C868 – Software Capstone Project Summary

Task 2 – Section C

Capstone Project Name: MedBookr – A Medical Appointment Tracker

Student Name:

Table of Contents

Table of Contents 2

Application Design and Testing 5

Design Document 5

Class Design 5

UI Design 6

Test Plan 8

Introduction 8

Purpose 8

Overview 9

Test Plan 9

Items 9

Functions/Features 10

Deliverables/Outcomes 10

Tasks 10

Needs 11

Pass/Fail Criteria 11

Specifications 11

Procedures 12

Results 13

Source Code 14

Link to Live Version 14

User Guide 14

Set Up and Run Application for Maintenance Purposes 14

Prerequisites 14

Installation 14

Login with an Admin Account 15

Viewing and Managing Doctors 15

Viewing Doctors 15

Creating a New Doctor 17

Updating Doctor Information 18

Removing a Doctor 19

Viewing and Managing Patients 20

Viewing Patients 20

Creating a New Patient 21

Updating Patient Information 22

Removing a Patient 22

Viewing and Managing Appointments 23

Viewing Appointments by Month and Week 23

Creating New Appointments 24

Updating Appointments 27

Deleting an Appointment 27

Viewing Appointment Reports 28

Appointments by Date Range 28

Appointments by Time of Day 29

Appointments by Doctor 29

Application Design and Testing

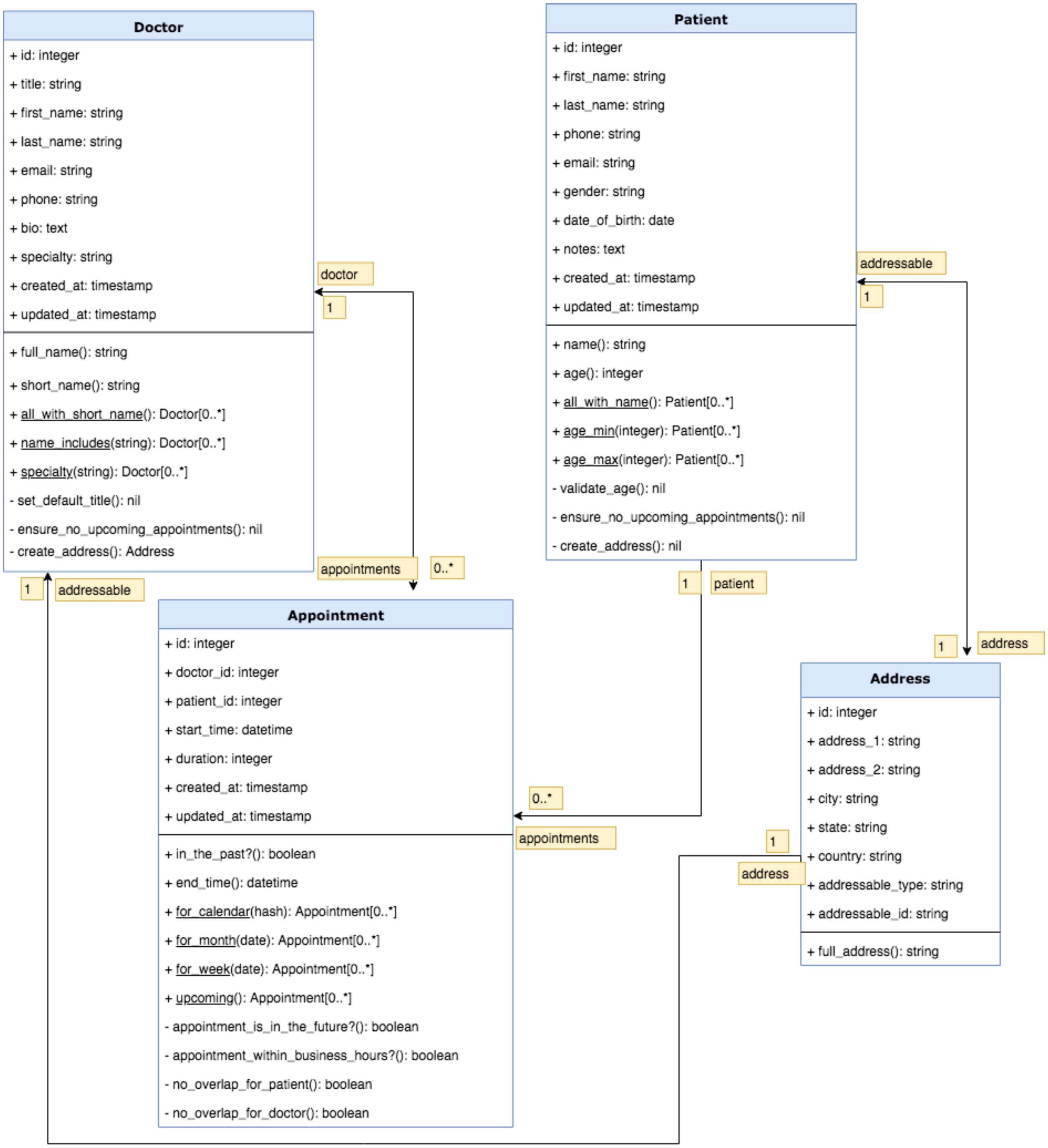
Design Document

Class Design

The figure below depicts the class design for the MedBookr web application. Since the Ruby on Rails framework used for this project embraces the MVC (Model View Controller) architecture, I've chosen to show the model classes as they directly map to the database objects that form the core of the application.

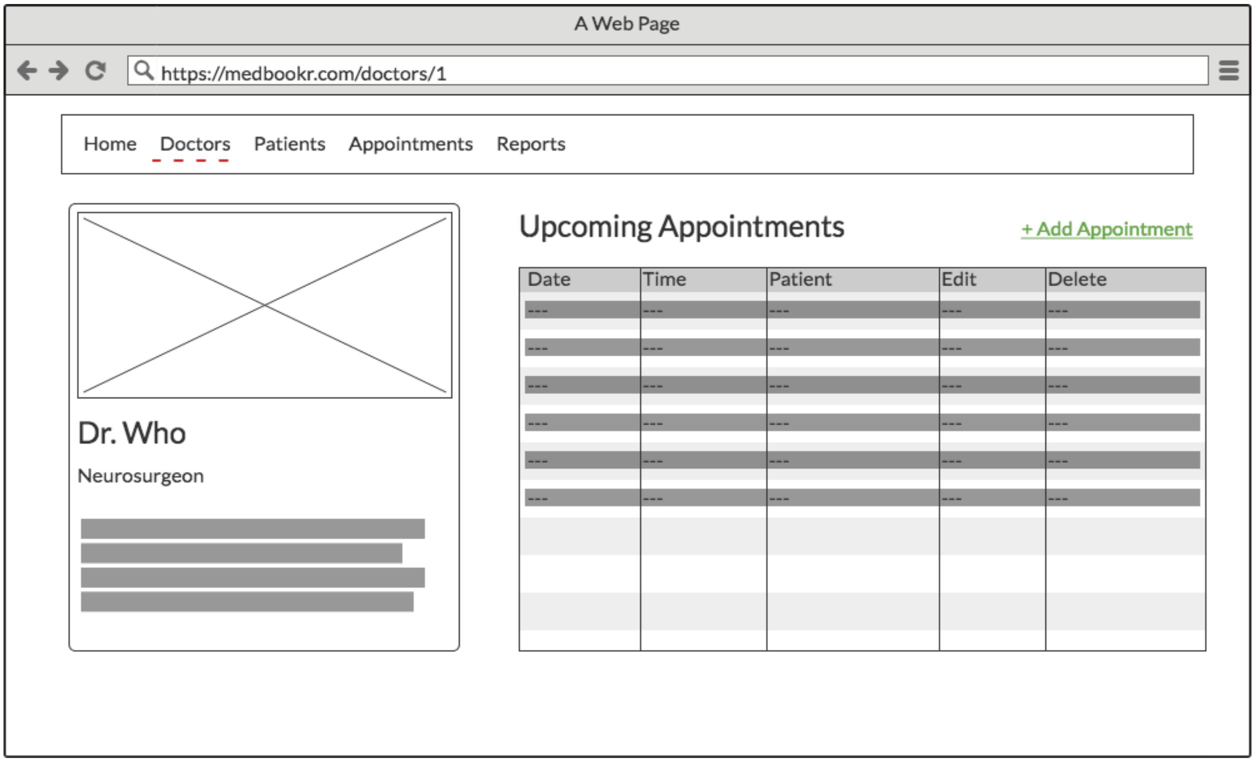
The three main model classes are Doctor, Patient, and Appointment. There is a one-to-many relationship between Doctor and Appointment as well as Patient and Appointment. Conversely, there is a necessary many-to-one relationship between Appointment and Doctor (an appointment has to belong to one and only one doctor) and Appointment and Patient (an appointment has to belong to one and only one patient). A polymorphic one-to-one relationship exists between Doctor/Patient and Address. In Rails, the Address model has two attributes to indicate its polymorphic nature, namely `addressable_id` (holding the id of either a doctor or a patient) and `addressable_type` (for the name of the class, i.e. Doctor or Patient). Not depicted is the User class used to create admin accounts.

Class methods are indicated with underlines. An example is the Patient's `age_min(integer)` class method that returns an array of patients with a minimum age specified by the argument to the method. Private methods are denoted with a minus sign (-). For instance, the Appointment's private validation method `no_overlap_for_doctor()` checks that there are no other appointments for the same doctor that would overlap with the current one.



