**Dokumentation**

**Verteilte Systeme**



**SS18**

Jan Gutnik

[Jan.Gutnik@student.reutlingen-university.de](mailto:Jan.Gutnik@student.reutlingen-university.de)

Valeria Pagliaro

[Valeria.Pagliaro@student.reutlingen-university.de](mailto:Valeria.Pagliaro@student.reutlingen-university.de)

**Betreuer:** Prof. Dr.-Ing. Michael Tangemann

[*Michael.Tangemann@reutlingen-university.de*](Michael.Tangemann@reutlingen-university.de)

Inhaltsverzeichnis

[1 Anforderungen 3](#_Toc517641763)

[2 UML Diagramme 4](#_Toc517641764)

[2.1 Use Case Diagramm 4](#_Toc517641765)

[3 Mockups 7](#_Toc517641766)

[4 Datenbank 17](#_Toc517641767)

[5 Server 17](#_Toc517641768)

[6 Probleme 18](#_Toc517641769)

[7 Änderungen 19](#_Toc517641770)

[8 Softwaretest 19](#_Toc517641771)

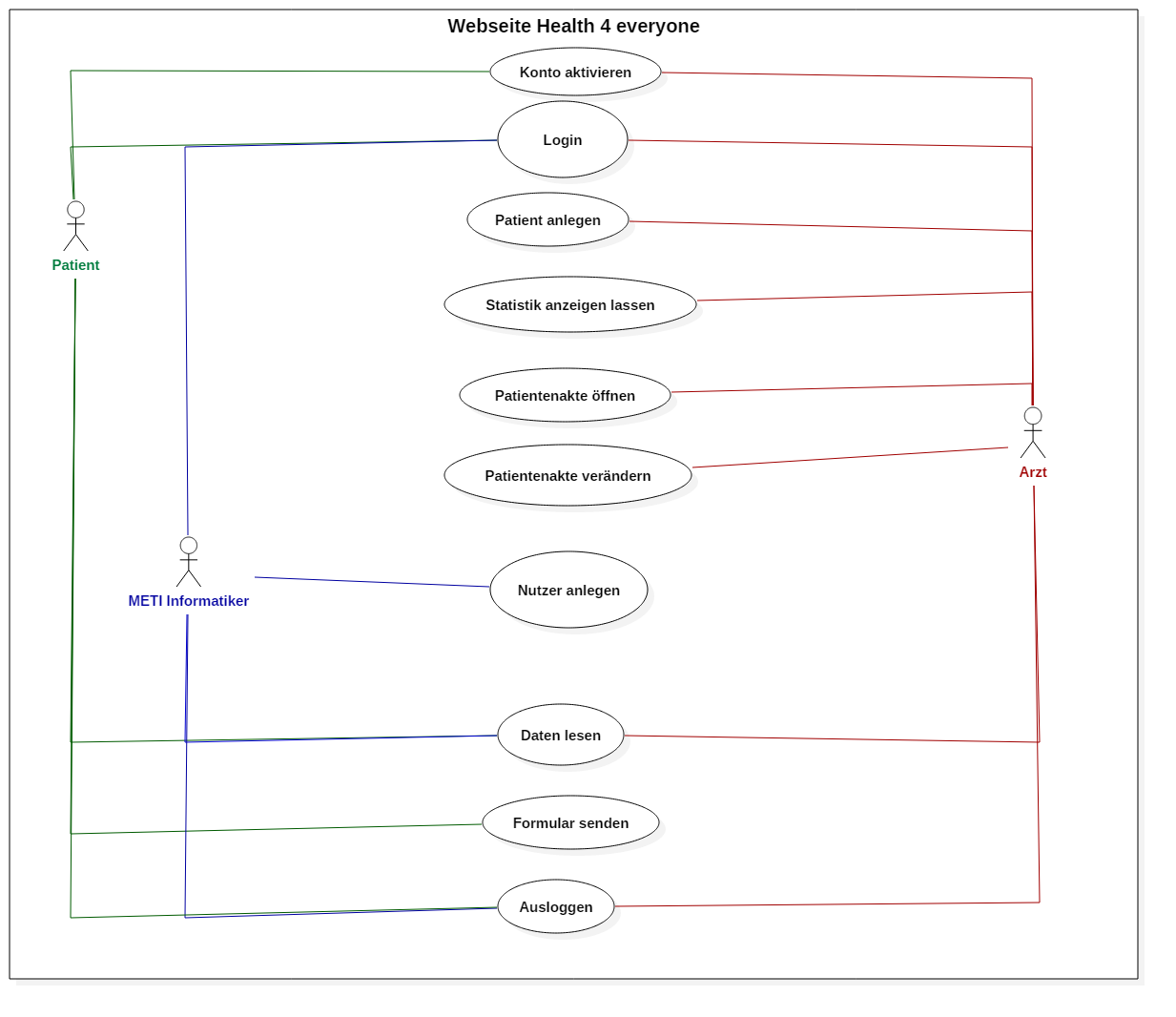
[9 Verwendete Software 19](#_Toc517641772)

# Anforderungen

* Es sollen unterschiedliche Vitalparameter (Gehirnwerte, Herzwerte, Lungenwerte und Nierenwerte) mithilfe des E-Health kits aufgenommen werden.
* Daten sollen verarbeitet, aufbereitet und in einer Webseite dargestellt werden.
* Es gibt drei verschiedene Rollen: Arzt, Patient und METI-Informatiker
* Daten sollen in einer Datenbank erfasst werden.
* Die verschiedenen Rollen dürfen außschließlich durch geprüfte Anmeldedaten auf die Webseite zugreifen.
* Patient: hat nur einen Lesezugriff und darf Veränderungen des Krankheitsbild in ein Formular eintragen und an den Arzt übermitteln.
* Arzt: Kann Patient anlegen und Patientenakte öffnen und verändern sowie statistische Berechnungen zu den Daten anzeigen lassen.
* Meti-Informatiker: Benutzername und Aktivierungspasswort generieren und automatische Versendung einer Email an den Patienten zur Accountaktivierung. Die statistischen Berechnungen werden vom Informatiker eingebettet.

# UML Diagramme

## Use Case Diagramm



**Name:** Konto aktivieren

**Beteiligte Akteure:** Patient, Arzt

**Vorbedingungen:**

* Patient wurde vom Arzt angelegt
* Arzt wurde vom Informatiger angelegt
* Patient oder Arzt hat eine Aktivierungsemail

**Auslöser:** Button „Aktivieren“ wurde betätigt.

Standardablauf: Button „Aktivieren“ betätigen, danach Benutzername und Aktivierungspasswort eingeben. Button „weiter“ drücken und neues Passwort eingeben und auf den Button „bestätigen“ klicken.

**Name:** Login

**Beteiligte Akteure:** Patient, Arzt und METI Informatiker

**Vorbedingungen:** In der Datenbank registriert

**Auslöser:** Button „Login“ betätigen und Formular Logindaten ausfüllen

**Standardablauf:** Button betätigen und einloggen. Danach wird man auf sein Profil weitergeleitet.

**Name:** Patient anlegen

**Beteiligte Akteure:** Arzt

**Vorbedingungen:** keine

**Auslöser:** Button „Patient anlegen“ betätigen und Formular ausfüllen.

**Standardablauf:** Button betätigen und Formular „Persönliche Daten“ danach Button „weiter“ klicken und „Vitalparamater“ und dann Button „bestätigen“ klicken ausfüllen. Die Daten werden dann in einer Datenbank hinterlegt.

**Name:** Statistik anzeigen lassen

**Beteiligte Akteure:** Arzt

**Vorbedingungen:** Patient ist bereits angelegt und es wurden Vitalparameter eingelesen

**Auslöser**: Button „Statistik“ betätigen

**Standardablauf:** Gewünschtes Feld anklicken (z.B Kopf) danach Button betätigen und es öffnet sich ein Fenster und zeigt die Statistik an.

**Name:** Patientenakte öffnen

**Beteiligte Akteure:** Patient, Arzt und METI Informatiker

**Vorbedingungen:** Patient ist angelegt

**Auslöser:** Button „Patientenakte öffnen“ betätigen

**Standardablauf:** Button anklicken, es öffnet sich ein Fenster „Suchmaske Patientenakte“ Daten eingeben und die gewünschte Akte öffnet sich.

