



SEP – Wintersemester 2016/2017

Terminbuchungssystem für Ärzte

Lastenheft

Christian Bachmaier, Thomas Bock

## 1 Motivation

Die Reservierung eines Arzttermines kann durch ungünstige Öffnungszeiten, Nicht-Erreichbarkeit wegen zu hoher Auslastung oder Eintritt von unvorhergesehenen Verhinderungsgründen oftmals sehr langwierig und nervenaufreibend sein. Die Verwendung eines Online-Portals zur Buchung von Terminen bei einem Arzt stellt eine erhebliche Erleichterung für den Betreiber, aber auch für den Patienten dar, da dieser zum Beispiel nicht mehr an die Öffnungszeiten und einer freien Telefonleitung der jeweiligen Praxis gebunden ist. Um das Portal einfach betreiben und verwalten zu können, sollte die Anwendung auf eigenem Webspaces ausgeführt werden können.

## 2 Aufgabenstellung

Ziel des Software-Engineering Praktikums im Wintersemester 2016/2017 ist es, eine Web-Anwendung zur einfachen Terminreservierung bei einem Arzt/in einer Arztpraxis oder Ärztezentrum zu entwickeln. Die Anwendung sollte für den Betreiber der Praxis leicht einzurichten sein. Außerdem sollte dieser die Möglichkeit haben, gewisse Konfigurationen der Anwendung vornehmen zu können, wie zum Beispiel das Festlegen des

Logos, des Namens der Praxis oder der Öffnungs- und Urlaubszeiten. Da nicht angenommen werden kann, dass der/die Ärzte/Betreiber der Praxis das Portal auch ständig verwaltet, soll er die Möglichkeit haben, einen oder mehrere Praxishelfer/Manager festzulegen, die die Verwaltung der Termine übernehmen. Manager können beispielsweise Termine bearbeiten, bestätigen, uvm. Außerdem soll im System zwischen einem registrierten und einem anonymen Benutzer unterschieden werden. Der anonyme Benutzer sollte die Möglichkeit haben, sich im System zu bewegen, um zum Beispiel freie Termine einzusehen. Die eigentliche Reservierung sollte allerdings nur von einem registrierten (und angemeldeten) Benutzer durchführbar sein. Des Weiteren sollte der Betreiber die Möglichkeiten haben, Statistiken über die Auslastung der Praxis abzurufen, da diese Informationen für den Betreiber von großem Interesse sein können. Optional könnte die Anwendung auch für mehrere Arztpraxen entwickelt werden. Dies wäre für Ärzte/Betreiber vorteilhaft, die mehr als eine Arztpraxis betreiben.

Die zu erstellende Web-Anwendung soll intuitiv zu bedienen sein, geringe Anwendungshürden aufweisen, zugleich aber fortgeschrittenen Benutzern eine Vielzahl von erweiterten Funktionen mit einfacher Bedienung bieten. Außerdem wird empfohlen, eine fixe Dauer von Terminen im Viertelstundentakt zu wählen. Länger dauernde Termine können dann als Vielfaches von 15 Minuten nur von Managern festgelegt werden.

Beispiele für ähnliche und bereits existierende Systeme sind *samedi*<sup>1</sup>, *TimeControl*<sup>2</sup>, *doxter*<sup>3</sup> oder allgemeine Terminvereinbarungssysteme nicht speziell für Ärzte wie *Appointmentmind*<sup>4</sup> oder *Easy!Appointments*<sup>5</sup>. Diese Systeme gehen jedoch teilweise deutlich über die hier geforderte Funktionalitäten hinaus.

Die Anwendung muss mit Java 8 und JavaServer Faces (JSF) entwickelt werden. Auf Benutzerseite kann ein moderner Web-Browser vorausgesetzt werden.

### 3 Produkteinsatz

Zur Zielgruppe der zu entwickelnden Anwendung gehören als Betreiber niedergelassene Ärzte ggf. in Gemeinschaftspraxen und weniger (kein spezieller Fokus) Klinken mit unterschiedlich spezialisierten Ärzten. Der freie Zugang über das Internet soll für ein produktiv eingesetztes System möglich sein, sodass Patienten dieser Ärzte das System benutzen können.