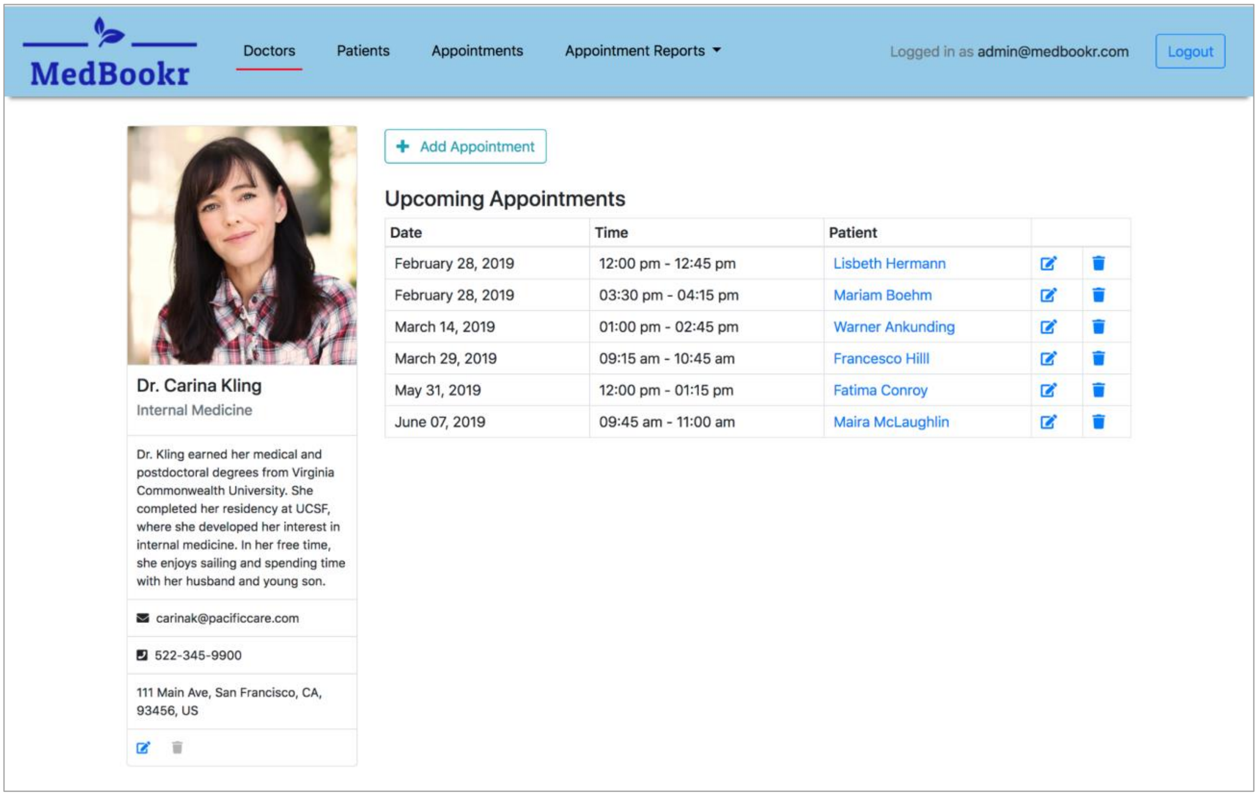
UI Design

Below are the low and high-fidelity versions of the detail page for a particular doctor. The left side displays information such as the doctor’s full name, specialty, and contact info and shows an optional profile image. If a bio is available, it is displayed above the contact details. On the right side, there is a table with the upcoming appointments for the doctor, ordered by date and time. The name and a link to the patient is provided along with links to edit or delete the appointment. Above the table there is a prominent link for creating a new appointment for the doctor. When clicked, it will trigger a modal popup window containing a form to select the patient and set a date and time for the new appointment.



Date	Time	Patient	Edit	Delete
---	---	---	---	---
---	---	---	---	---
---	---	---	---	---
---	---	---	---	---
---	---	---	---	---
---	---	---	---	---

Low fidelity version of doctor detail page



Upcoming Appointments

Date	Time	Patient		
February 28, 2019	12:00 pm - 12:45 pm	Lisbeth Hermann		
February 28, 2019	03:30 pm - 04:15 pm	Mariam Boehm		
March 14, 2019	01:00 pm - 02:45 pm	Warner Ankunding		
March 29, 2019	09:15 am - 10:45 am	Francesco Hilll		
May 31, 2019	12:00 pm - 01:15 pm	Fatima Conroy		
June 07, 2019	09:45 am - 11:00 am	Maira McLaughlin		

Dr. Carina Kling
Internal Medicine

Dr. Kling earned her medical and postdoctoral degrees from Virginia Commonwealth University. She completed her residency at UCSF, where she developed her interest in internal medicine. In her free time, she enjoys sailing and spending time with her husband and young son.

✉ carinak@pacificcare.com

☎ 522-345-9900

111 Main Ave, San Francisco, CA, 93456, US

High-fidelity version of doctor detail page

Test Plan

Introduction

Purpose

In order to test the validations of the Appointment class, I wrote a number of unit tests using RSpec, a Ruby testing library. Since the appointment class is central to the application, I wanted to ensure that an invalid appointment cannot be saved to the database. There are many ways an appointment can be invalid but here I'm focusing on making sure that a doctor cannot be double booked. Validations in Rails are set up to run automatically whenever a record is created or updated. This unit tests verifies that the appropriate validation method is in place to prevent an appointment record from being persisted if it overlaps another one with the same doctor.

Overview

Since at its core the application is an appointment tracker, the logic around creating and saving appointments needed to be carefully considered. Validations are methods that ensure that a record is in a correct state according to certain pre-defined validation rules. They are invoked just before a record is saved and if any validation rules were violated, the record will have certain error properties and will not be persisted to the database. This test attempts to ensure that the validation rule responsible for preventing a doctor to be double booked does indeed stop erroneous appointments from getting into the database. Other validation tests are in place for appointments, and there is an equivalent validation test for preventing a patient to be double booked.

The unit test is written using the RSpec DSL (Domain Specific Language) and executed on the command line with the `rspec` command. RSpec prints success and failure messages on the terminal that indicate whether or not the test was successful.

Test Plan

Items

To test that an overlapping appointment with the same doctor cannot be created, we need three model instances: one doctor and two appointments. Both appointments will be associated with the same doctor. The first one will be given a certain start time and a duration and be saved to the database. A second appointment instance will need to be created and given a start time that overlaps the time range of the first appointment. We can then prove through a test that the second appointment is in an error state and won't be persisted to the database.

A second test is needed to demonstrate that two appointments that do not overlap *can* be created for the same doctor. For this purpose, we set the start time of the second appointment to have no overlap with the first one and make sure that it does get saved successfully.

To test edge cases, we can create second appointments that are adjacent but do not overlap the first one. It is expected that those appointments are valid and do get saved to the database.

Functions/Features

To set up the validation for overlapping appointments for the Appointment class, I invoked the `validate` class method with the name of the private instance method that contains the validation logic: `validate :no_overlap_for_doctor.` The `validate` method is inherited from `ApplicationRecord`, the base class that Rails' models derive from.

To test the validation, we need to invoke certain methods on the Appointment class and appointment instances. The following methods of the Appointment class will be exercised:

- Instance setter methods for `start_time`, `duration` and `doctor`
- Instance method `valid?` that checks for general validity
- Instance method `errors` that returns an array of error objects
- The `create` class method that instantiates a new appointment object and attempts to save it to the DB at the same time
- The `new` and `build` class methods that just instantiate a new appointment in memory without saving it to the DB

Deliverables/Outcomes

When the test is run, RSpec produces structured console output. Each test is given a descriptive name that indicates the exact behavior under scrutiny, e.g. "should not allow overlaps". The full line in the test output will be "Appointment validation for other appointments of doctor should not allow overlaps." The other parts of the output sentence reflect the context of the test in question.

Tasks

To execute this unit test the following steps are required:

1. Write the code to be tested, i.e. set up the validation.

2. Create an RSpec file in the spec directory of the application to house the test code.
3. Use the RSpec DSL to create the nested context levels all the way down to the actual test.
4. Write the test code using RSpec methods, and methods from other helper libraries.
5. Run the test using the `rspec` command followed by the path to the test file.
6. Examine the output to determine pass or fail.

Needs

The described setup has a variety of dependencies. In Ruby packaged libraries are called “gems” and the following gems were used here:

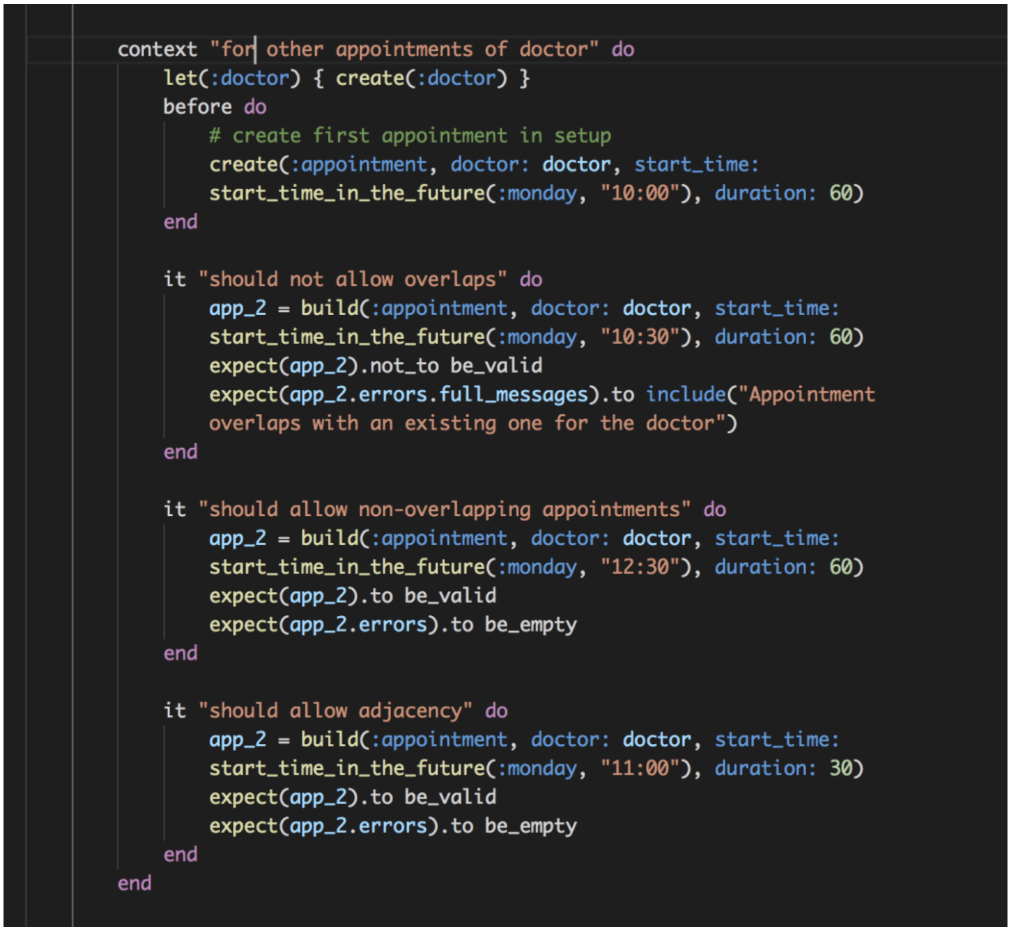
- `rspec 3.8.0` (RSpec test library for Ruby)
- `rspec-rails 3.8.2` (Rspec integration library for the Ruby on Rails framework)
- `factory_bot_rails 5.0.1` (Library for creating model instances based on present values)

Pass/Fail Criteria

The criteria for a successful validation of an overlapping appointment instance is that its `valid?` method returns false and its `errors` method returns a non-empty array containing the appropriate error message. A failing test on the other hand would not be able to confirm these two criteria.

