um mit der Benutzeroberfläche zu kommunizieren.
- Frontend: Die Frontend-Anwendung ist in Java Swing implementiert.

Beschreibung der „Controller“ Klassen



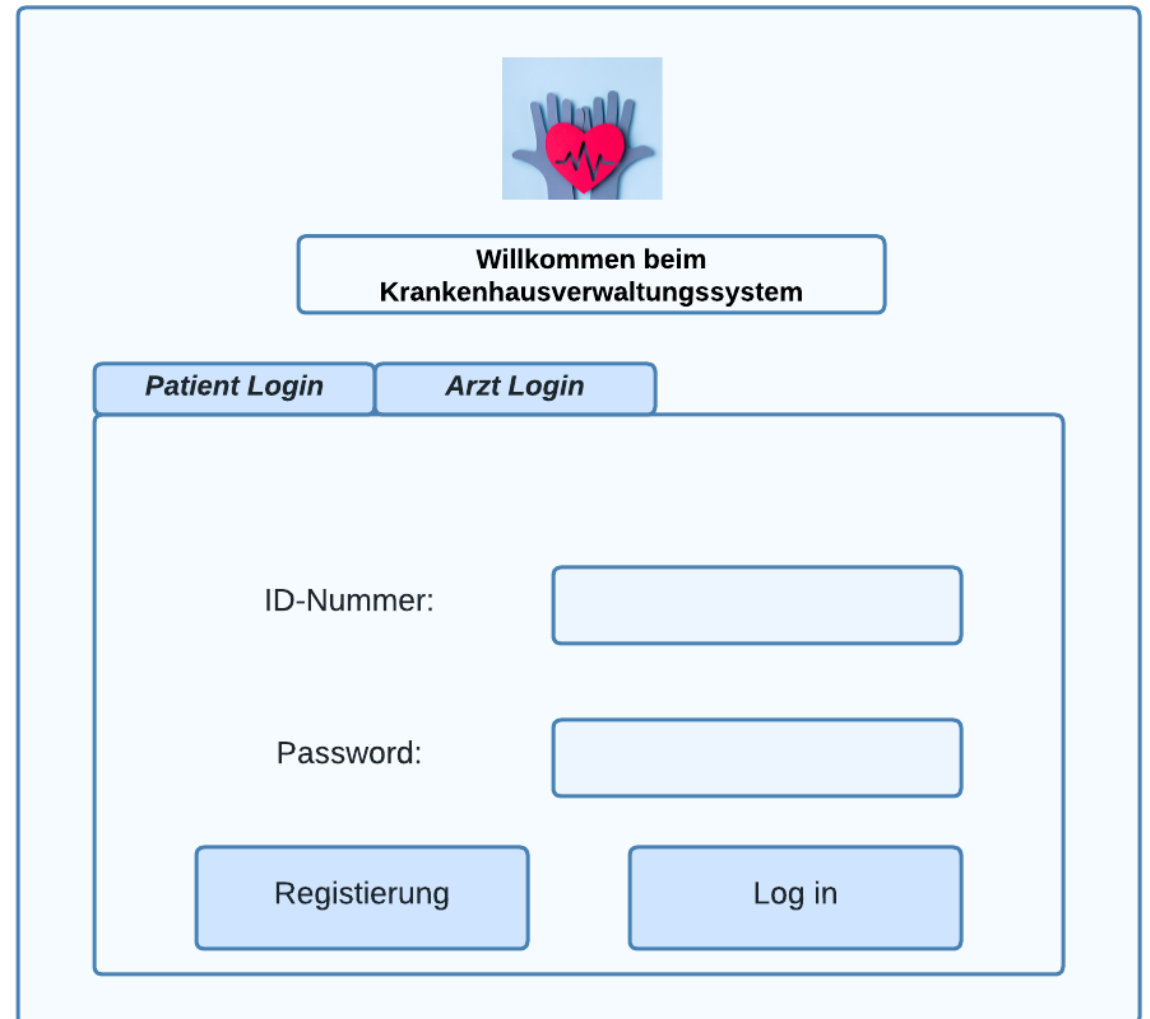
UM EINEN TERMIN ZU VEREINBAREN, IST DER CONTROLLER DER ANWENDUNG DAFÜR VERANTWORTLICH, DIE ANFRAGEN DES BENUTZERS ENTGEGENZUNEHMEN UND DIE ENTSPRECHENDEN MAßNAHMEN ZU ERGREIFEN. DER CONTROLLER KOMMUNIZIERT MIT DEM SERVICE, UM AUF DIE DATENBANK ZUZUGREIFEN UND DIE ERFORDERLICHEN DATEN ABZURUFEN ODER ZU SPEICHERN.