**Name:** Patientenakte verändern

**Beteiligte Akteure:** Arzt

**Vorbedingungen:** Patient ist angelegt

**Auslöser:** Button „Ändern“ betätigen

**Standardablauf:** Button anklicken und gewünschten Wert im Feld eintragen.

**Name:** Nutzer anlegen

**Beteiligte Akteure:** METI Informatiker

**Vorbedingungen:** keine

**Auslöser**: Button „Nutzer anlegen“ drücken

**Standardablauf:** Button betätigen, dann Art des Nutzers wählen (Patient, Mitarbeiter) und Nutzerdaten eingeben.

**Name:** Formular senden

**Beteiligte Akteure:** Patient

**Vorbedingungen:** In der Datenbank registriert und eingeloggt

**Auslöser:** Button „Formular ausfüllen“ betätigen

**Standardablauf:** Button anklicken, Formular ausfüllen und dann auf Button „Senden“ drücken

**Name:** Ausloggen

**Beteiligte Akteure:** Patient, Arzt und METI Informatiker

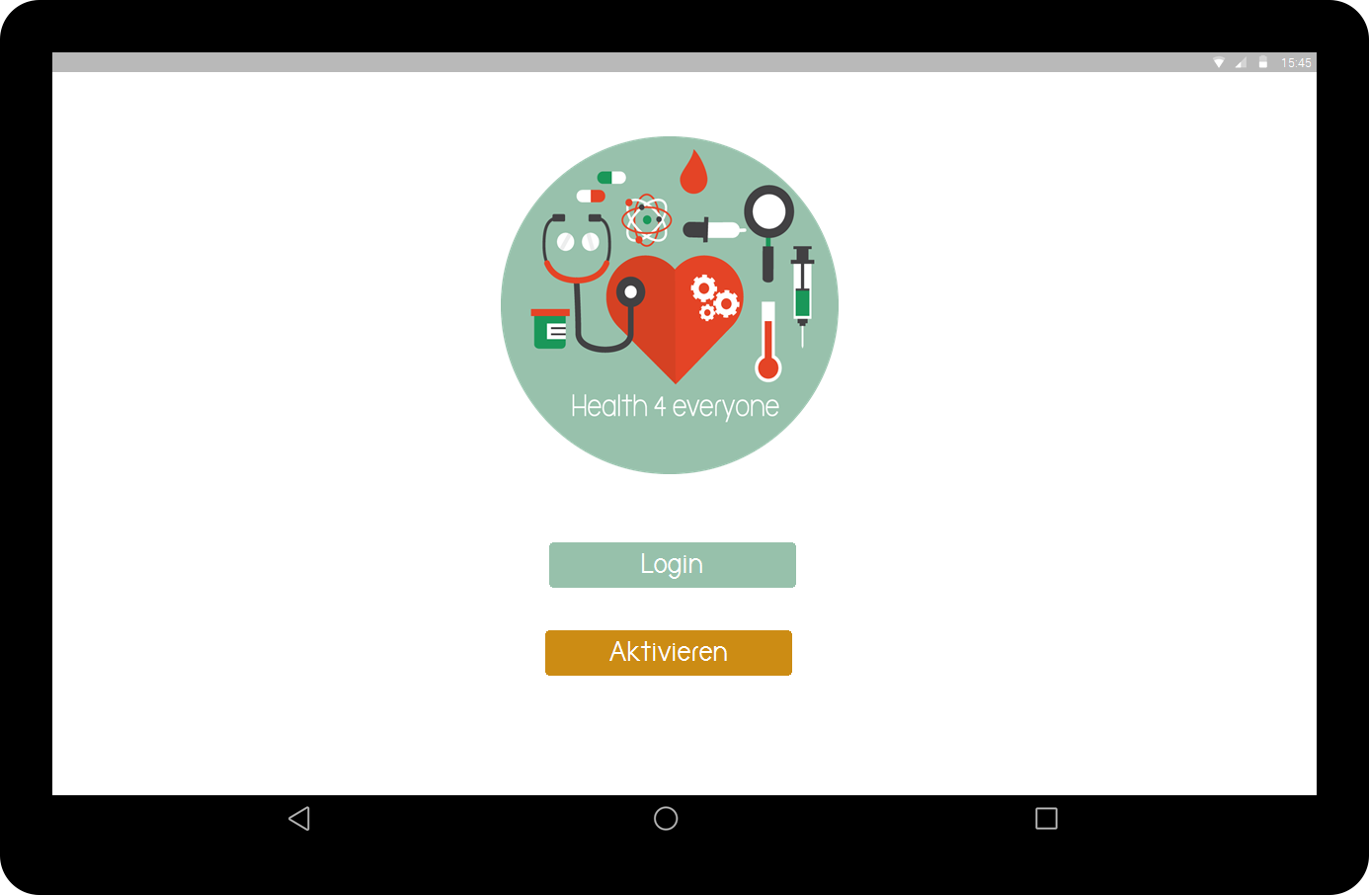
**Vorbedingungen:** Eingeloggt

**Auslöser:** Auf Button „Ausloggen“ drücken

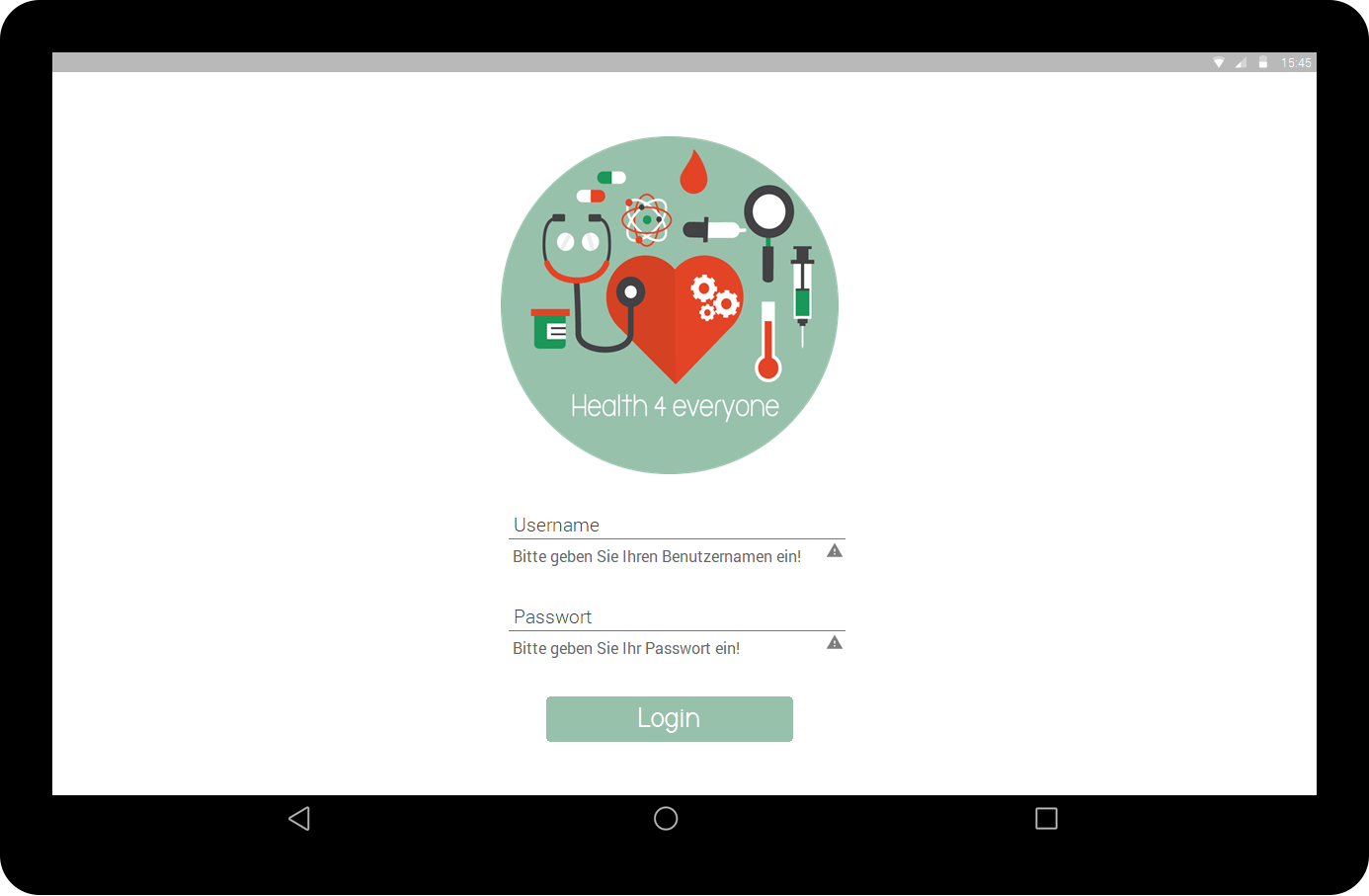
**Standardablauf:** Button drücken, der Nutzer wird dann ausgeloggt.