Specifications

This is a screenshot of the test code described above. It is an excerpt from the main test class and utilizes several helper methods. The test file is located at `spec/models/appointment_spec.rb` in the root directory of the application.



```
context "for other appointments of doctor" do
    let(:doctor) { create(:doctor) }
    before do
        # create first appointment in setup
        create(:appointment, doctor: doctor, start_time:
        start_time_in_the_future(:monday, "10:00"), duration: 60)
    end

    it "should not allow overlaps" do
        app_2 = build(:appointment, doctor: doctor, start_time:
        start_time_in_the_future(:monday, "10:30"), duration: 60)
        expect(app_2).not_to be_valid
        expect(app_2.errors.full_messages).to include("Appointment
        overlaps with an existing one for the doctor")
    end

    it "should allow non-overlapping appointments" do
        app_2 = build(:appointment, doctor: doctor, start_time:
        start_time_in_the_future(:monday, "12:30"), duration: 60)
        expect(app_2).to be_valid
        expect(app_2.errors).to be_empty
    end

    it "should allow adjacency" do
        app_2 = build(:appointment, doctor: doctor, start_time:
        start_time_in_the_future(:monday, "11:00"), duration: 30)
        expect(app_2).to be_valid
        expect(app_2.errors).to be_empty
    end
end
```

Procedures

As seen in the code excerpt above, there are three separate tests that verify the behavior of the overlapping appointments validation. Since the three tests are in the same RSpec context they can share variables and setup code. In this case, the tests share a common doctor instance and setup code that creates an appointment with the doctor at 10:00 am for next Monday with a duration of 60 minutes.

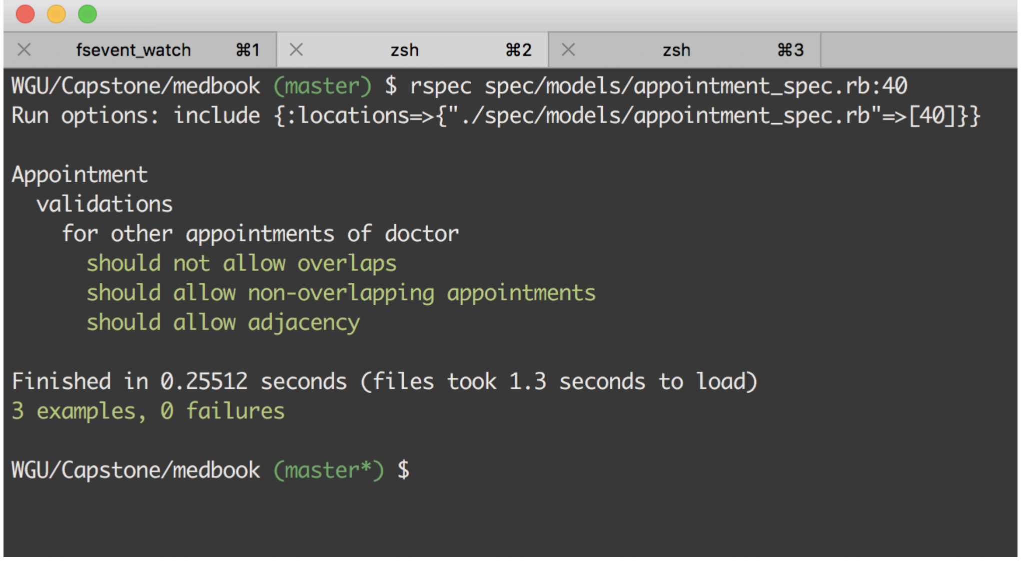
The first test, “should not allow overlaps,” creates a new appointment instance for 10:30 am on the same day and with a duration of 60 minutes. This would overlap the first appointment by half an hour. Using RSpec’s `expect` method, the test sets up the expectation that the appointment would not be valid and have an error with the message “Appointment overlaps with an existing one for the doctor.”

The second test, “should allow non-overlapping appointments” attempts to confirm that another appointment can be created for the doctor on the same day as long as the time does not overlap. An appointment created for 12:30 pm on the same day is expected to be valid and to have no errors.

The third test, “should allow adjacency,” creates an appointment that starts right after the first appointment ends. It has a start time of 11:00 am and a duration of 60 minutes. The expectation here is also that it will be valid and not have any errors.

Results

The following console output is printed when running these three tests in isolation. The green color indicates that the tests ran successfully and the last line of the output that reads “3 examples, 0 failures” means that 3 out of 3 tests ran successfully. As such, the expectations expressed in the test could be confirmed.



```
WGU/Capstone/medbook (master) $ rspec spec/models/appointment_spec.rb:40
Run options: include {:locations=>{"./spec/models/appointment_spec.rb"=>[40]}}

Appointment
  validations
    for other appointments of doctor
      should not allow overlaps
      should allow non-overlapping appointments
      should allow adjacency

Finished in 0.25512 seconds (files took 1.3 seconds to load)
3 examples, 0 failures

WGU/Capstone/medbook (master*) $
```

Source Code

A compressed version of the source code can be found in the file medbookr-master.zip

Link to Live Version

https://medbookr.herokuapp.com

User Guide

Set Up and Run Application for Maintenance Purposes

Several steps are needed to setup and run this Ruby on Rails web application in a local environment for the purposes of changing the code base or fixing bugs. This is a summary of the steps required to set up the website on Unix-based systems. For Windows machines, please follow the instructions here: https://gorails.com/setup/windows/10.

Prerequisites

The following system programs need to be installed:

- Ruby version 2.6.1 or higher
- The Bundler gem with a version of 2.0.1 or higher for managing library dependencies
- The PostgreSQL database management system

Installation

Change to the project directory and run these commands:

1. `bundle install` (this will install all dependencies)
2. `rails db:create && rails db:migrate && rails db:seed` (this will create and seed the database)
3. `rails s` (this will start the server locally)

Navigate to http://localhost:3000 in your browser to view the site.

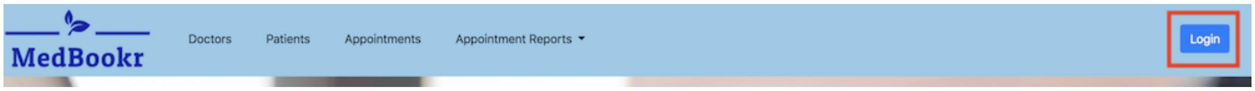
Guide For Admin User

The rest of the User Guide will provide a walk-through of the MedBookr medical appointment tracking application to demonstrate its functionality in detail.

Login with an Admin Account

Since the data tracked by the application is highly sensitive, new account creation has been disabled by default. A superuser, which is usually a member of the development team, will create admin accounts and then distribute credentials to select admin users.

To log in with an admin account click on the "Login" button in the top left corner of the home page. This will redirect you to the login page.

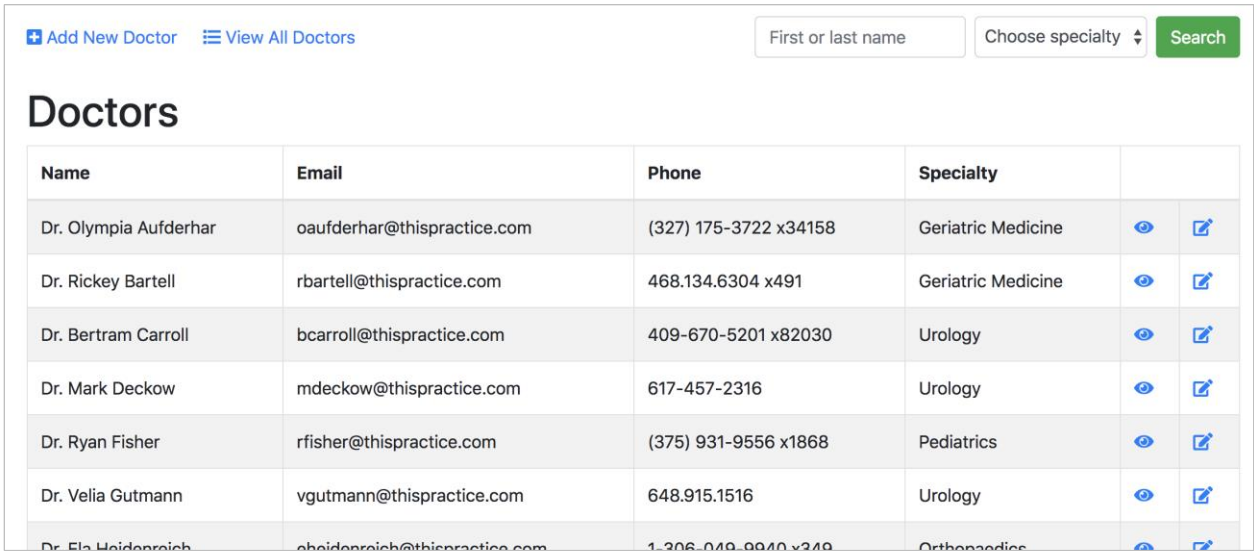


On the login page, enter admin@medbookr.com as the email and "medbookr" (all lower case) as the password. Then click the Login button. You will be redirected to the home page with the message "Signed in successfully." All menu items in the main navigation bar should be accessible to logged-in users. (Note to evaluators: This site is hosted on Heroku's free tier and as such the initial page load might take up to 30s. Once this first page loads, it will be up and running normally for subsequent requests)

Viewing and Managing Doctors

Viewing Doctors

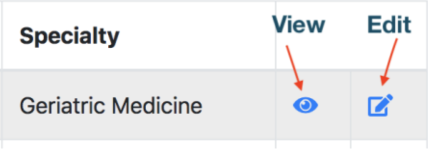
To view all doctors, click on the "Doctors" link in the top navigation bar. This will take you to a page that lists all doctors in a tabular format. The data in the table is sorted by last name and paginated. Pagination links can be found under the table.