## 4 Produktfunktionen

### 4.1 Begriffsdefinitionen

Es gibt mehrere Gruppen von Benutzern (= Akteuren) des Systems:

---

<sup>1</sup><http://www.samedi.de/>

<sup>2</sup><http://www.timecontrol.co/>

<sup>3</sup><http://www.doxter.de/>

<sup>4</sup><http://www.appointmind.de/>

<sup>5</sup><http://easyappointments.org/>

**Betreiber/Administrator:** Ist für die die Konfiguration des Systems verantwortlich. Dieser kann beispielsweise das Logo oder den Namen der Praxis festlegen. Zudem kann er weitere Einstellungen, wie die zu verwendende Datenbank oder ob Termine automatisch bestätigt werden, vornehmen. Eine sehr wichtige Aufgabe des Betreibers/Admins ist die Konfiguration der Öffnungs- und Urlaubszeiten und wieviele Termine parallel (beim Vorhandensein mehrerer "gleichwertiger" Ärzte) vergeben werden. Eine separate Terminverwaltung für jeden Arzt der Praxis ist optional. Der Betreiber/Admin ist außerdem in der Lage einen oder mehrere Manager der Praxis zu ernennen, die sich um die Verwaltung der Termine kümmern.

**Manager:** Ist für alle Verwaltungsaufgaben der Anwendung zuständig. Dieser soll in der Lage sein, Termine von Benutzern zu bearbeiten oder wieder freizugeben. Außerdem muss er in der Lage sein, selbst Termine für Patienten zu reservieren, z.B. falls eine persönliche oder telefonische Terminanfrage eingeht. Eine wichtige Funktion des Managers ist die Umsortierung/-organisation der reservierten Termine. Dadurch ist dieser z.B. in der Lage, die Auslastung der Ärzte zu optimieren oder größere freie Slots zu schaffen, um längere Termine einschieben zu können. Außerdem sollten von Managern größere Slots "bis auf Weiteres" blockiert/von der Terminvergabe ausgenommen werden können.

**Registrierter Benutzer:** Hat die Möglichkeit sich im System anzumelden und danach Termine in der Praxis zu reservieren. Hierfür muss sich dieser zuvor im System mit einer gültigen E-Mail-Adresse registrieren. Benutzer dürfen keine Daten (auch Namen) anderer Benutzer erfahren um der ärztlichen Schweigepflicht zu genügen.

**Anonymer Benutzer:** Hat lediglich die Möglichkeit, sich im System zu bewegen und beispielsweise freie Termine einzusehen.

## 4.2 Minimale Anforderungen

Im Folgenden ist mit Erstellen gleichzeitig auch nachträgliches Editieren und Löschen gemeint.

**Customizing** Administratoren/Betreiber sollen das System an die Praxis anpassen können, z.B. Look & Feel, Registrierungsmodalitäten von Benutzern (E-Mails müssen verifiziert werden), Kontaktmöglichkeiten, Impressum, usw. Des Weiteren muss der Admin/Betreiber die Möglichkeit haben, Öffnungszeiten, Urlaub und Feiertage zu spezifizieren. Optional können für mehrere Ärzte mehrere Terminpläne eingerichtet werden.

**Terminbuchung** Buchung eines Termins durch den angemeldeten Benutzer oder durch einen Manager (z.B. nach telefonischer Reservierung). Nur ein Manager kann längere Slots buchen und zu beliebigen Zeiten (auch aktuell blockierten oder außerhalb der Öffnungszeiten). Jegliche Termine müssen durch den Benutzer oder

durch den Manager wieder freigegeben werden können. In einer durch den Admin vorgegebenen Zeitspanne vor einem bevorstehenden Termin wird ein Patient dann automatisch durch eine E-Mail daran erinnert. Der Anlass des benötigten Termins kann angegeben werden.

**Öffnungszeiten** Öffnungszeiten, Feiertage und Betriebsurlaub können durch den Betreiber eingestellt werden. Blockierte Zeiten (auch) durch Manager. Falls hierbei Überschneidungen mit bereits bestehenden Reservierungen auftreten, müssen die betroffenen Patienten per E-Mail informiert werden.

**Präsentation der Praxis** Die Anwendung sollte die Möglichkeit bieten, die jeweilige Praxis zu präsentieren (Beschreibung, Bild, Kontaktmöglichkeiten, Spezialgebiete, etc.).

**Betreibernachricht** Beim Buchen eines Termins kann eine Nachricht an die Praxis gesendet werden, z.B. dass man gehbehindert ist.