# Mockups

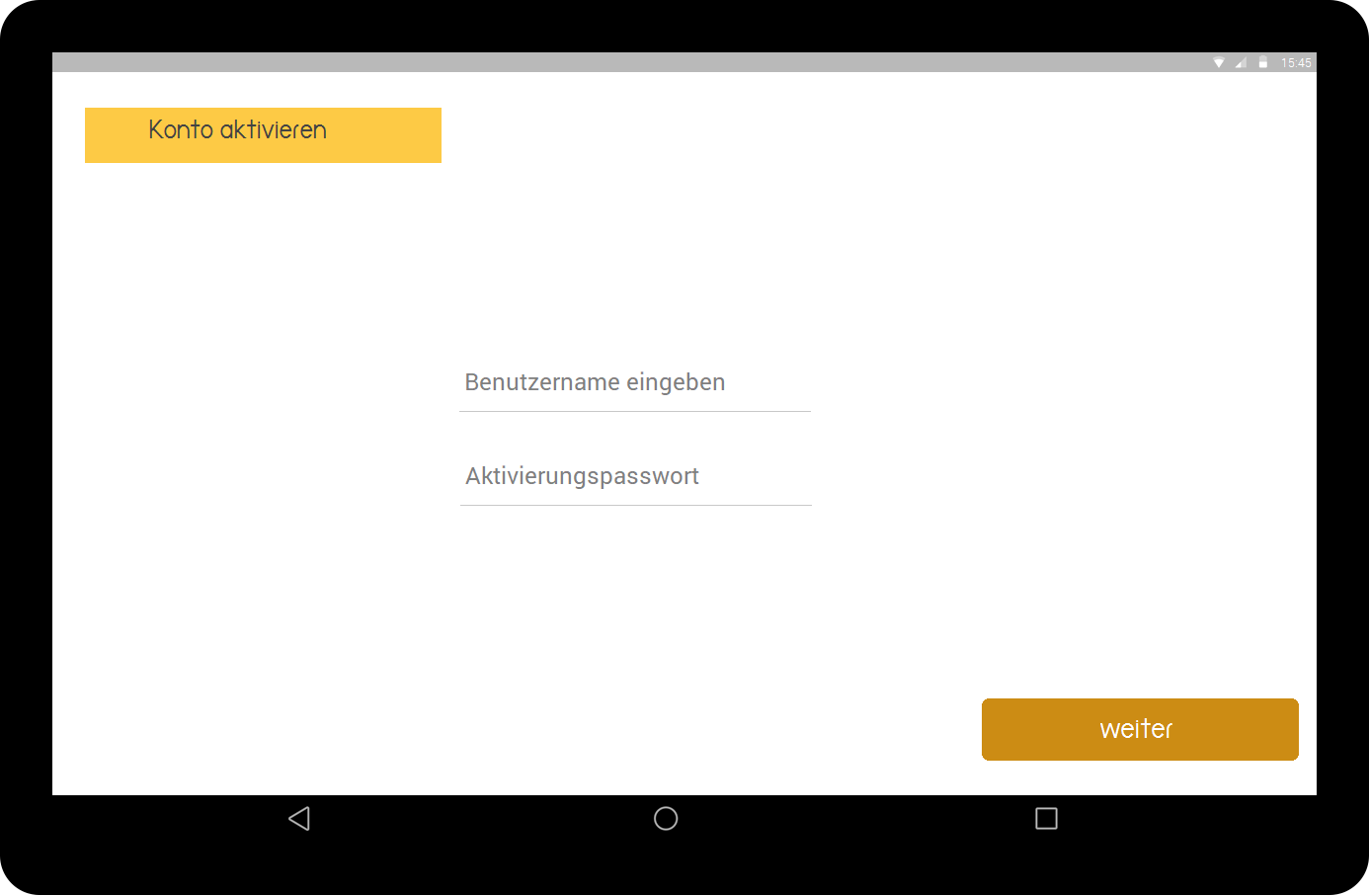
**Start Screen**



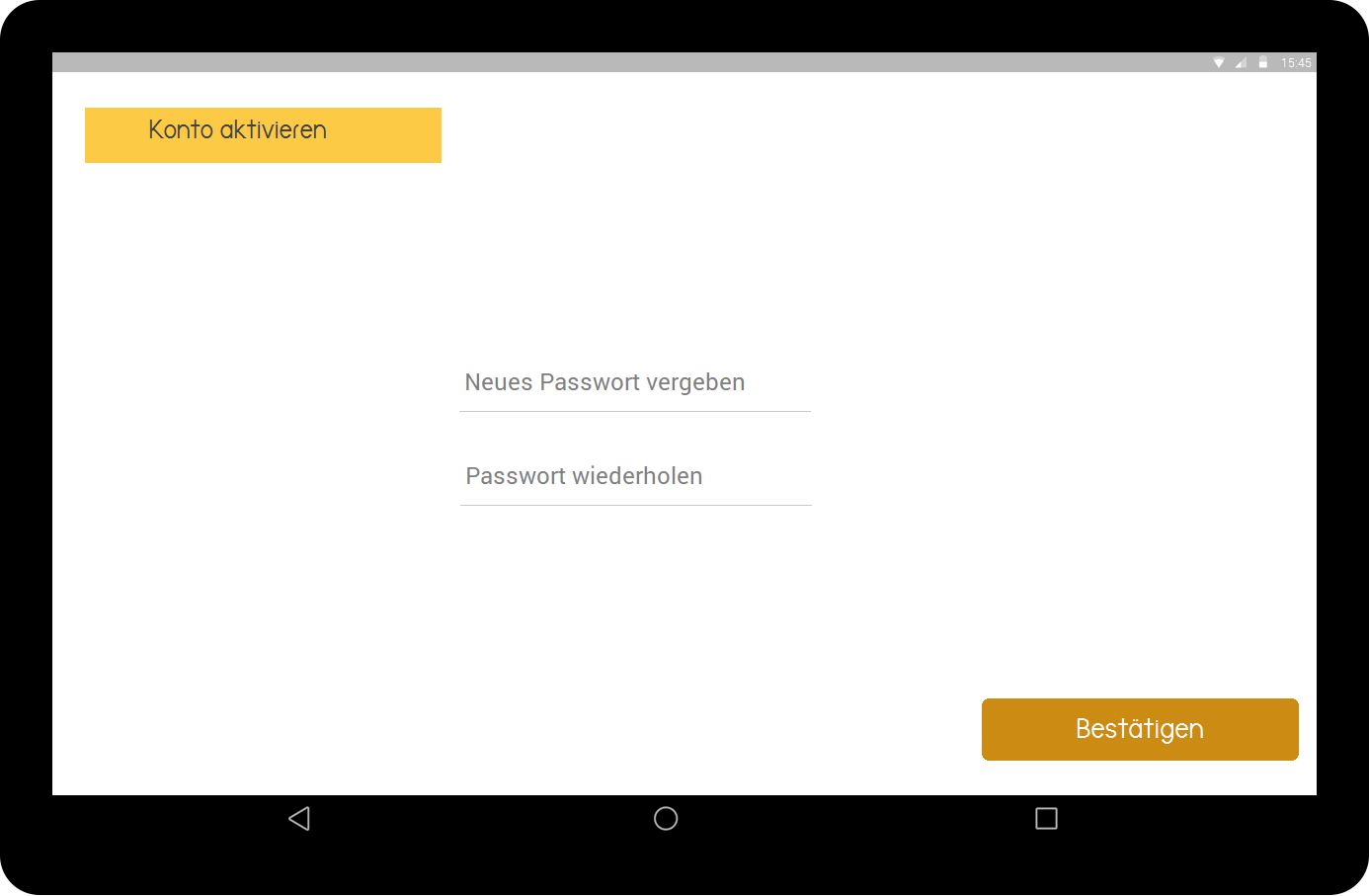
**Login Screen**



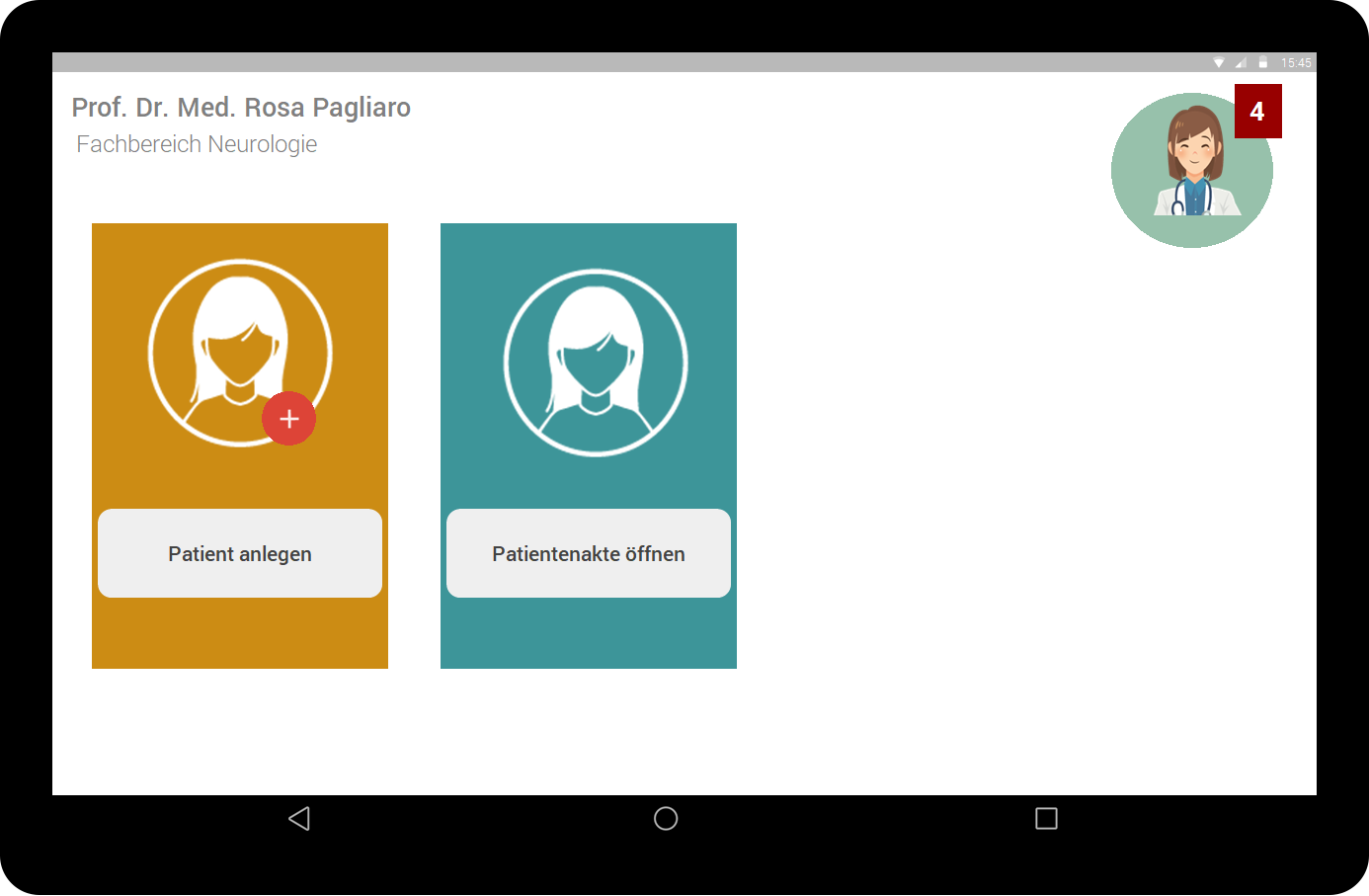
**Konto aktivieren**



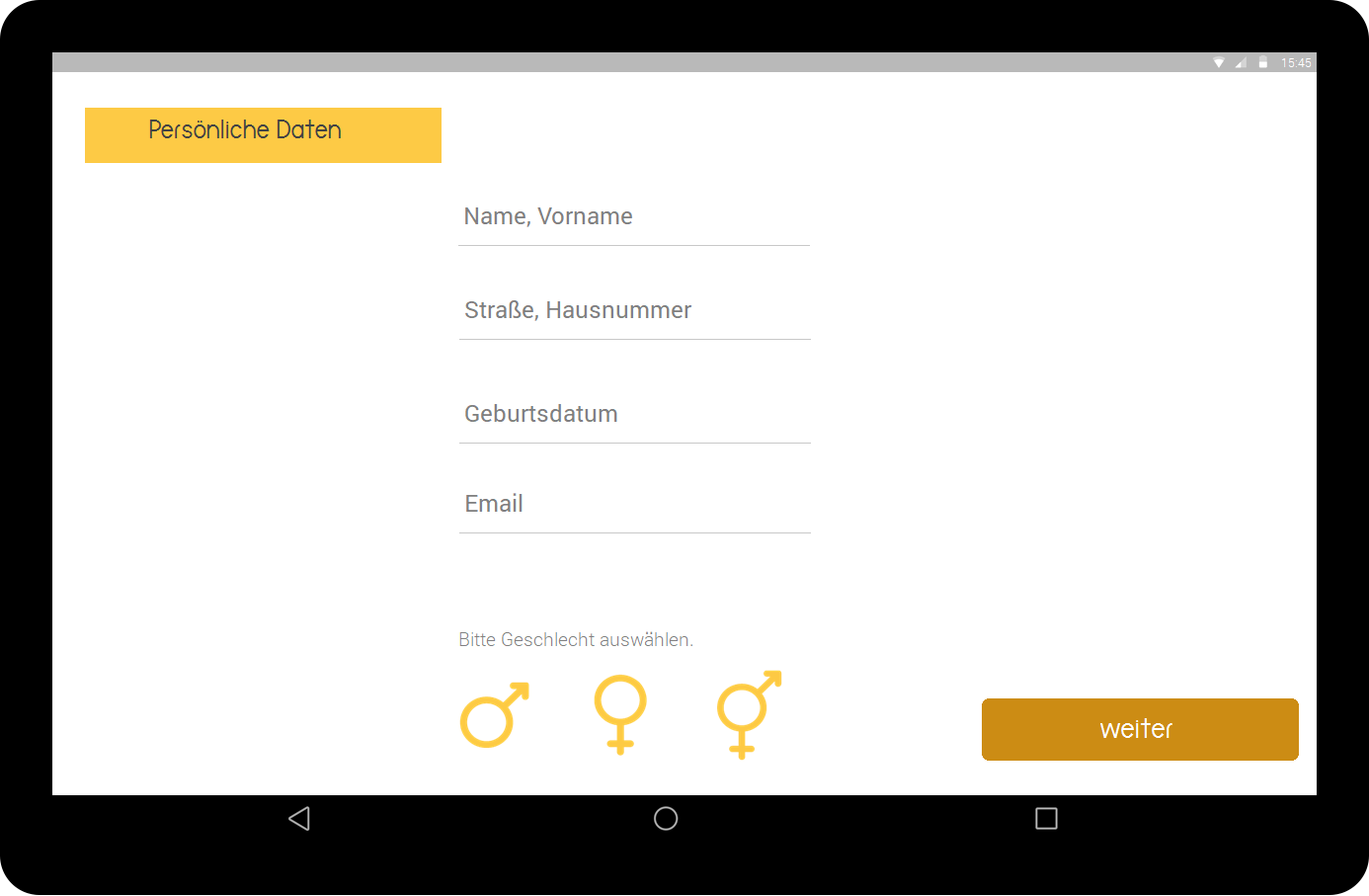