Doctors

Name	Email	Phone	Specialty		
Dr. Olympia Aufderhar	oaufderhar@thispractice.com	(327) 175-3722 x34158	Geriatric Medicine	👁	✎
Dr. Rickey Bartell	rbartell@thispractice.com	468.134.6304 x491	Geriatric Medicine	👁	✎
Dr. Bertram Carroll	bcarroll@thispractice.com	409-670-5201 x82030	Urology	👁	✎
Dr. Mark Deckow	mdeckow@thispractice.com	617-457-2316	Urology	👁	✎
Dr. Ryan Fisher	rfisher@thispractice.com	(375) 931-9556 x1868	Pediatrics	👁	✎
Dr. Velia Gutmann	vgutmann@thispractice.com	648.915.1516	Urology	👁	✎
Dr. Ela Heidenreich	eheidenreich@thispractice.com	1-306-049-9940 x349	Orthopaedics	👁	✎

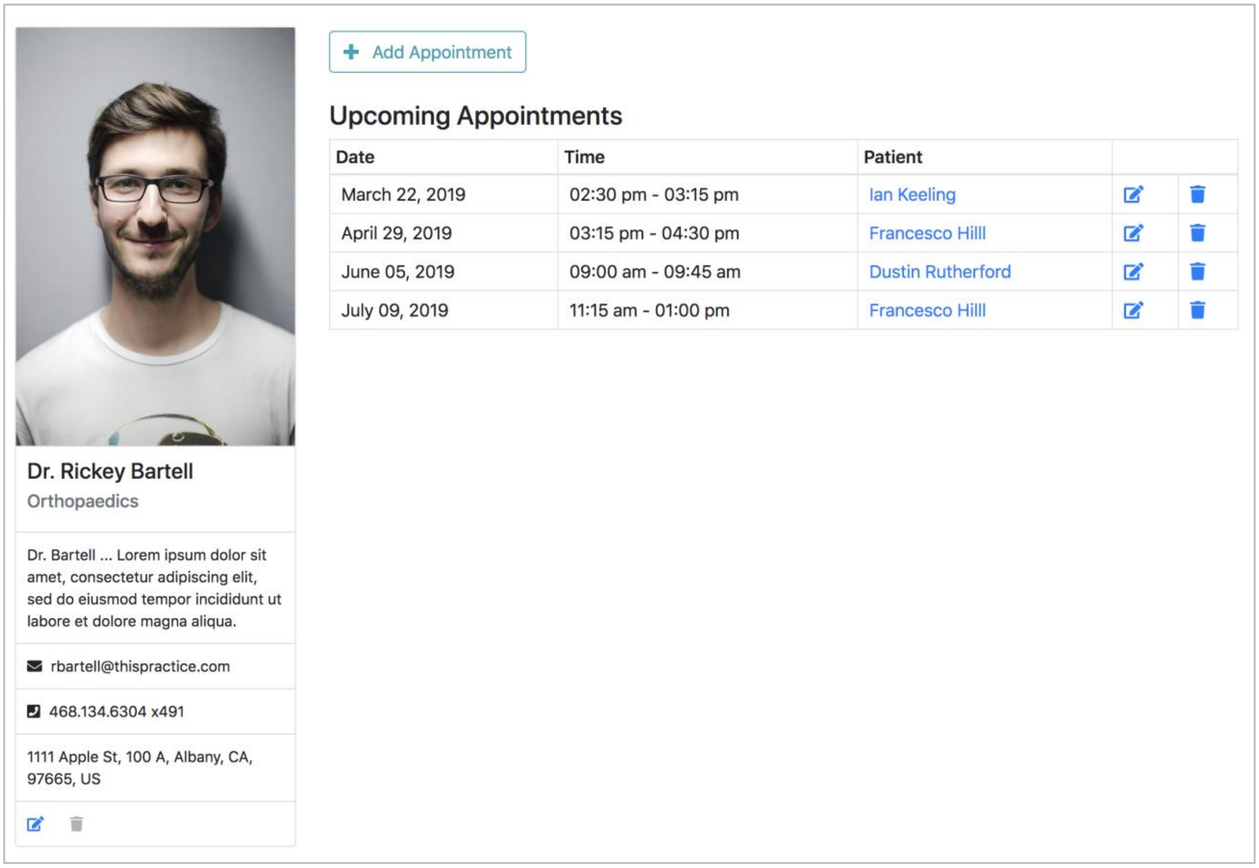
There is a search/filter form above the table that searches doctors by first or last name and/or specialty. For instance, if the field with the placeholder text “First of last name” is filled with “Auf ” then all doctors with that string of characters in their name will be returned, e.g. “Dr. Olympia Aufderhar.” If the dropdown “Choose specialty” is used, then only doctors with the chosen specialty will be shown. The two fields are independent and can be used separately or in conjunction. Note that the link “View all doctors” on the top left side will reset the data in the table after the search form has been used.



Specialty	View	Edit
Geriatric Medicine	👁	✎

The icons on each row are links to either view or edit the given doctor.

The view icon will redirect to a page with profile details and upcoming appointments for a given doctor. The profile page will have the doctor’s full contact details, including the address, a profile picture (if available) and a short bio.



Upcoming Appointments

Date	Time	Patient		
March 22, 2019	02:30 pm - 03:15 pm	Ian Keeling	✎	🗑
April 29, 2019	03:15 pm - 04:30 pm	Francesco Hilll	✎	🗑
June 05, 2019	09:00 am - 09:45 am	Dustin Rutherford	✎	🗑
July 09, 2019	11:15 am - 01:00 pm	Francesco Hilll	✎	🗑

Dr. Rickey Bartell
Orthopaedics

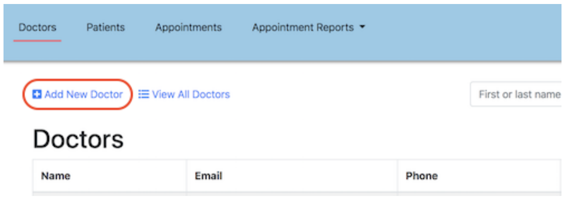
Dr. Bartell ... Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

✉ rbartell@thispractice.com

☎ 468.134.6304 x491

1111 Apple St, 100 A, Albany, CA,
97665, US

Creating a New Doctor

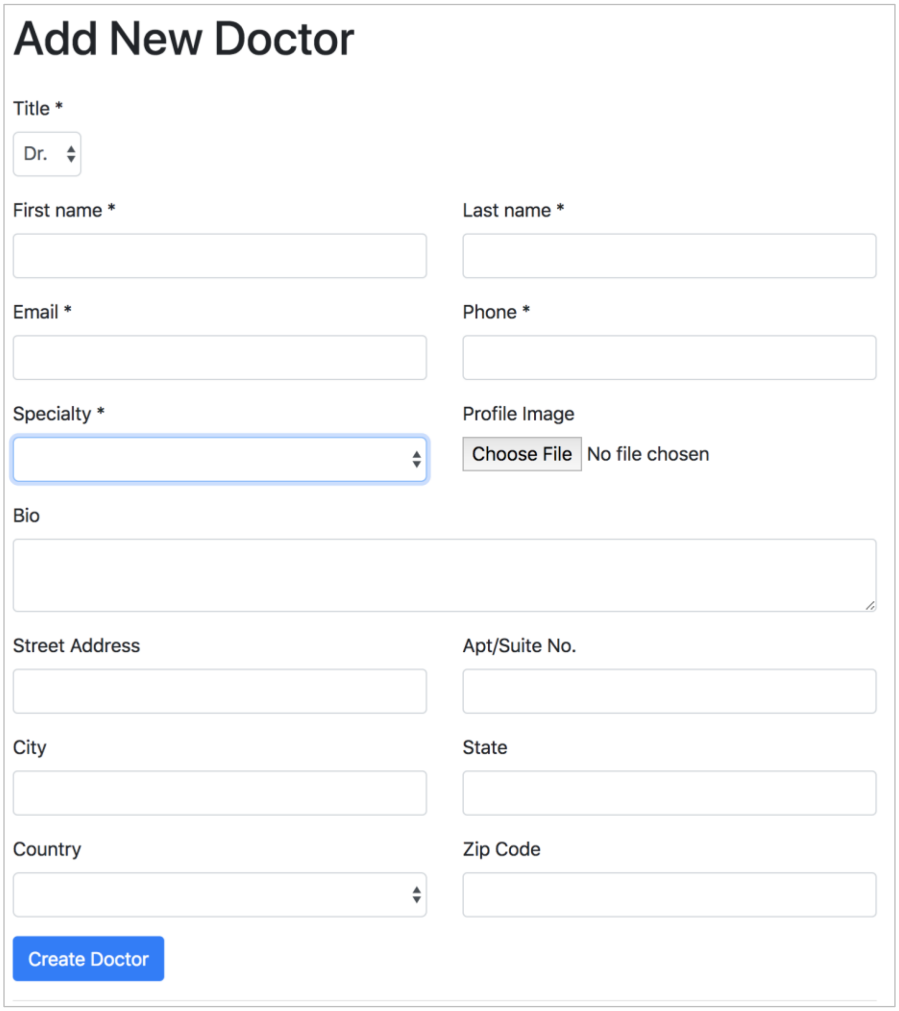


Doctors

Name	Email	Phone

To create a new doctor, click on the “+ Add New Doctor” link on the doctors listing page (accessible via the main navigation link).