UM EINEN TERMIN FÜR DIESES PROJEKT ZU VEREINBAREN, MUSS DER PATIENT DIE FOLGENDEN SCHRITTE AUSFÜHREN:

Beschreibung der „Controller“ Klassen Login UI

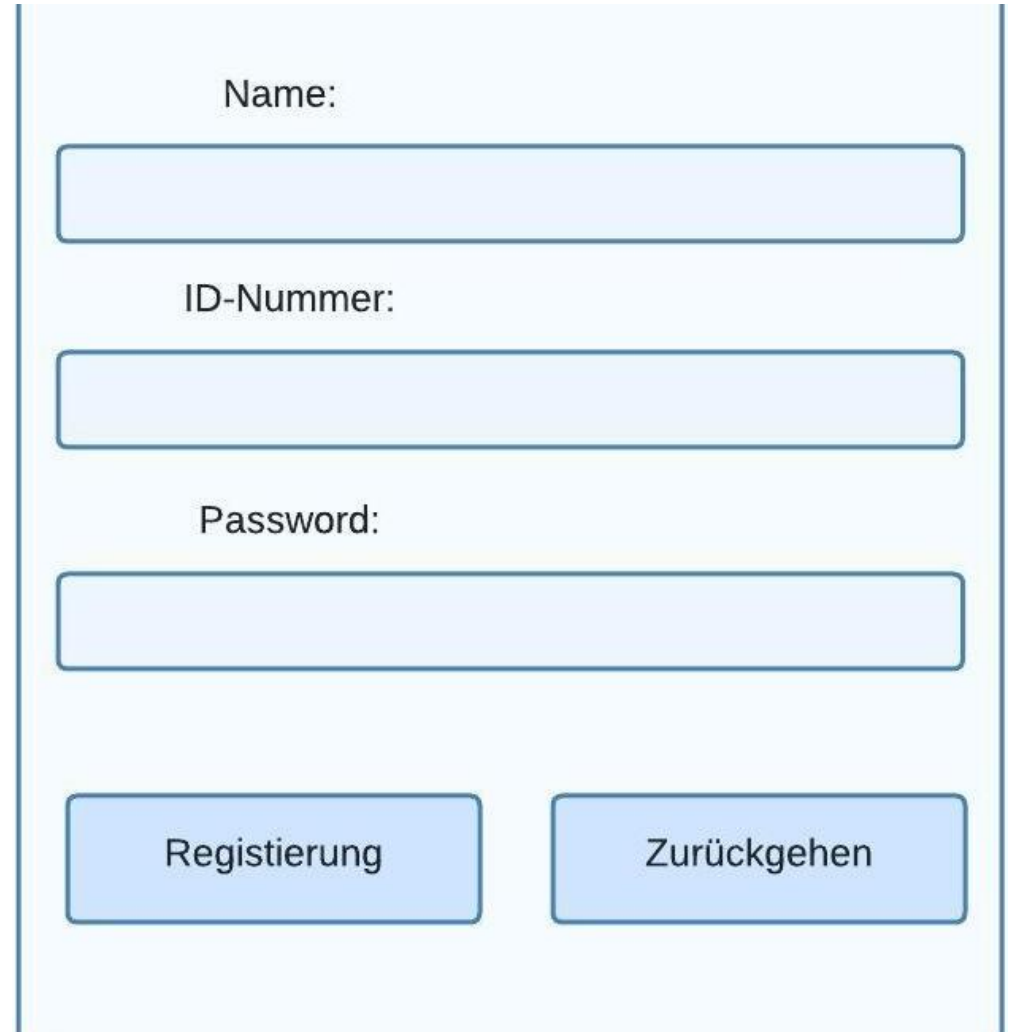
- Wenn der Benutzer auf die Login-Button klickt, werden die ID-Nummer und das Passwort an den Server gesendet, um die Anmeldeinformationen zu überprüfen. Wenn die Anmeldeinformationen korrekt sind, wird der Benutzer auf die Startseite weitergeleitet. Wenn die Anmeldeinformationen falsch sind, wird dem Benutzer eine Fehlermeldung angezeigt und er wird aufgefordert, seine Anmeldeinformationen erneut einzugeben.
(getPatientID-getPatientPassword.equals() ==> Database(MySQL))