**Neues Passwort vergeben**



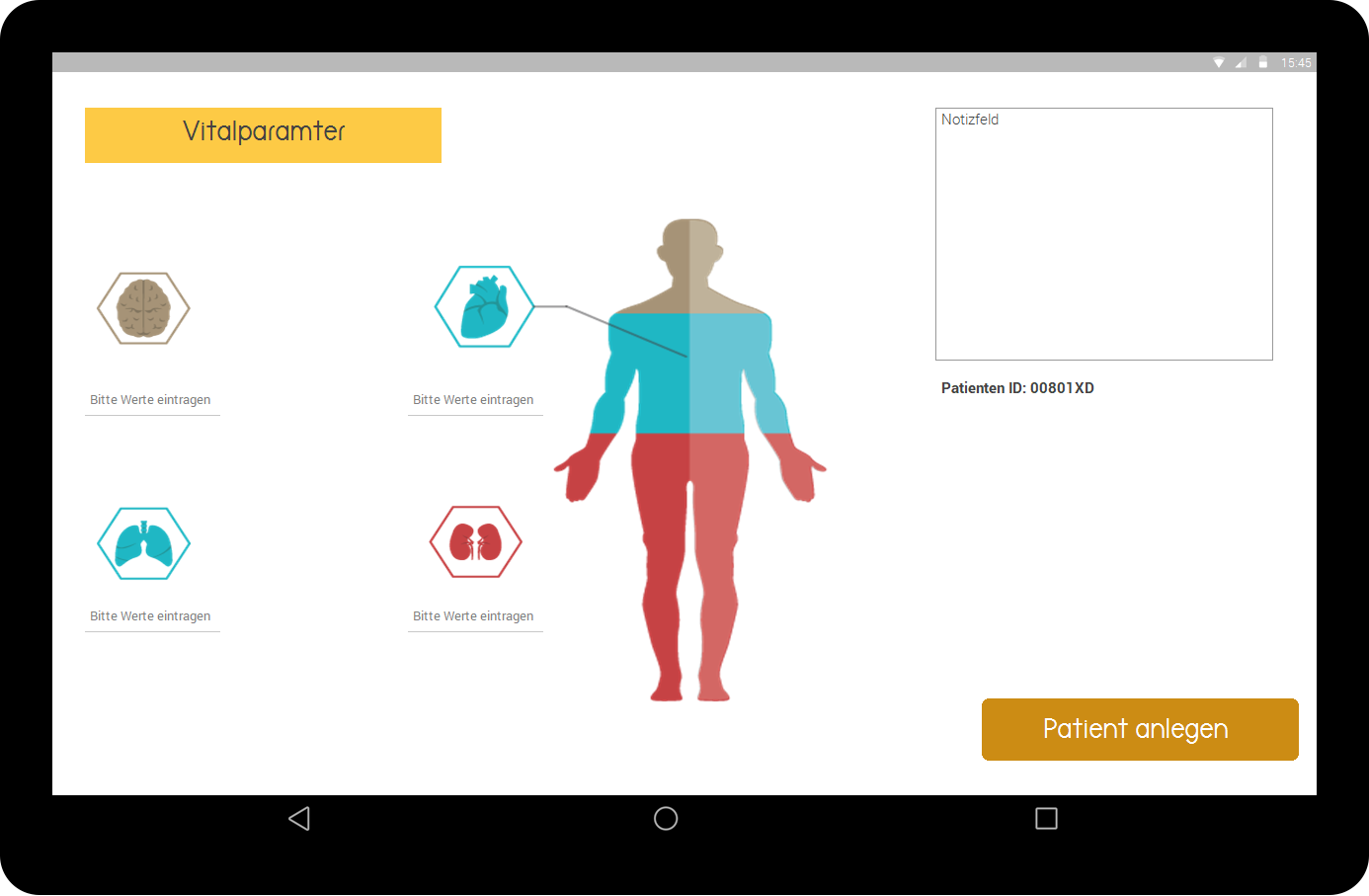
**Ansicht Arzt Home Screen**



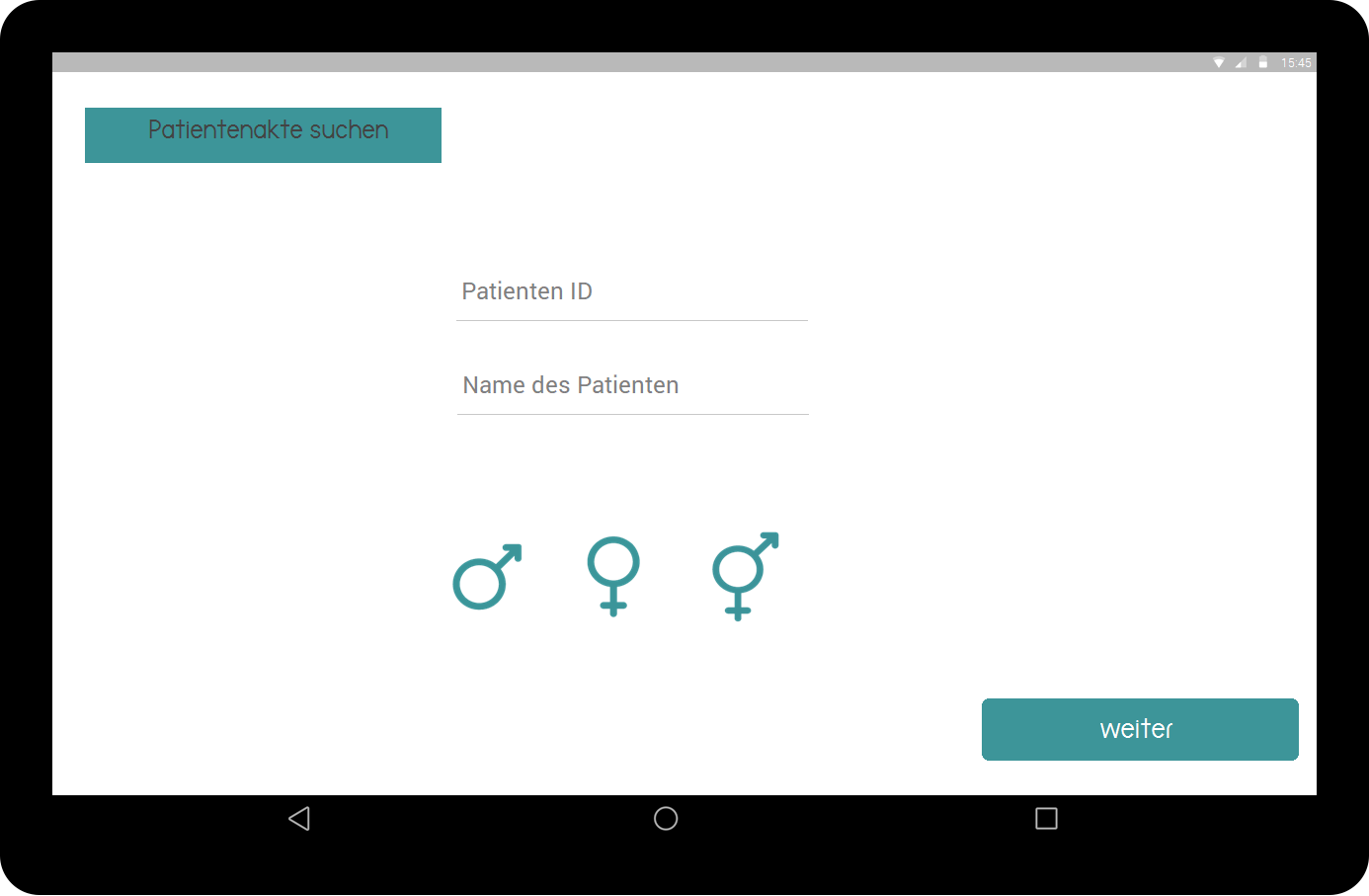
**Ansicht Patient anlegen**