**Benachrichtigung** Bei Änderungen seitens des Betreibers, oder pauschal an bestimmte oder alle Patienten mit Terminvereinbarungen in einem Zeitraum.

**Umorganisation** Der Betreiber oder ein Manager soll Termine umorganisieren können mit jeweils angebotener Benachrichtigung an die betroffenen Patienten. Ein dazu vorformulierter Text soll dabei jeweils änderbar sein.

**Registrierung** Ein anonymes Benutzerkonto kann sich ein Benutzerkonto anlegen. Die Verifikation eines Benutzers kann entweder durch E-Mailverifikation und/oder durch die Bestätigung des Betreibers oder Managers (einstellbar) erfolgen.

**Statistiken** Dem Betreiber soll die Möglichkeit geboten werden, sich Statistiken über die Praxis anzeigen zu lassen. Zum einen soll es möglich sein, die Auslastung der Praxis zu bestimmten Zeiten darzustellen. Zum anderen sollen Patienten mit häufigen Terminen ermittelt werden können.

**Reservierungshilfe** Falls zum gewünschten Zeitpunkt kein passender Termin für einen Patienten zur Verfügung steht, soll diesem, wenn möglich, ein Ersatztermin zu einem früheren oder späteren Zeitpunkt angezeigt werden (max. 2 Std. vor bzw. 2 Std. nach gewünschtem Zeitpunkt). Falls kein Vorschlag ermittelt werden kann, besteht für den Patienten die Wahl, einen/die Manager über das System darüber inkl. Anlass des Termins zu benachrichtigen. Dieser kann dann selbst überprüfen ob eine alternativer Termin besteht. Der Manager kann anschließend den jeweiligen Patienten informieren.

**Benutzerverwaltung** Im System gibt es verschiedene Benutzerrollen: Betreiber, Manager, registrierte Nutzer und anonyme Nutzer. Der Betreiber kann alle Benutzer und die damit verbundenen Rollen verwalten: Anlegen neuer Benutzer, Bearbeiten/Löschen existierender Benutzer, Zuordnen/Entziehen von Rollen. Beispielsweise werden Manager durch einen Betreiber ernannt. Eine Funktion zum Suchen nach bestimmten Benutzern erleichtert einem Betreiber die Verwaltung.

**Anmeldung** Ein Benutzer kann sich in ein Benutzerkonto einloggen und ist damit angemeldet. Hierfür ist eine Authentifizierung notwendig.

**Kontoverwaltung** Ein angemeldeter Benutzer kann sein Benutzerkonto editieren oder löschen. Zu den Kontodaten gehören Name, Geburtsdatum, E-Mailadresse und Passwort. Krankengeschichte, Versicherungsdaten, etc. sind jedoch nicht im Primärfokus des Projekts.

**Abmeldung** Ein angemeldeter Benutzer kann sich abmelden.

**Online-Hilfe und FAQ** Der Benutzer muss zu jedem Zeitpunkt schnellen Zugriff auf die für die aktuelle Seite relevante Online-Hilfe und die optional vorhandene FAQ haben.

## 5 Nicht-Funktionale Anforderungen

Folgend die Produktleistungen und Qualitätsanforderungen.

### 5.1 Usability

1. Einfache und intuitive Bedienbarkeit des Systems.
2. Die Benutzeroberfläche soll sich an allgemein geläufigen Bedienkonzepten und den damit verbunden Funktionen orientieren; alle Tabellen sollen z.B. nach den dargestellten Spalten sortierbar sein; Tabellen, die eine gewisse Größe überschreiten, sollen, um kurze Seitenladezeiten zu ermöglichen, auf mehrere Seiten aufgeteilt werden (Pagination).
3. Häufig wiederkehrende Aufgaben sind durch das System möglichst benutzerfreundlich zu unterstützen; häufig genutzte Funktionen sind möglichst einfach zugänglich zu machen.
4. Die Seiten der Applikation sind übersichtlich und einfach verständlich zu gestalten.
5. Daten sollen nicht nur leicht auffindbar und gut lesbar sein, sondern auch leicht einzugeben.
6. Bei Fehleingaben in ein HTML-Formular und der darauf folgenden Korrektur sollen die zuvor eingetragenen Felder nicht erneut einzugeben sein, sondern schon vorbesetzt sein. Außerdem sollte die Überprüfung der Eingaben nicht nach der ersten fehlerhaften Eingabe abbrechen, sondern alle Eingaben überprüfen und eine akkumulierte Fehlermeldung an den Benutzer zurückgeben.
7. Das System soll mit steigender Last skalieren.