The image shows a user interface for a hospital management system. At the top center is a logo featuring two hands holding a heart with a pulse line. Below the logo is a welcome message in a rounded rectangle: "Willkommen beim Krankenhausverwaltungssystem". Underneath this are two tabs: "Patient Login" and "Arzt Login". The main content area contains two input fields labeled "ID-Nummer:" and "Password:". At the bottom of this area are two buttons: "Registrierung" and "Log in".

Beschreibung der „Controller“ Klassen Register UI

- Die Registrierungsfunktion ermöglicht es Ihnen, neue Benutzer im System zu registrieren. Zu diesem Zweck wird eine Registrierungsseite bereitgestellt, auf der Benutzer ihre persönlichen Daten eingeben können. Nachdem der Benutzer alle erforderlichen Informationen eingegeben und auf die Button "Registrieren" geklickt hat, werden diese Informationen an den Server gesendet, wo sie überprüft und gespeichert werden. (addPatient(Name, ID-Nummer, Password)
==>Database(MySQL))



A registration form with a light blue background and a thin blue border. It contains three input fields for 'Name:', 'ID-Nummer:', and 'Password:'. At the bottom, there are two buttons: 'Registrierung' and 'Zurückgehen'.

Name:

ID-Nummer:

Password:

Registrierung

Zurückgehen

Beschreibung der „Controller“ Klasse Arzt UI

- In diesem Projekt können Ärzte ihre Termine nur einsehen, da das System darauf ausgelegt ist, dass Patienten Termine online buchen können. Termine werden über die Benutzerschnittstelle angezeigt, auf die der Arzt über die Benutzerschnittstelle zugreift. (getPatientID -> getDoctorID ==> Database(MySQL))

Willkommen, lieber Arzt Müller

Log out

Termine

PatientID	PatientName	TerminZeit
-----------	-------------	------------

Beschreibung der „Controller“ Klasse Patient UI

- Der Patient muss sich zuerst registrieren oder auf der Benutzeroberfläche anmelden.
- Danach muss nach der Anmeldung der Patient den Namen des Polikliniknames und des Arztes auswählen, bei dem er einen Termin vereinbaren möchte. (getDoctor-getPoliklinik)
- Der Patient muss das Datum und die Uhrzeit wählen, zu der er einen Termin vereinbaren möchte. (getDate-addAppointment)
- Die Arztliste und die Zeiten werden links und rechts angezeigt. Sie können Termine auch in «Meine Termine» einsehen. (getDoctorID-getDateID-getAppointment)

The image shows a web application interface for a patient appointment system. At the top, it says "Willkommen, lieber Patient Müller" and has a "Log out" button. Below this are two tabs: "Terminvereinbarung" (selected) and "Meine Termine". The main content area is divided into three sections: "List von Ärzte" (a large empty box on the left), "Poliklinikname" (a dropdown menu with a downward arrow), and "Terminzeit" (a large empty box on the right). In the center, there are two buttons labeled "Wählen". The first button is preceded by the text "Arzt wählen" and the second by "Terminzeit wählen".

Beschreibung der „Controller“ Klasse Chefarzt UI

- Der Chefarzt des Krankenhauses fungiert hier als Administrator. Er fügt den Ärzten des Krankenhauses in dem System hinzu und definiert für sie ein Passwort. An dieser Oberfläche können Sie die von Ihnen hinzugefügten Ärzte mit ihren Funktionen anzeigen. (addDoctor() - addPoliklinik() - deleteDoctor() - deletePoliklinik() ==> Database(MySQL))
- Darüber hinaus kann der Chefarzt dem Krankenhaussystem Kliniken hinzufügen und dieser Poliklinik Ärzte zuweisen. Chefarzt kann auch die Polikliniken im System einsehen.

The interface is titled "Willkommen, lieber Chefarzt Müller" and includes a "Log out" button. It features two tabs: "Ärzte Verwaltung" (selected) and "Poliklinik Verwaltung". The main content area is divided into a table and a form. The table has columns for "ID-Nummer", "Ausweis", "Name", and "Password". The form on the right contains input fields for "Name:", "Ausweis", and "Password:", followed by an "Add" button. Below the form is an input field for "ID-Nummer" and a "Löschen" button.