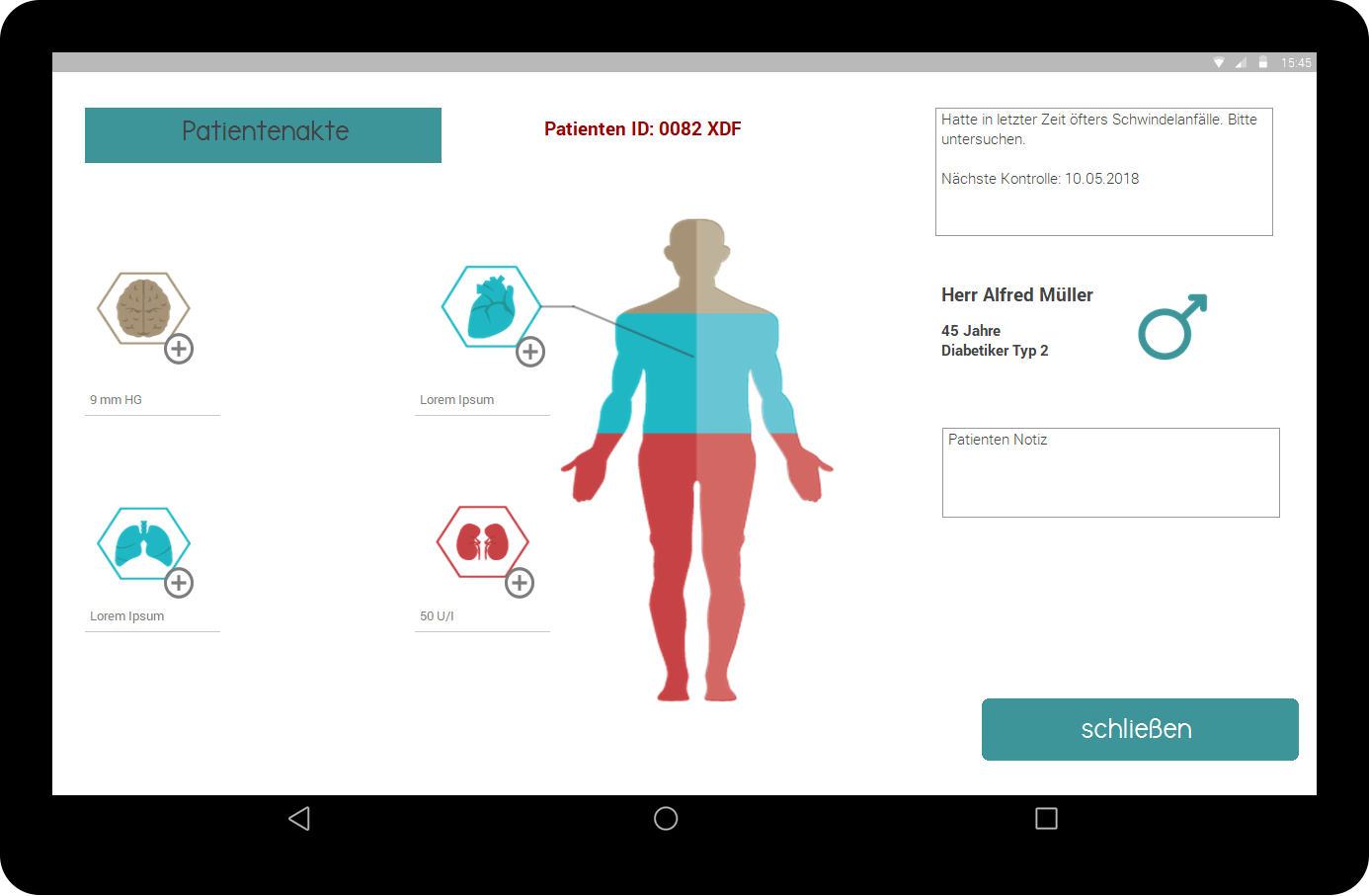
**Ansicht Patient anlegen Teil 2**



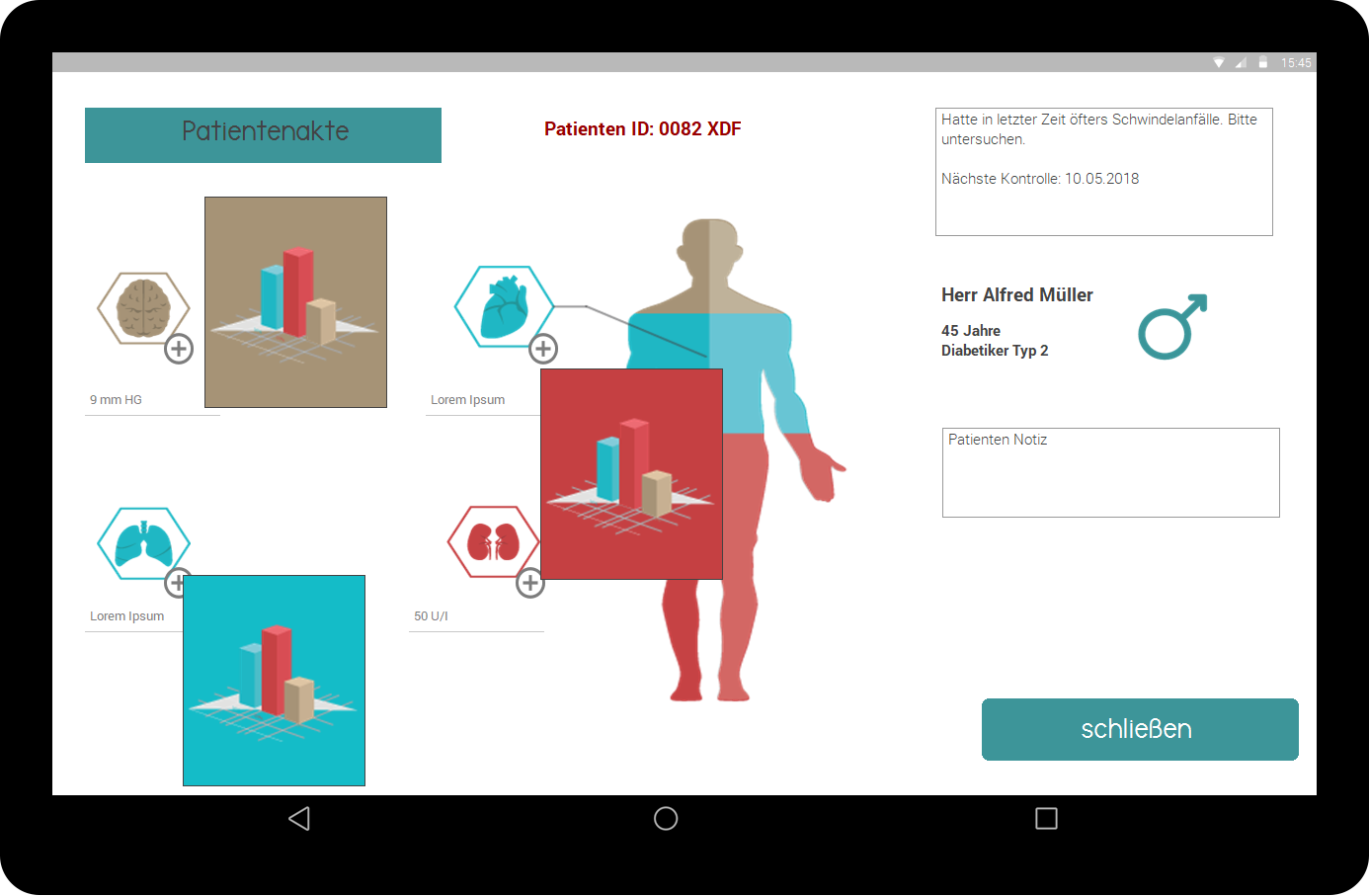
**Ansicht Patientenakte öffnen**



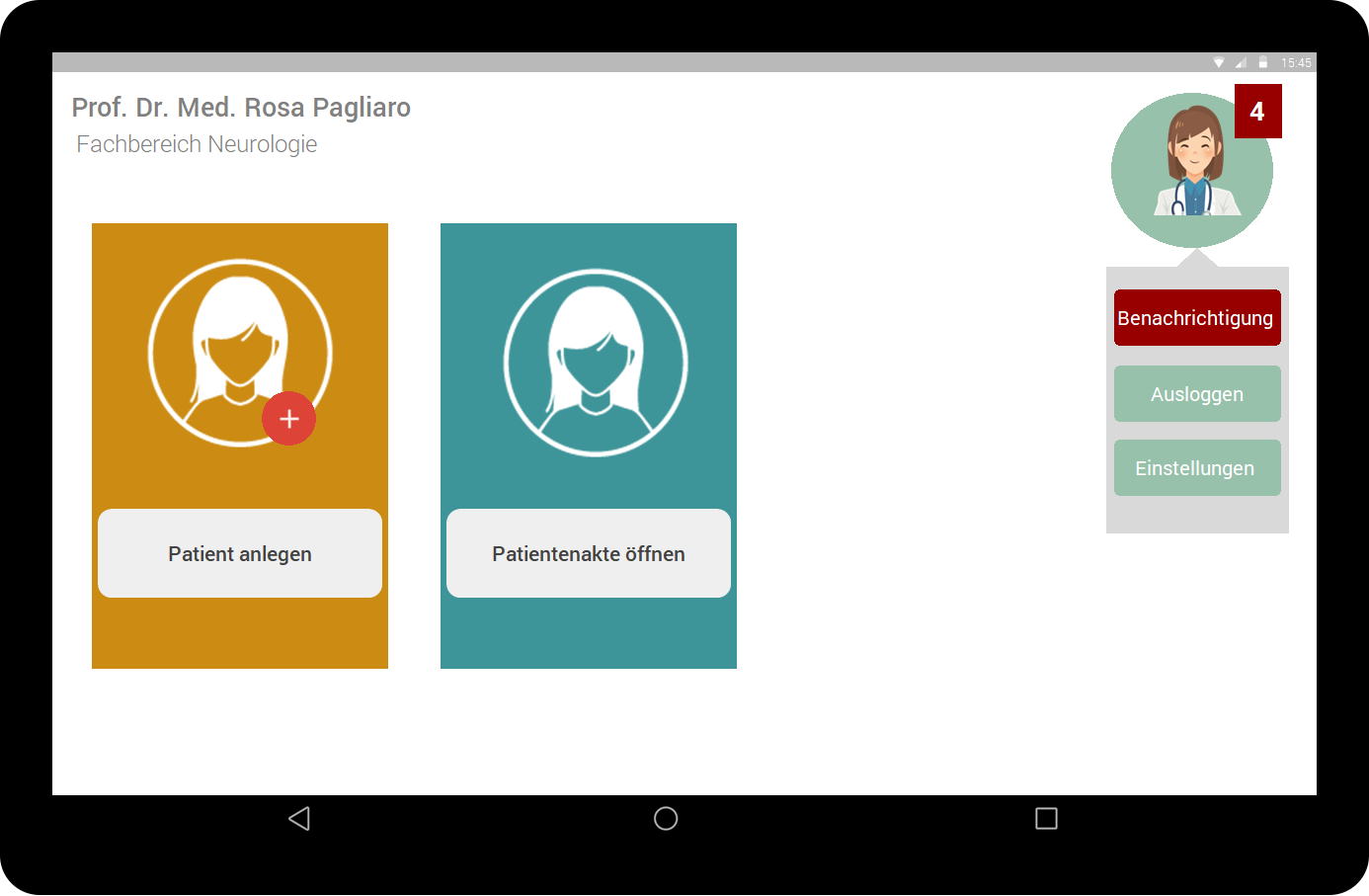