The form to create a new doctor is fairly self-explanatory. Required fields are marked with an asterisk. The specialty can be chosen from a pre-defined list. The profile picture, bio and the address fields are optional.



Add New Doctor

Title * Dr.

First name *

Last name *

Email *

Phone *

Specialty *

Profile Image Choose File No file chosen

Bio

Street Address

Apt/Suite No.

City

State

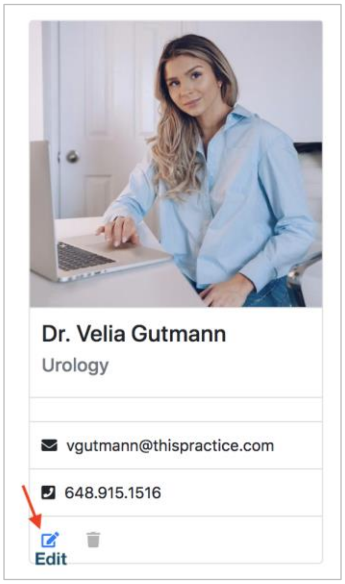
Country

Zip Code

Create Doctor

Updating Doctor Information

The form to update a doctor can either be reached from the list of doctors (by clicking the edit icon), or from the doctor’s profile page by clicking the edit icon under the contact details.



Dr. Velia Gutmann

Urology

✉ vgutmann@thispractice.com

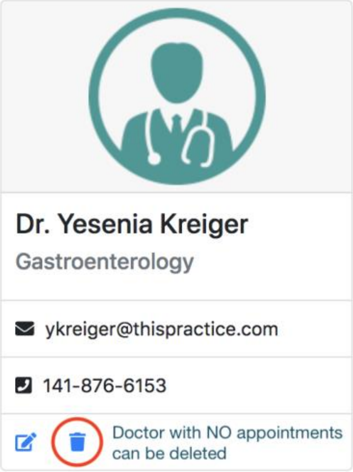
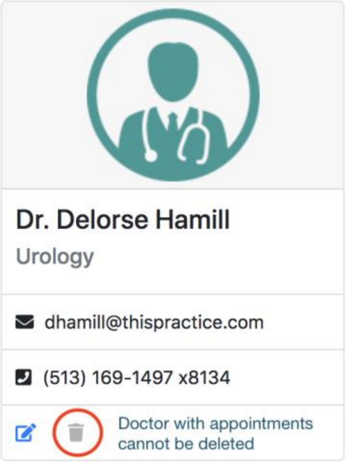
☎ 648.915.1516

Edit

The update form is very similar to the form to create a new doctor, except that it has the existing properties of the doctor filled in. Validation rules apply in the same way.

Removing a Doctor

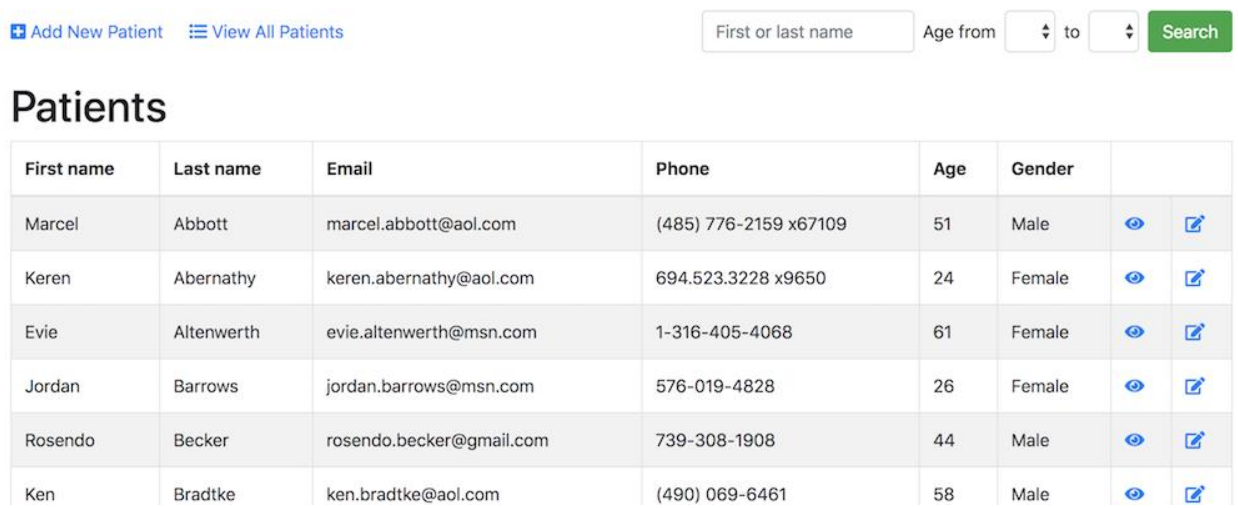
Deleting a doctor record is only possible if he or she has no upcoming appointments. As such, the delete icon under a doctor's image is grayed out if the doctor cannot be removed. Otherwise, the icon is blue and clickable. If a doctor with no appointments is deleted, the user gets redirected to the list of doctors with a message indicating that the doctor has been removed.

 Dr. Yesenia Kreiger Gastroenterology ✉ ykreiger@thispractice.com ☎ 141-876-6153 Doctor with NO appointments can be deleted	 Dr. Delorse Hamill Urology ✉ dhamill@thispractice.com ☎ (513) 169-1497 x8134 Doctor with appointments cannot be deleted

Viewing and Managing Patients

Viewing Patients

To view all patients, click on the “Patients” link in the top navigation bar. This will take you to a page that lists all patients in a tabular format. The data in the table is sorted by last name and paginated. Pagination links can be found under the table.

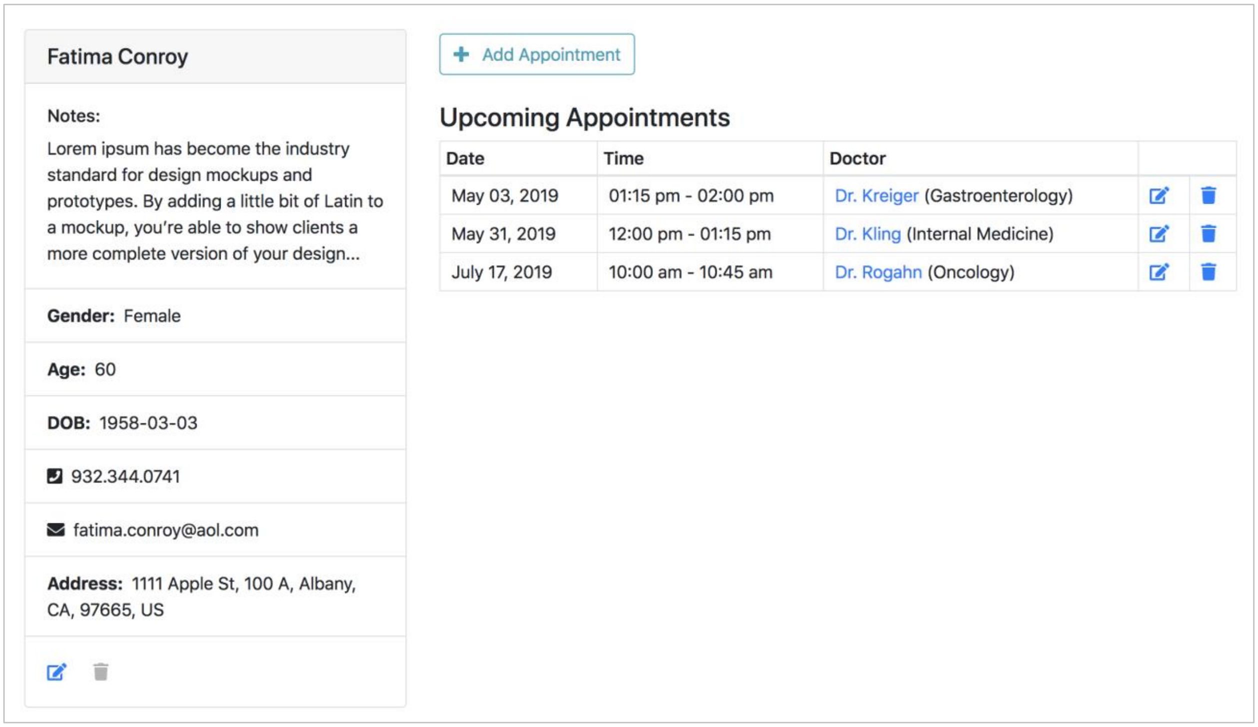


Patients

First name	Last name	Email	Phone	Age	Gender		
Marcel	Abbott	marcel.abbott@aol.com	(485) 776-2159 x67109	51	Male	👁	✎
Keren	Abernathy	keren.abernathy@aol.com	694.523.3228 x9650	24	Female	👁	✎
Evie	Altenwerth	evie.altenwerth@msn.com	1-316-405-4068	61	Female	👁	✎
Jordan	Barrows	jordan.barrows@msn.com	576-019-4828	26	Female	👁	✎
Rosendo	Becker	rosendo.becker@gmail.com	739-308-1908	44	Male	👁	✎
Ken	Bradtke	ken.bradtke@aol.com	(490) 069-6461	58	Male	👁	✎

There is a search and filter form above the table to allow searching for a patient by first or last name. Partial name fragments can also be used in the search. In addition, patients can be filtered by minimum or maximum age, or age range using the “Age from … to …” dropdowns. The name and age fields are independent and can be used individually or in conjunction. The “View All Patients” link will reset the table to show all patients if the search/filter form was used.

To view the details of a particular patient, click the “eye” icon in the table row of the patient you want to view. This will take you to a profile page for a particular patient with information such as address, and optional notes. Like the doctor’s profile page, it also lists the patient’s upcoming appointments:



Fatima Conroy

Notes:
Lorem ipsum has become the industry standard for design mockups and prototypes. By adding a little bit of Latin to a mockup, you're able to show clients a more complete version of your design...