ID-Nummer	Ausweis	Name	Password
-----------	---------	------	----------

Name:

Ausweis

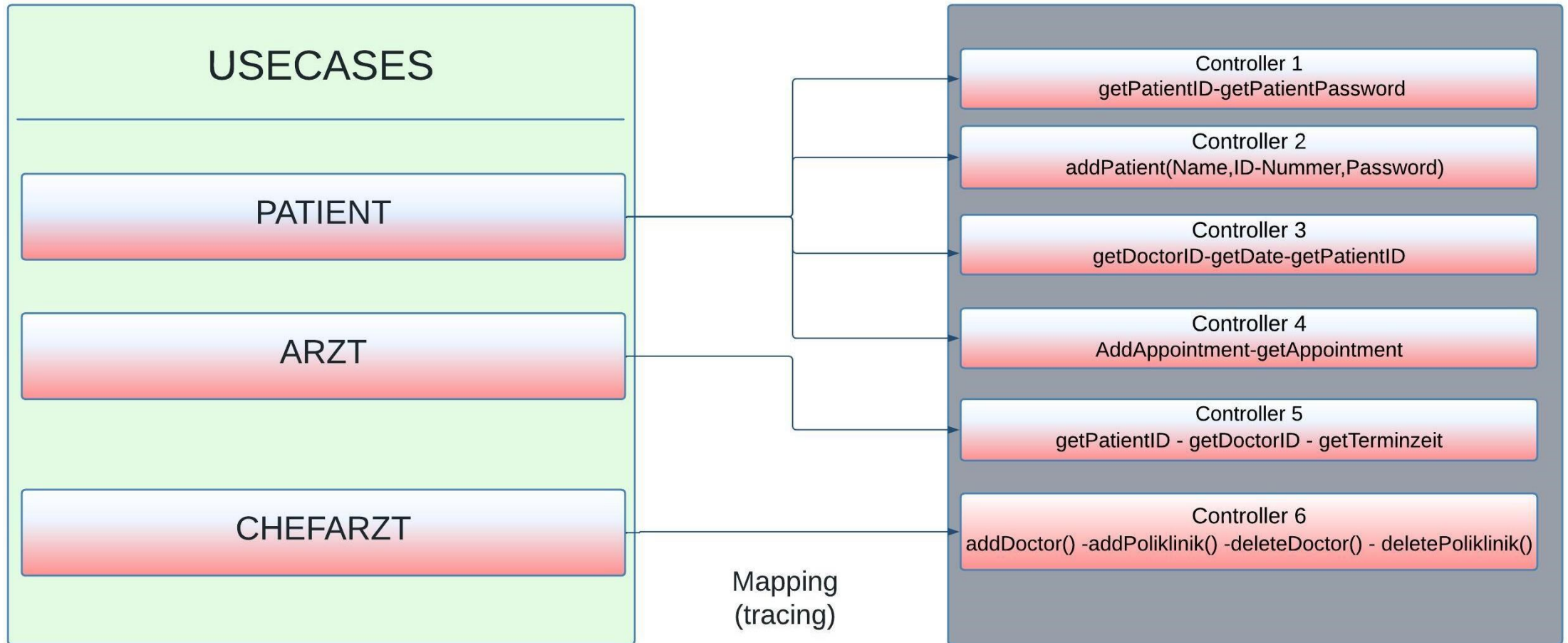
Password:

Add

ID-Nummer

Löschen

Rückverfolgbarkeit der Anforderungen





Datenbank Verwaltung

- Unsere Datenbank speichert die ID, die Identität, den Namen, den Typ und das Passwort des Patienten. Die ID, die Identität, der Name, der Typ und das Passwort des Arztes werden ebenfalls aufgezeichnet. Chefarzt-IDs, Identitäten, Namen, Typen und Passwörter werden ebenfalls aufgezeichnet. Die Zeitintervalle werden in der Datenbank gespeichert. Auch die Namen und ID-Nummer der Kliniken sind erhalten geblieben. Wenn der Chef jemanden hinzufügt oder entfernt, oder wenn sich jemand registriert, werden Name, Passwort und ID-Nummer dieser Person in die Datenbank übertragen.