**Ansicht Patientenakte geöffnet**



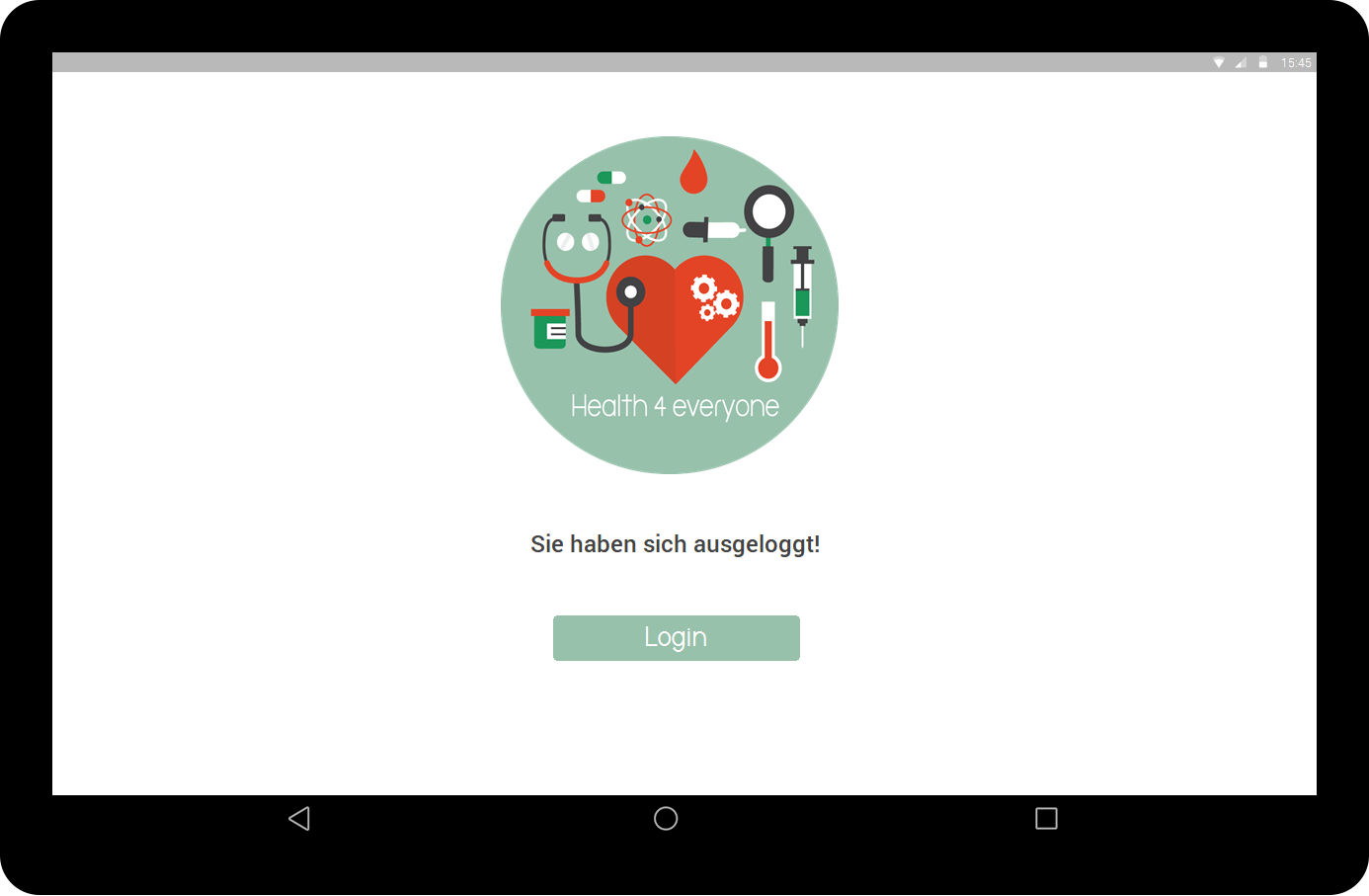
**Ansicht Patientenakte Statistiken anzeigen lassen**



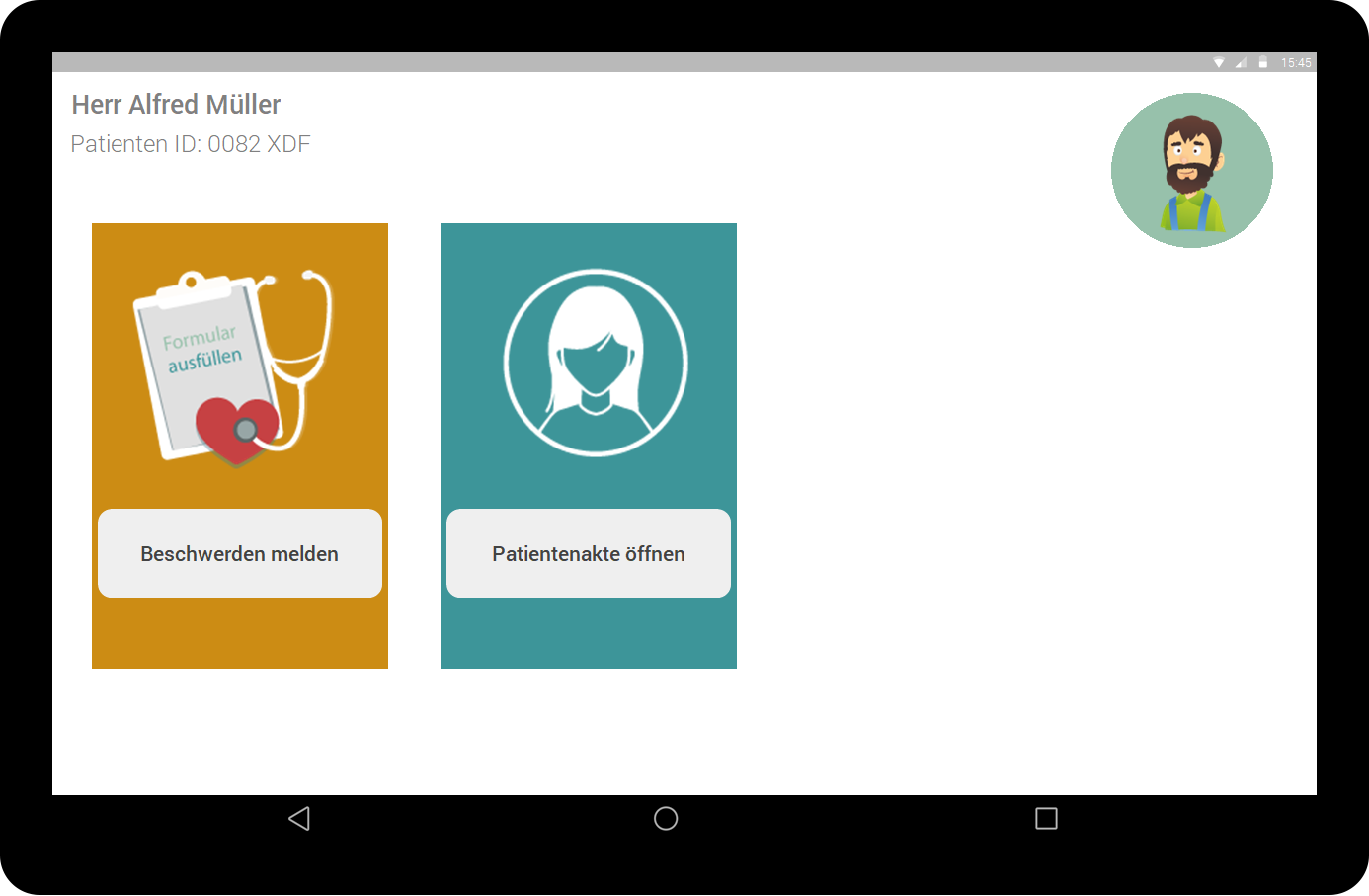
**Ansicht Ausloggen und Benachrichtigung Arzt Screen**