Gender: Female

Age: 60

DOB: 1958-03-03

☎ 932.344.0741

✉ fatima.conroy@aol.com

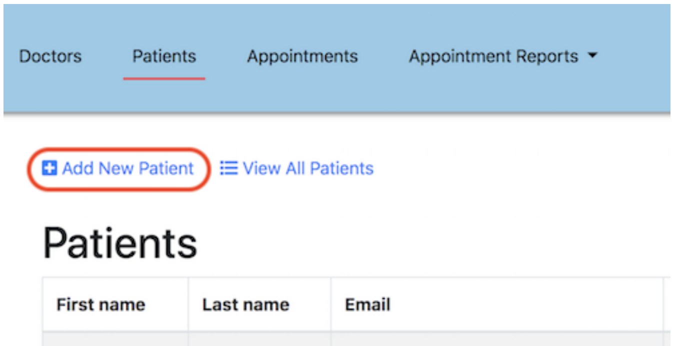
Address: 1111 Apple St, 100 A, Albany, CA, 97665, US

Upcoming Appointments

Date	Time	Doctor		
May 03, 2019	01:15 pm - 02:00 pm	Dr. Kreiger (Gastroenterology)	✎	🗑
May 31, 2019	12:00 pm - 01:15 pm	Dr. Kling (Internal Medicine)	✎	🗑
July 17, 2019	10:00 am - 10:45 am	Dr. Rogahn (Oncology)	✎	🗑

Creating a New Patient

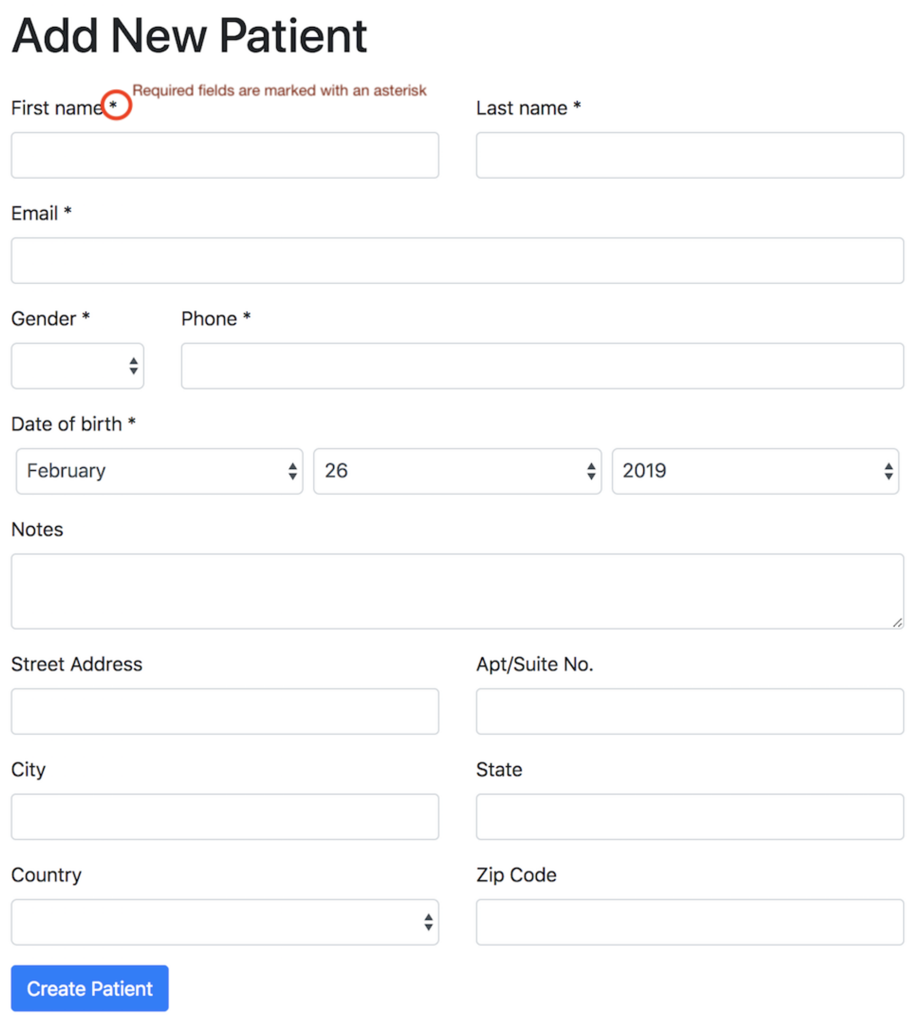
To create a new patient, click on the "+ Add New Patient" link on the patients listing page (accessible via the main navigation link).



Patients

First name	Last name	Email

The form to create a new patient is also fairly straightforward. Required fields are marked with an asterisk. Every field except for "Notes" and the address fields are required.



Add New Patient

Required fields are marked with an asterisk

First name *

Last name *

Email *

Gender *

Phone *

Date of birth *

February 26 2019

Notes

Street Address

Apt/Suite No.

City

State

Country

Zip Code

Updating Patient Information

To update a patient, either click on the “edit” icon on the patient profile page, or in the tabular display of all patients. The form to update a patient is the same as the form to create a new one, except that it has the existing properties of the patient filled in. The same validation rules apply.

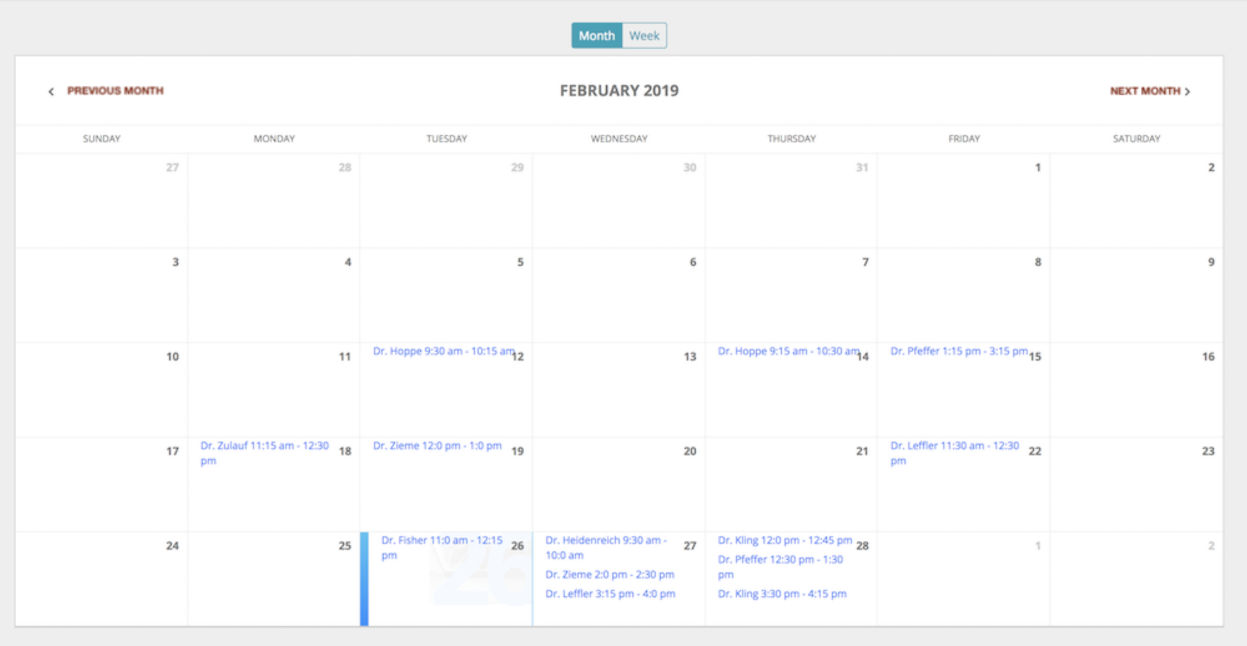
Removing a Patient

As is the case for doctors, a patient cannot be removed if he or she has any upcoming appointments. The delete icon on the patient profile page will be grayed out and disabled for a patient with appointments and be blue and active otherwise. Deleting a patient will redirect an admin user back to the patient listing page with an appropriate confirmation message.

Viewing and Managing Appointments

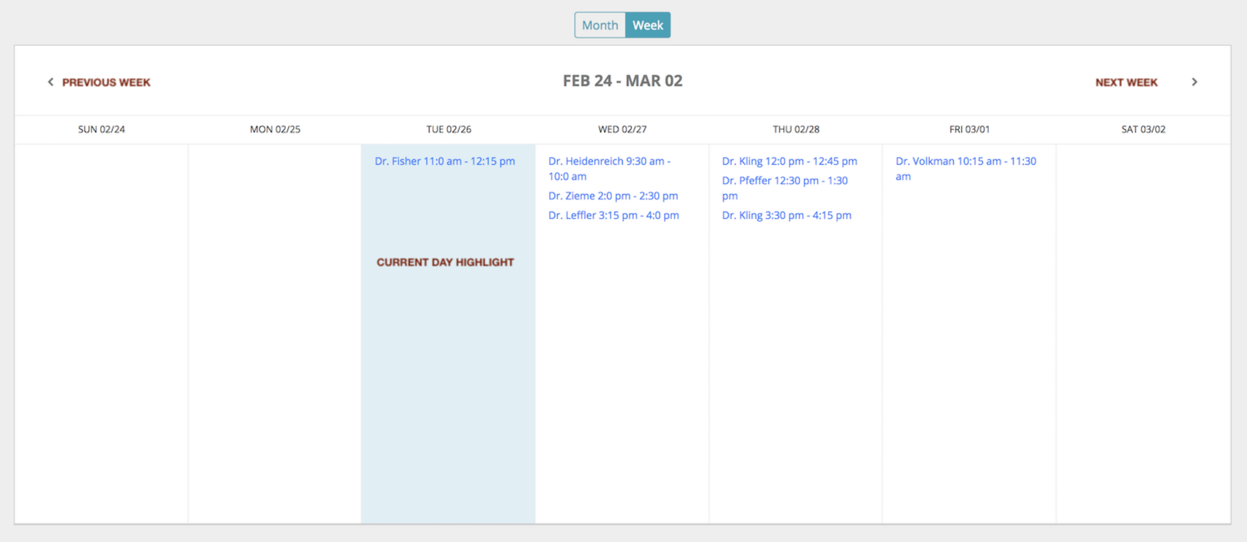
Viewing Appointments by Month and Week

To view the appointment calendar, click on the “Appointments” link in the main navigation bar. This will reveal a calendar view that initially lists all appointments by month. The current month is shown by default and the current day is highlighted. The arrows on the very left and the very right are for navigating to the previous or the next month.



SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12 Dr. Hoppe 9:30 am - 10:15 am	13	14 Dr. Hoppe 9:15 am - 10:30 am	15 Dr. Pfeffer 1:15 pm - 3:15 pm	16
17	18 Dr. Zulauf 11:15 am - 12:30 pm	19 Dr. Zieme 12:0 pm - 1:0 pm	20	21	22 Dr. Leffler 11:30 am - 12:30 pm	23
24	25	26 Dr. Fisher 11:0 am - 12:15 pm	27 Dr. Heidenreich 9:30 am - 10:0 am Dr. Zieme 2:0 pm - 2:30 pm Dr. Leffler 3:15 pm - 4:0 pm	28 Dr. Kling 12:0 pm - 12:45 pm Dr. Pfeffer 12:30 pm - 1:30 pm Dr. Kling 3:30 pm - 4:15 pm	1	2

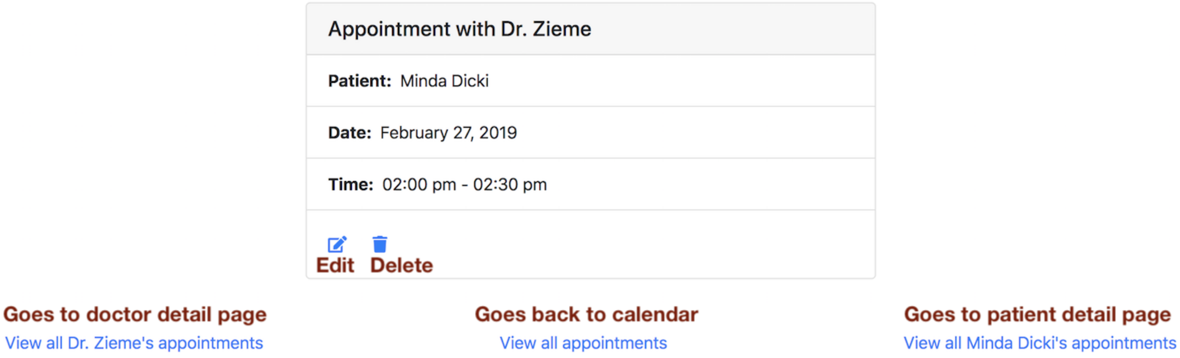
To switch to the weekly view of the calendar, click on “Week.” Here too, the left and right arrows allow for navigation to previous and next weeks.



FEB 24 - MAR 02

SUN 02/24	MON 02/25	TUE 02/26	WED 02/27	THU 02/28	FRI 03/01	SAT 03/02
		Dr. Fisher 11:0 am - 12:15 pm	Dr. Heidenreich 9:30 am - 10:0 am Dr. Zieme 2:0 pm - 2:30 pm Dr. Leffler 3:15 pm - 4:0 pm	Dr. Kling 12:0 pm - 12:45 pm Dr. Pfeffer 12:30 pm - 1:30 pm Dr. Kling 3:30 pm - 4:15 pm	Dr. Volkman 10:15 am - 11:30 am	

To view details, click any of the appointments in the calendar. They are linked to a detail page with full information on the selected appointment. The page displays the names of doctor and patient, the date and time, as well as links to the profile pages of doctor and patient:


Edit
Delete
View all Dr. Zieme's appointments
View all appointments
View all Minda Dicki's appointments

Appointment with Dr. Zieme

Patient: Minda Dicki

Date: February 27, 2019

Time: 02:00 pm - 02:30 pm

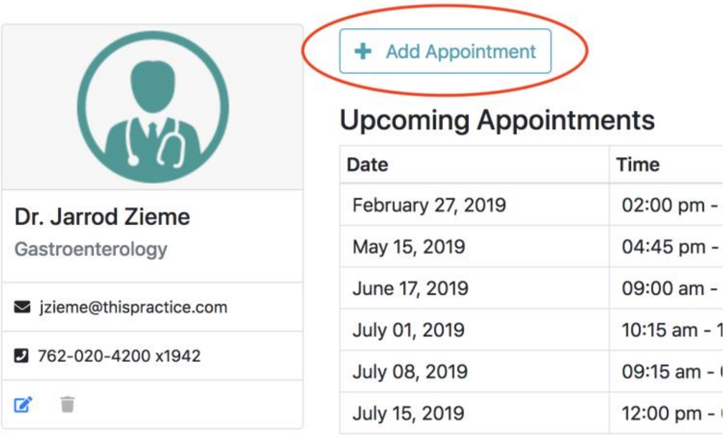
Goes to doctor detail page

Goes back to calendar

Goes to patient detail page

Creating New Appointments

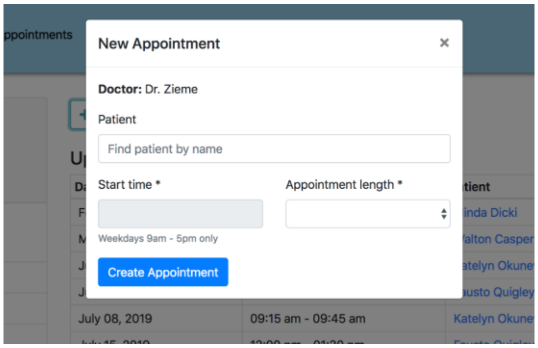
To create a new appointment, go to the detail page of a particular doctor or patient. There will be a big button to create new appointments. See for instance the page of Dr. Zieme:



Dr. Jarrod Zieme
Gastroenterology
✉ jzieme@thispractice.com
☎ 762-020-4200 x1942

Upcoming Appointments

Date	Time
February 27, 2019	02:00 pm -
May 15, 2019	04:45 pm -
June 17, 2019	09:00 am -
July 01, 2019	10:15 am - 1
July 08, 2019	09:15 am -
July 15, 2019	12:00 pm -



New Appointment ×

Doctor: Dr. Zieme

Patient

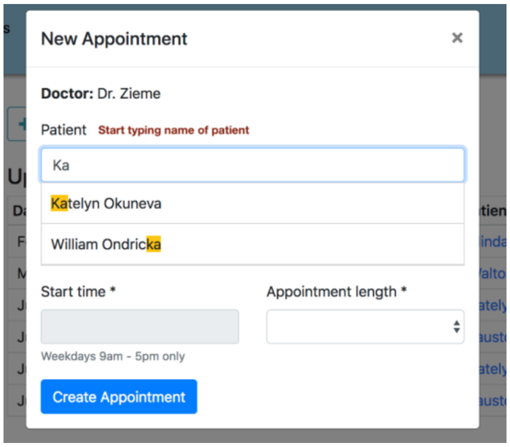
Find patient by name

Start time * | Appointment length *

Weekdays 9am - 5pm only

July 08, 2019 09:15 am - 09:45 am Katelyn Okune

Clicking on the button will trigger a modal window with a form to add the appointment. Note that the doctor is pre-selected.



New Appointment ×

Doctor: Dr. Zieme

Patient **Start typing name of patient**

Ka

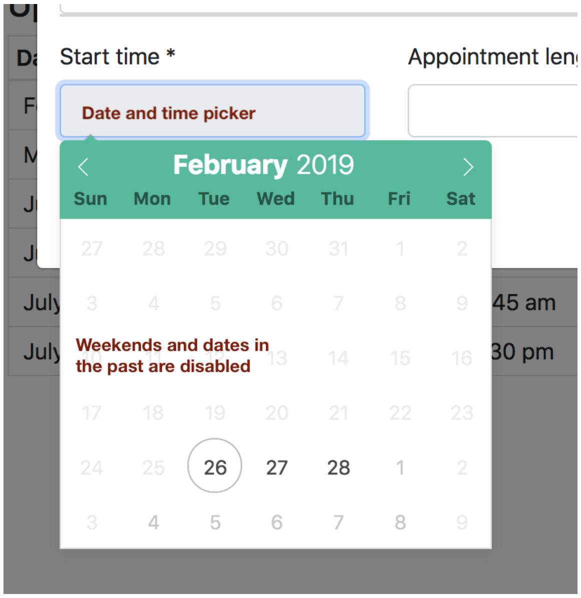
Katelyn Okuneva

William Ondricka

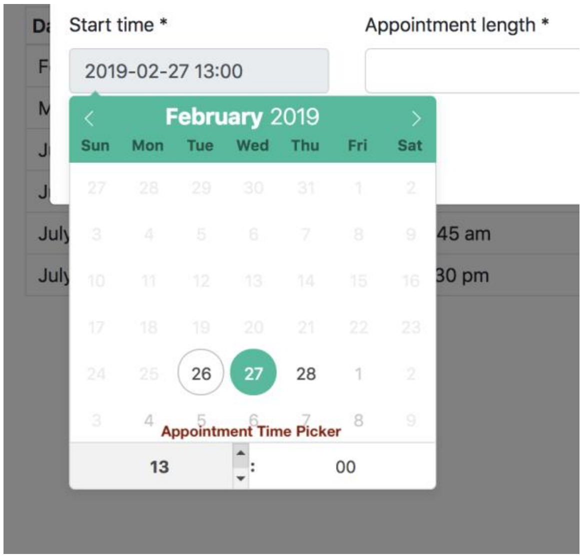
Start time * | Appointment length *

Weekdays 9am - 5pm only

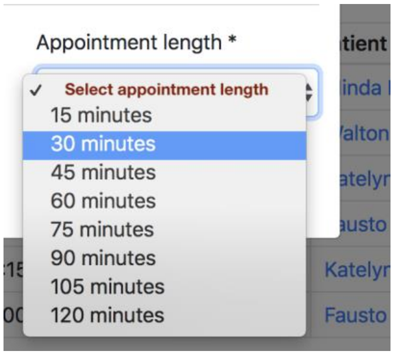
To set a patient for the appointment, start typing the name of the patient, e.g. “Ka” to find the patient named “Katelyn Okuneva.” The patient autocomplete field will show patients that have the letters as part of their name.