## 5.2 Datensicherheit

1. Alle im System erfassten Daten sind persistent in einer Datenbank abzulegen; die Konsistenz der Daten ist sicherzustellen (Mehrbenutzerbetrieb!). Speziell, wenn Änderungen über mehrere Datenbanktabellen hinweg vorgenommen werden, sind Transaktionen zu nutzen.
2. Für die persistente Speicherung der Daten soll eine Datenbank (PostgreSQL) verwendet werden, die auf den Infosun-Rechnern zur Verfügung gestellt wird. Als Referenzplattform für die Implementierung dienen die Rechner im CIP-Pool.
3. Beim Löschen von Daten sind die Abhängigkeiten zwischen Daten einzelner Tabellen zu berücksichtigen; hat das Löschen eines Datensatzes das Löschen anderer Datensätze zur Folge, muss der Benutzer vorher deutlich darauf hingewiesen werden!

## 5.3 Datenschutz

1. Es muss sichergestellt werden, dass durch das System zu keinem Zeitpunkt sensible Daten für unberechtigte Dritte zugänglich sind.
2. Es sollen möglichst wenige technische Informationen über das System nach außen gegeben werden.
3. Alle personenbezogenen Daten, wie z.B. Login-Daten, sind sensibel und dürfen daher nur per SSL-Verbindung übertragen werden.
4. Passwörter dürfen nicht im Klartext gespeichert werden.
5. Die Nutzerdaten sind so zu speichern, dass kein unautorisierter Zugriff durch Dritte oder durch andere Nutzer im System stattfinden kann.
6. Das System darf nur im Rahmen der oben genannten oder vorgesehenen Funktionalitäten verändert werden.
7. Änderungen durch Manipulationen mit bekannten Angriffsmethoden wie SQL-Injection oder Cross-Site-Scripting müssen ausgeschlossen werden. Es müssen außerdem Maßnahmen ergriffen werden, um die Sitzungen der einzelnen Nutzer zu schützen (Session-Hijacking).

## 5.4 Internationalisierbarkeit

1. Für die Texte auf der Website ist die Zeichenkodierung UTF-8 zu wählen.
2. Die Sprache des Systems kann Deutsch oder Englisch sein; eine mehrsprachige Implementierung ist optional.

## 5.5 Weiterentwickelbarkeit

Das zu erstellende System soll flexibel gegenüber zukünftigen Erweiterungen sein. Eine einfache und kostengünstige Weiterentwickelbarkeit des Systems ist sicherzustellen.

## 5.6 Installation

Es soll eine komfortable Installation für Systembetreiber bereitgestellt werden. Die Installation sollte einfach und schnell sein und automatisch das entsprechende Datenbank-Setup vornehmen.

## 6 Ergänzungen

Die Benutzung des Systems sollte mit allen gängigen Web-Browsern möglich sein. Wir raten allerdings davon ab, für jeden Browsertyp unterschiedlichen HTML-Code zu generieren. Der HTML-Code soll logisches Markup darstellen und nicht dazu missbraucht werden, eine bestimmte graphische Darstellung zu erzwingen. Deswegen dürfen Features wie Frames nicht und explizites Javascript nur nach Absprache eingesetzt werden. Der HTML-Code muss HTML-konform sein und z.B. durch <http://validator.w3.org/> validierbar sein. Die Verwendung von Cascading Stylesheets (CSS) wird dringend angeraten.

Die Sessionverwaltung darf die Verwendung von Cookies nicht erzwingen.

Das System muss ein Log über alle Fehler führen, um das Debugging und den Betrieb der Anwendung zu vereinfachen. Achten Sie darauf, dass die Fehlerbeschreibungen detailliert genug sind, um auf einen Fehler bzw. dessen Ursache schließen zu können.

## 7 Leistungsnachweis

Mit der funktionalen Umsetzung o.g. Musskriterien kann die Note 1,0 erreicht werden. Wunschkriterien sind (wie der Name sagt) wünschenswert, aber deren Umsetzung ist nicht zwingend notwendig für das Erreichen der vollen Punktzahl.