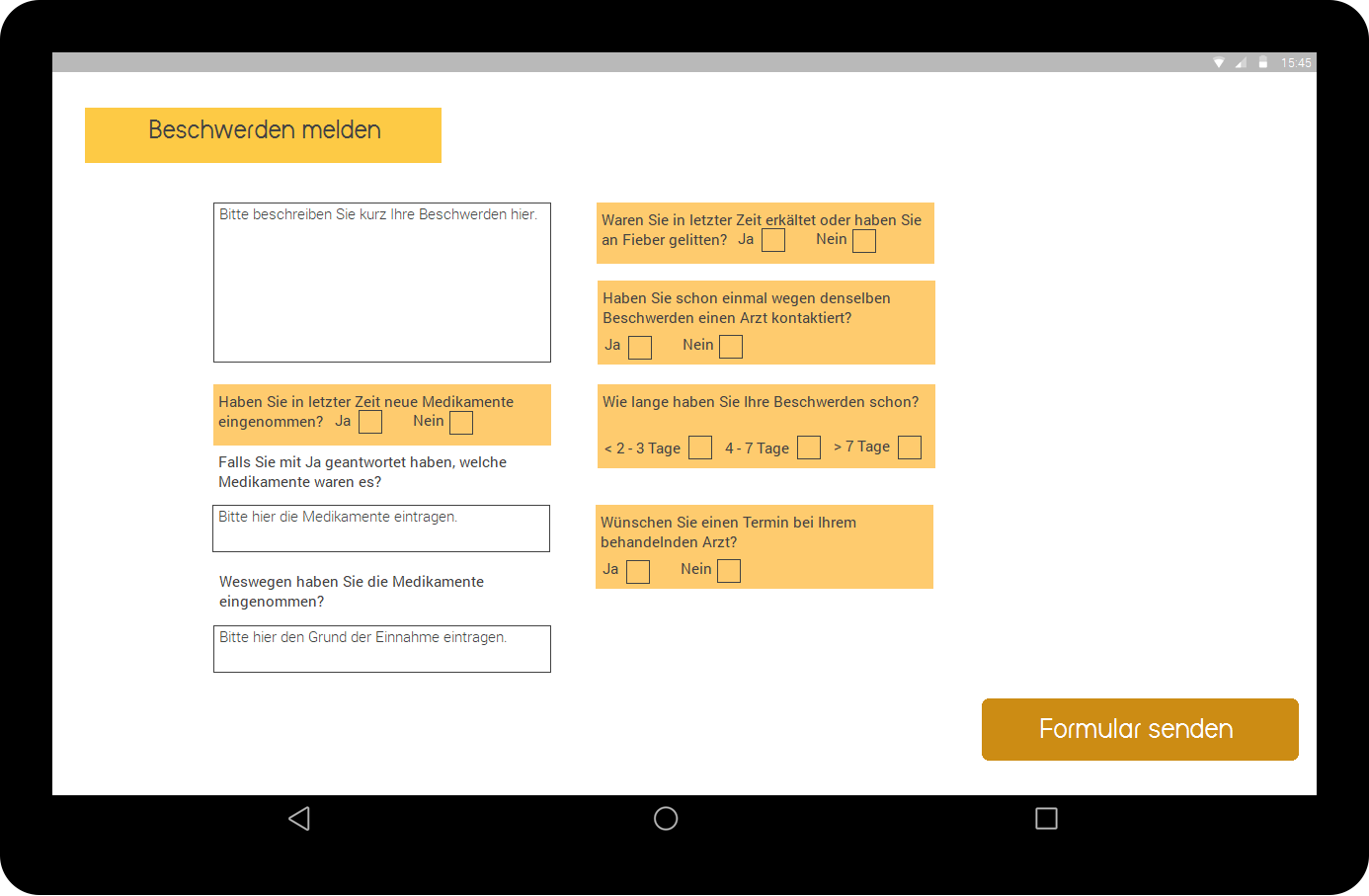
**Ansicht Sie haben sich ausgeloggt**



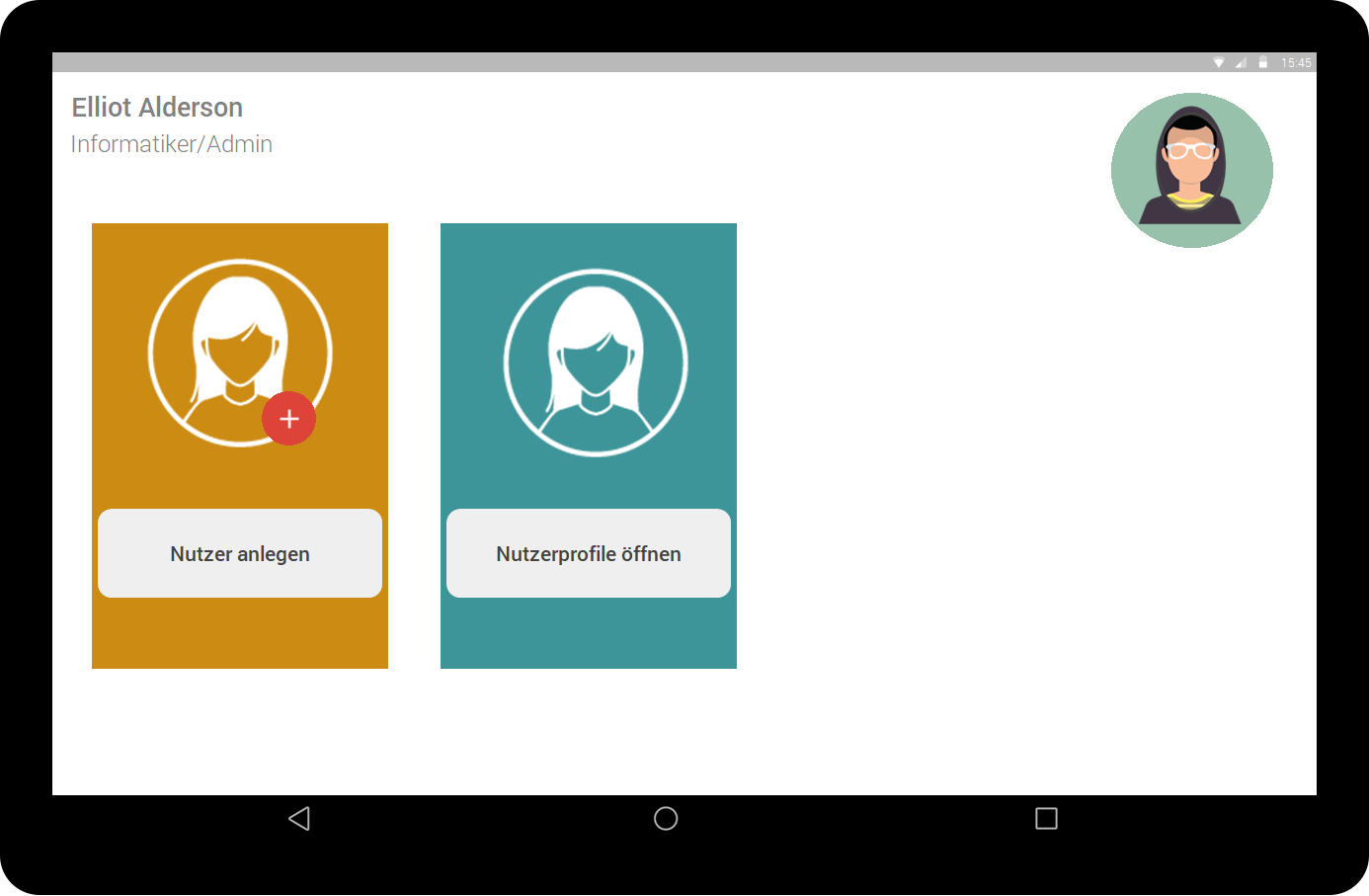
**Ansicht Patient Home Screen**



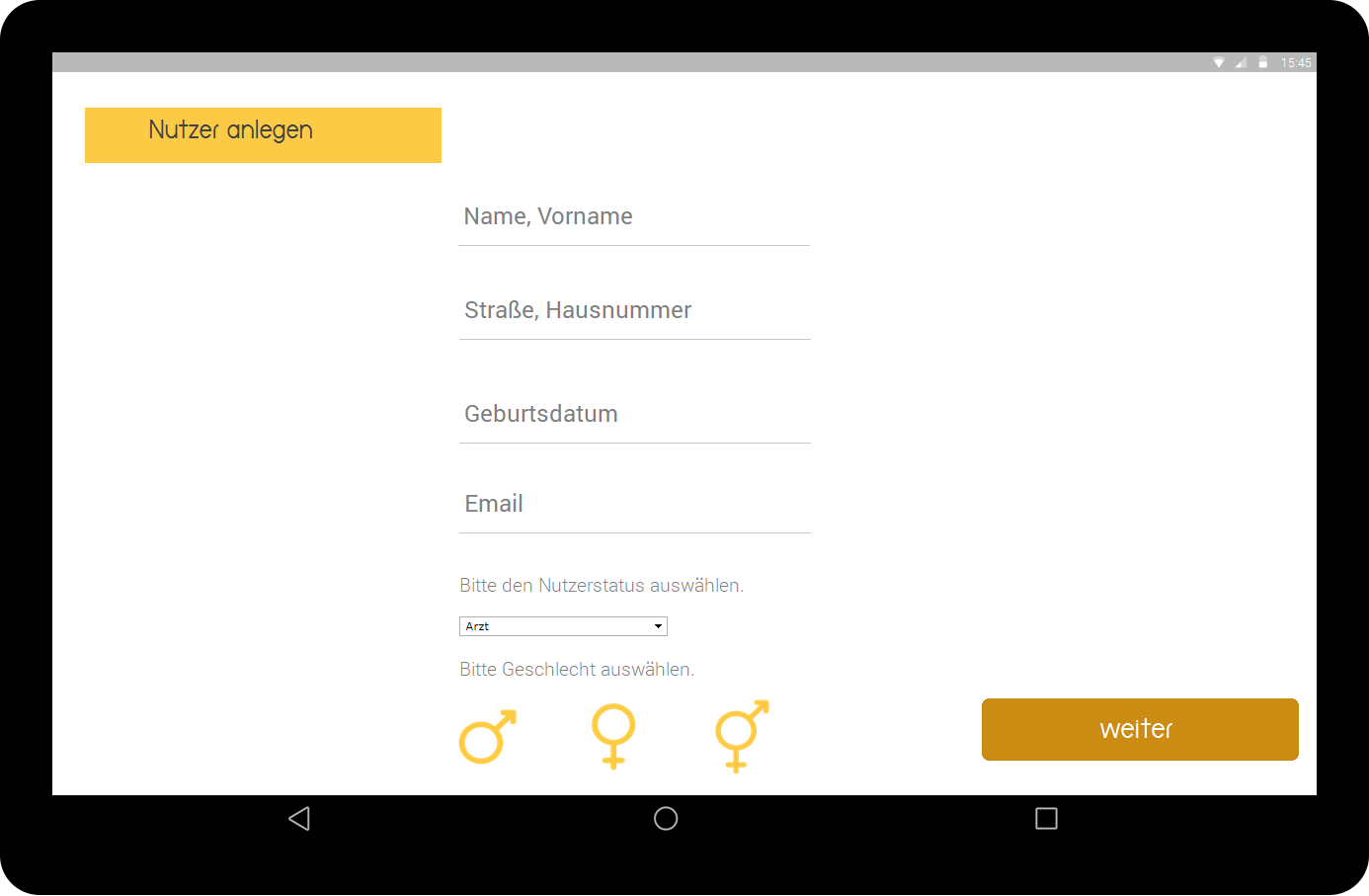
**Ansicht Beschwerden melden**



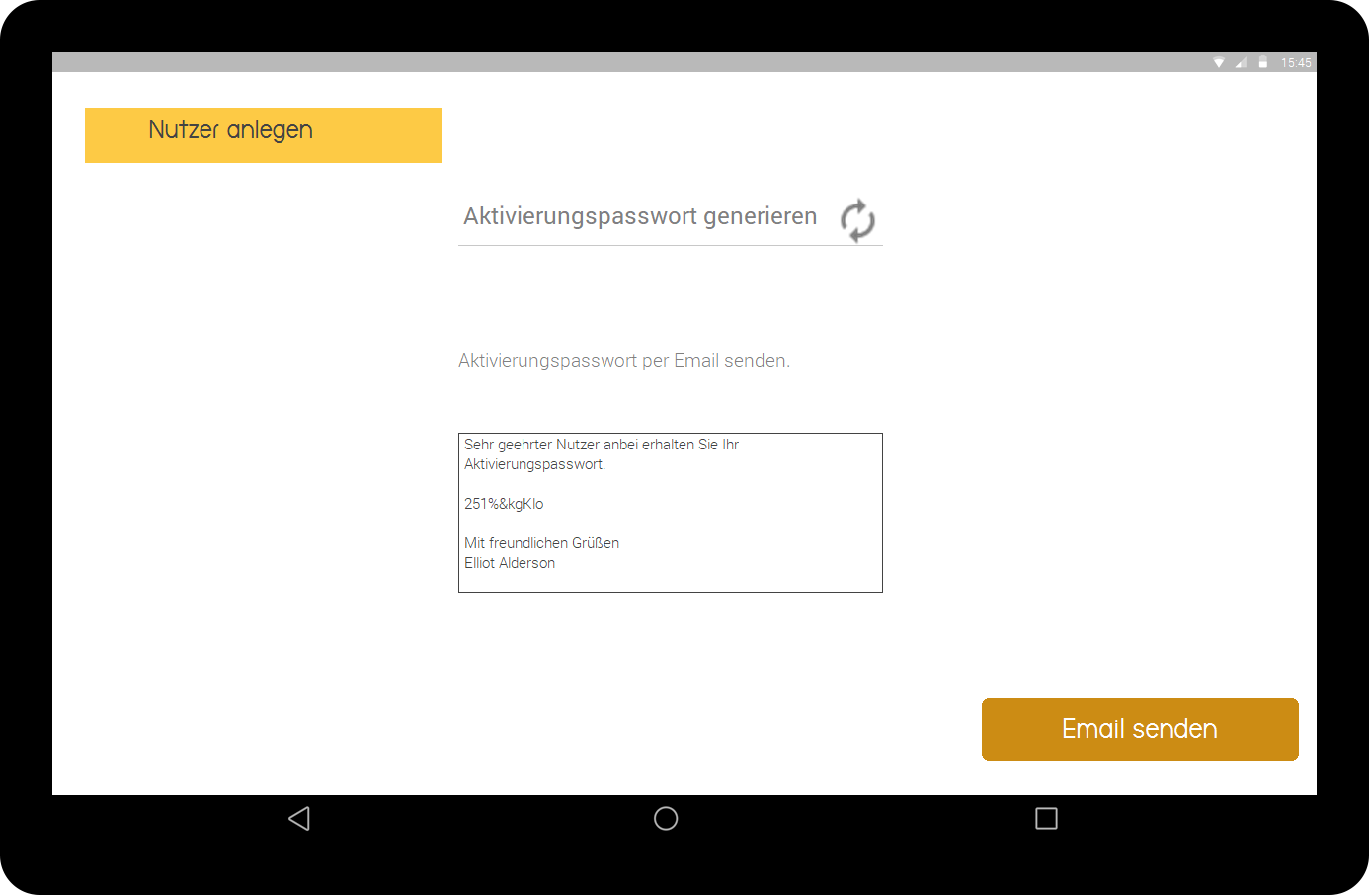
**Ansicht Informatiker Home Screen**



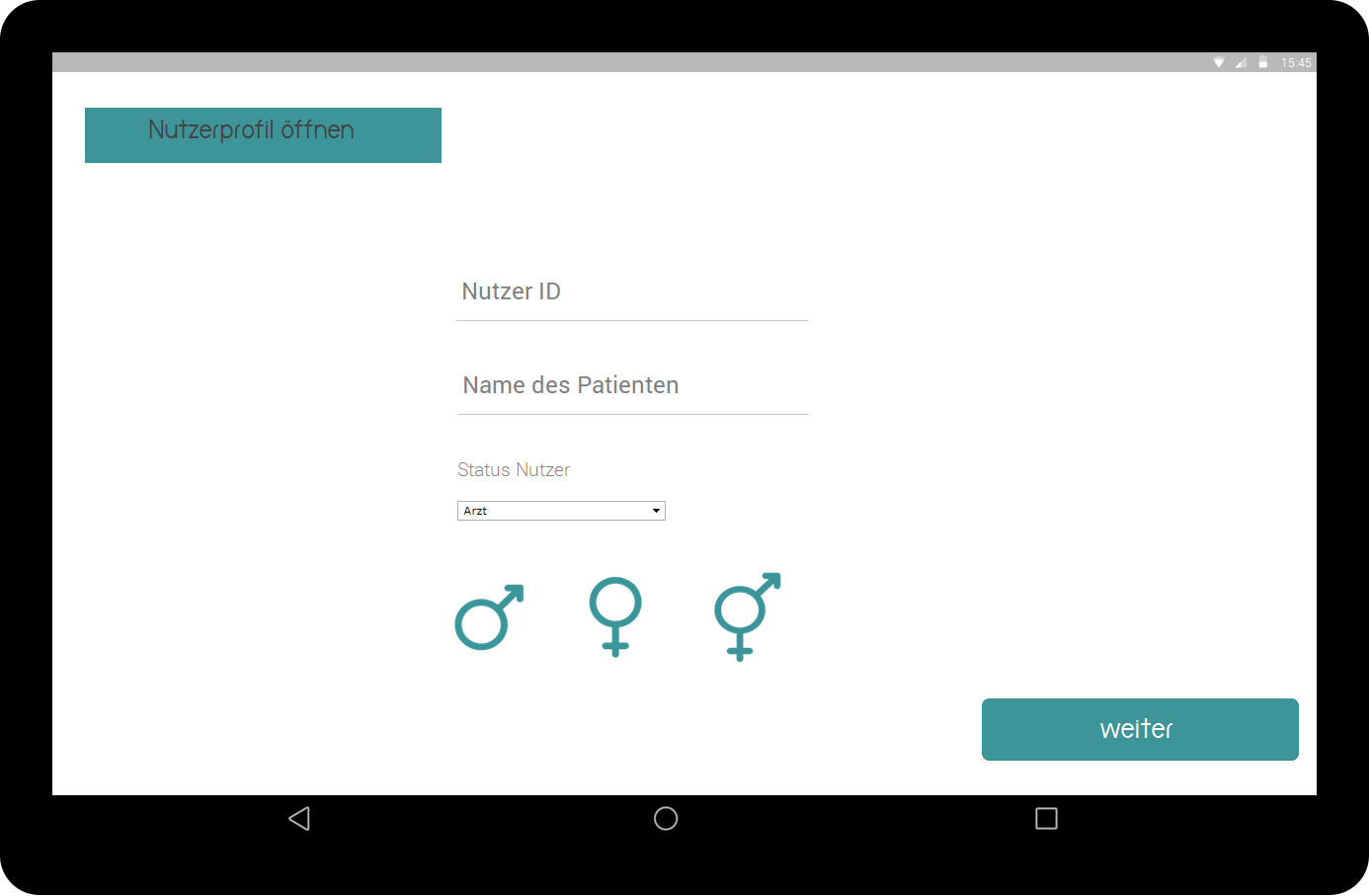
**Ansicht Nutzer anlegen (Nutzerstatus: Arzt, Patient und Informatiker)**



**Aktivierungspasswort generieren**



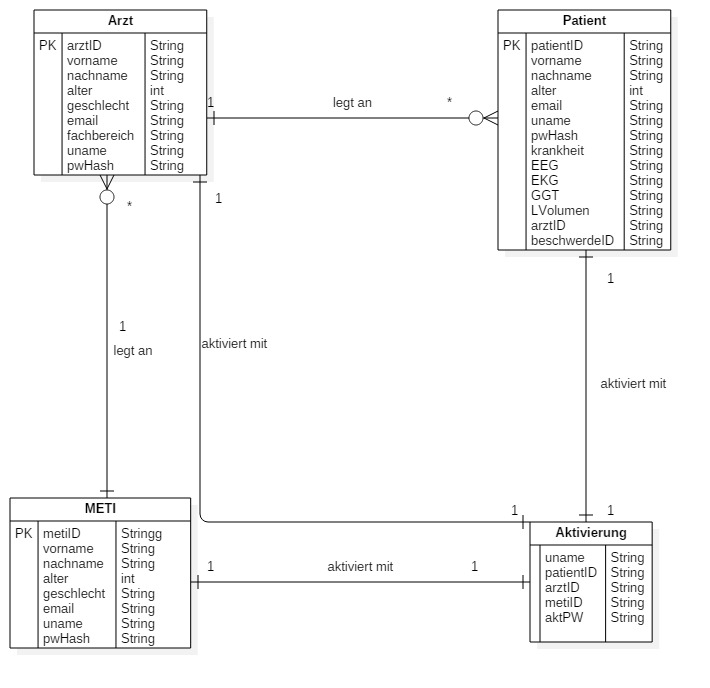
**Ansicht Nutzer Profil öffnen**



Das „Ausloggen“ ist bei allen drei Rollen identisch.

# Datenbank

**Entitäten:** Arzt, Patient, Meti und Aktivierung



# Server

**Module:**

* Express
* Express-Handlebars
* MongoDB
* Crypto
* Body-parser
* Express-session
* UUID
* UNIQUID

**Methoden:**

* get und post handler mit Express (**app.get()** und **app.post()** )
* **app.use()** zum Konfigurieren der benutzten Module
* **app.engine()** und **app.set()** zum Einstellen der view Engine
* **mongoClient.connect(path, function(err,db))** zum verbinden von MongoDB Server
* **db.collection()** mit verschiedenen Methoden zum Zugriff auf die Collection z. B. **db.collection(„meti“).find({})** zum finden eines Queries innerhalb der meti Collection
* **res.render()** zum Rendern eines Templates
* **res.redirect()** zum Weiterleiten auf ein anderes Dokument
* **db.close()** zum Schließen der Datenbankverbindung
* **.toArray()** zum Umwandeln in ein Array
* **res.sendFile()** zum Senden eines Dokuments an den User
* **app.listen()** zum Einstellen des Ports auf dem der Server Anfragen entgegennimmt

**Abläufe:**

METI

* Nutzer anlegen **->** Daten eingeben **->** weiter **->** Startseite METI
* Nutzerprofil öffnen **->** ID und Status eingeben**->** weiter **->** Ausgabe Nutzer