To set a date for the appointment, click the "Start Time" field. A date picker will appear to allow setting the date and time. By default, dates on weekends (Sat and Sun) and dates in the past are disabled and cannot be selected. Click on a date to select. The left and right arrows allow for navigation between months.


Start time * 2019-02-27 13:00

Once the date has been selected, a time picker will appear to set the start time of the appointment. By default, only times during business hours (9-17) are selectable. Set the hour and date of the appointment by using the up and down arrows next to the hour and the minute.

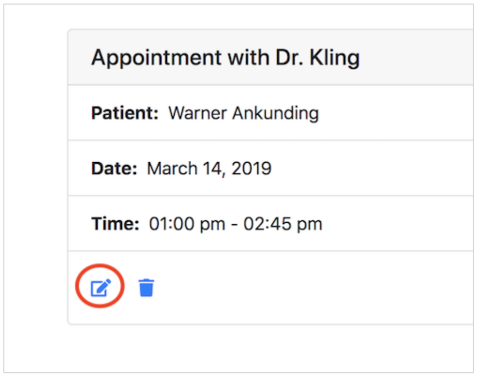


Finally, select the appointment duration using the "Appointment length" select and click on "Create Appointment." The appointment will be created and appear in the list of the doctor's appointments.

The procedure for adding an appointment for a patient is exactly the same but would start from the patient's detail page and the patient rather than the doctor would be pre-selected in the appointment modal.

Updating Appointments

The update form for an appointment can be reached from the appointment detail page or the upcoming appointment listings on a doctor's or a patient's profile page. For instance:



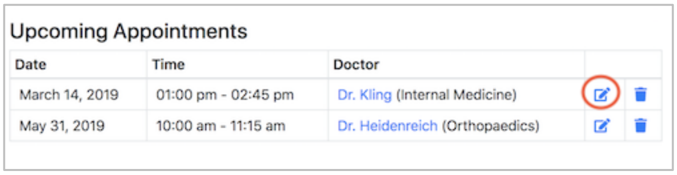
Appointment with Dr. Kling

Patient: Warner Ankunding

Date: March 14, 2019

Time: 01:00 pm - 02:45 pm

Appointment detail page (accessible from calendar) with edit and delete links.



Upcoming Appointments

Date	Time	Doctor
March 14, 2019	01:00 pm - 02:45 pm	Dr. Kling (Internal Medicine)
May 31, 2019	10:00 am - 11:15 am	Dr. Heidenreich (Orthopaedics)

Appointment edit link on a patient's profile/detail page. The same table with links also exists for the doctor.

Updating an appointment does not occur in a modal but on a separate page. The form, however, works the same. There are two auto-complete fields – one for the doctor and one for the patient. To change either, highlight and remove the doctor's or the patient's name and start typing the name of another. The start time can be changed via the date and time picker and the duration via the duration select menu.

Deleting an Appointment

The same locations that have an appointment edit link, also have link to delete the appointment. Click the trash icon next to any appointment to delete it. It will delete the appointment and show a confirmation notice.

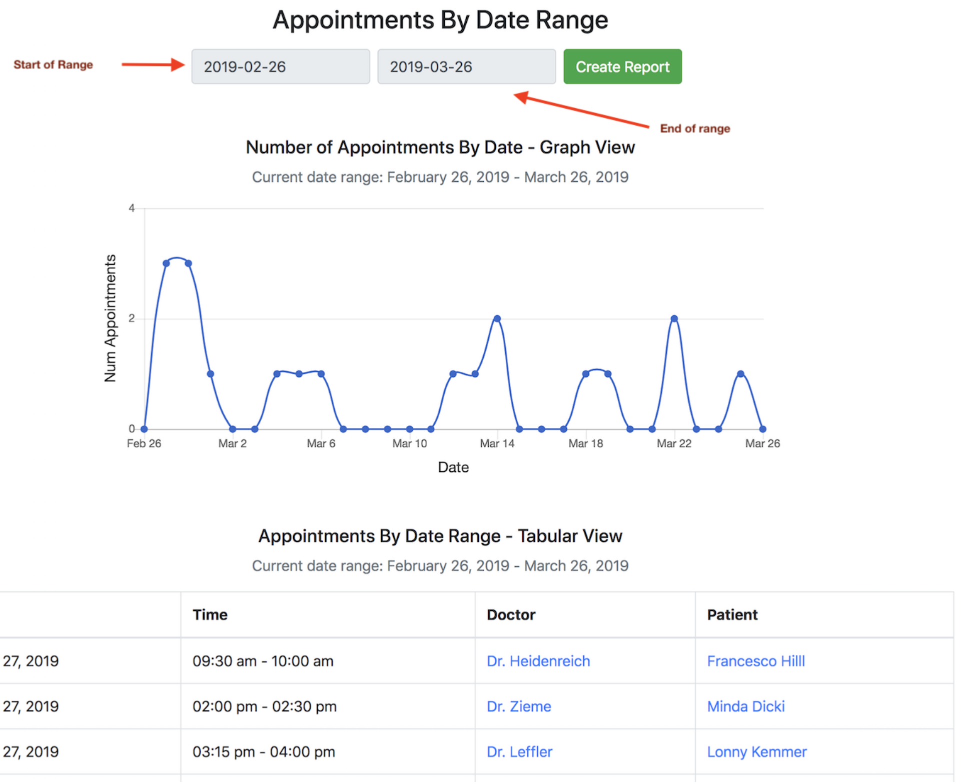
Viewing Appointment Reports

Three types of reports are available to users of the application:

1. A graphical and tabular report to show appointments within a certain date range.
2. A graphical report on the frequency of appointments based on time of day.
3. A graphical report on the frequency of appointments based on the doctor.

Appointments by Date Range

This report can be viewed by selecting “Appointment Reports” > “By Date Range” from the dropdown in the top navigation bar. The page displays both a graphical summary and a tabular view of all appointments within a certain date range. By default, the range starts with today’s date and show all appointments for the next month. The graph displays the appointment counts on each day. The table shows the appointment date and time and the name of the doctor and patient involved. You can change the range by using the date pickers on the top of the page.



Appointments By Date Range

Start of Range

2019-02-26 2019-03-26

End of range

Number of Appointments By Date - Graph View

Current date range: February 26, 2019 - March 26, 2019

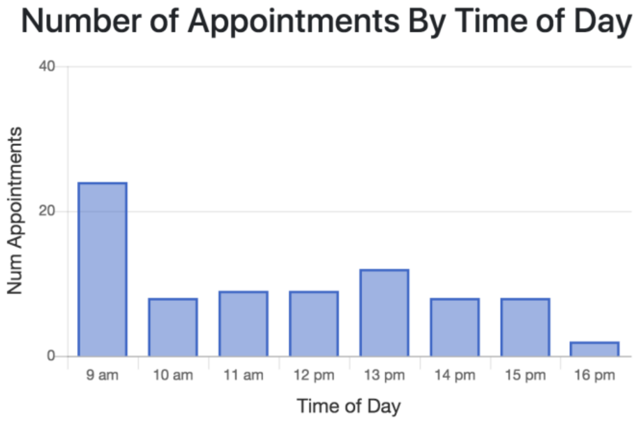
Appointments By Date Range - Tabular View

Current date range: February 26, 2019 - March 26, 2019

Date	Time	Doctor	Patient
February 27, 2019	09:30 am - 10:00 am	Dr. Heidenreich	Francesco Hilll
February 27, 2019	02:00 pm - 02:30 pm	Dr. Zieme	Minda Dicki
February 27, 2019	03:15 pm - 04:00 pm	Dr. Leffler	Lonny Kemmer

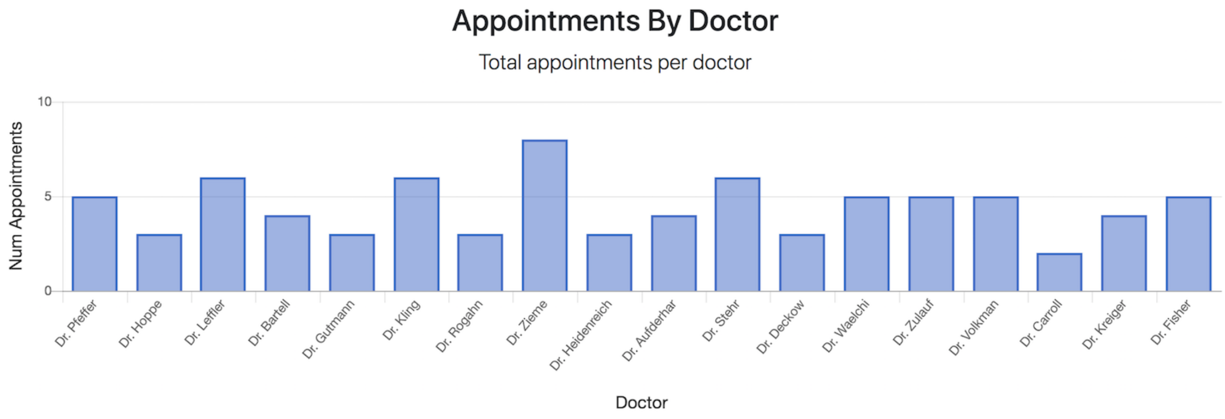
Appointments by Time of Day

The page with a graph for appointment frequencies by time of day can be reached by clicking the “By Time of Day” link under the “Appointment Reports” dropdown. The report shows a simple bar chart with the number of appointments grouped by time of day. This information can be used to make scheduling decisions and distribute appointment frequencies more evenly throughout a day.



Appointments by Doctor

The report with the number of appointments by doctor can be accessed by selecting “By Doctor” from the “Appointment Reports” dropdown.


Appointments By Doctor
Total appointments per doctor
Num Appointments
Dr. Pfeffer
Dr. Hoppe
Dr. Leffler
Dr. Bartell
Dr. Gutmann
Dr. Kling
Dr. Rogahn
Dr. Zieme
Dr. Heidenreich
Dr. Aufderhar
Dr. Stehr
Dr. Deckow
Dr. Waelchi
Dr. Zulauf
Dr. Volkman
Dr. Carroll
Dr. Kreiger
Dr. Fisher
Doctor