* Ausloggen **->** Logout Seite **->** (optional) Login Seite

Arzt

* Patient anlegen **->** Daten eingeben **->** weiter **->** Vitalparameter eingeben **->** bestätigen **->** Startseite Arzt
* Patientenakte öffnen **->** Daten eingeben **->** weiter –> Patientenakte **->** schließen  
  -**>**Startseite Arzt
* Ausloggen **->** Logout Seite **->** (optional) Login Seite

Patient

* Patientenakte öffnen **->** Patientenakte **->** schließen **->** Startseite Patient
* Ausloggen **->** Logout Seite **->** (optional) Login Seite

# Probleme

* Callbacks mit **async** und **await** Funktion vereinfacht.
* Bei der Patientenakte war am Anfang geplant, dass man auf die einzelnen Körperteile mithilfe von **onClick()** klickt und sich die Vitalparameter dazu öffnen. Jetzt sind die Vitalparameter von Anfang an sichtbar.
* CSS Einrückungen, Problem wurde mithilfe des Entwicklertools von Firefox behoben.
* Aus Zeitmangel wurden die Funktionen „Beschwerde melden“ und „Aktivierungspasswort senden“ nicht implementiert.

# Änderungen

* Auswahl der Geschlechter mit Dropdown Menüs geändert.
* In der Patientenakte werden keine statistischen Berechnungen der Gesundheitsdaten durchgeführt.
* METI Informatiker erstellt nur Arzt und METI, der Arzt erstellt den Patienten.

# Softwaretest

* HTML und CSS durch Entwicklertool des Browsers getestet und anschließend angepasst in den Code.
* Starten des nodeJS files mit der Konsole und dann mit Firefox testen

# Verwendete Software

* **UML Diagramme:** Star UML <http://staruml.io/>
* **Mockups:** Justinmind Prototyper <https://www.justinmind.com/>
* **IDE:** Webstorm <https://www.jetbrains.com/webstorm/>
* **Verwaltung:** Github <https://github.com/valeriapag/Health4Everyone.git>