Large and Cloud-based Software Systems

Multi-service Appointment Booking Web App

System Architecture

authors:	Mohammad Alhussan, Jaser Bin Rahman, Sharath Tirumalaraju, Sami Sallam, Mohammad Alsane
date	13/05/2019
revision	0.1

Technology
Arts Sciences
TH Köln

Table of Contents

1 Revisions and Responsibilities

revision	date	Authors	remarks
0.1	05/13/2019	Alhussan, Rahman, Sallam, Tirumalaraju, Alsane	some bullets for each chapter
…			

Table 1 document revisions

author	authorship of / responsible for
Mohammad Alhussan	Chapters 7, 11 and 12
Jaser Bin Rahman	Chapters 5 and 6
Sami Sallam	Chapters 2, 3 and 4
Sharath Tirumalaraju	Chapters 10 and 13
Mohammad Alsane	Chapters 8 and 9

2 Functional Requirements

This project aims to introduce a Scheduling Online Services that would ease off the appointment-scheduling journey for users and pave the path of a better different online services experience. The proposed system advances with online facilities that eliminate the chaos of traditional appointment services. This system offers online appointment booking, to view doctors list, restaurants list, Theaters lists, etc. in the near area to him/her and to cancel and update appointments. It provides advanced functionality to streamline the process thus easy access to personal services that help organizations to stay connected with their customers, clients and most importantly users and can result in significant time and monetary savings in order to develop a successfully running online Scheduling, the system is required to interact with system database, scheduling module and the administrative module for example, to achieve the best implementation, the scheduling system would be able to interact with several organizations such as clinics, restaurants car rent companies.

Functional **Requirements:**

1) SQL DataBase: using google instant to create SQL DataBase
2) Web Server VM Ubuntu (inside that VM):
    a) Apache (To active the web Server service)
    b) PHPMyAdmin
    c) MySQL Client
    d) MySQL Server
    e) Wordpress (for the web application design)
3) Documentation for IP Addresses for each machine to connect the DataBase with the web application.
4) DataBase schema.
5) Simple design for the pages of web application.

For now, basic functionalities have been implemented but for the future, work will be done to many organizations to the system. Online follow-up for the distant users is another feature that is aimed to add at later stages.

3 Quality Attributes

Developing systematic ways to relate the software quality attributes of a system to the system’s architecture provides a sound basis for making objective decisions about design tradeoffs and enables engineers to make reasonably accurate predictions about a system’s attributes that are free from bias and hidden assumptions. The goal is the ability to quantitatively evaluate and trade off multiple software quality attributes to arrive at a better overall system. The purpose of this section is to take a small step in the direction of developing a unifying approach for reasoning about multiple software quality attributes. This section examines APPointment software quality attributes: performance, dependability, security, and safety. Each attribute has matured (or is maturing) within its own community, each with their own vernacular and point of view. We propose a generic taxonomy for describing each attribute and attempt to use this taxonomy to suggest a direction for developing an attribute-based methodology for evaluating APPointment software architectures.

We are going to characterize APPointment system by the quality attributes, which are most important as explained in Table 2, and it is used to give each category a rating from 1 (very unimportant) to 5 (very important).

requirement category	**rating**	**Explanation**
speed of development	2	APPointment is not a product that has to be pushed on the market with high time pressure. In addition to the speed of development for later versions is severely hampered. But with Breaking down the entire project into small parts, making the overall task management better, using good planning tools, right frameworks, libraries, extension languages, plugins, decreasing system complexity and discourage cowboy coding can save a lot of time during development and maintainability.
inexpensiveness	2	APPointment is expensive through the development and deployment.
size of scope	3	International System
performance		
response time	4	APPointment is a web application and its response time is of major importance. With higher response times conversion rate, i.e. the probability that a visitor becomes a paying customer, drops significantly. Due to the variation of response times, the response time is planned to be below 1000ms in the 99th percentile.
throughput	4	APPointment must cope with a workload of 1000 appointment requests and responses per minute and this number will increase through later development, and due to the relation between hardware resources like RAM, storage and network and throughout, hardware can play an important role during the system design which will be very considered.
capacity	4	APPointment should store a maximum of about 100.000.000.000 appointment tickets and may vary in the future according to the market evolution.
scalability	3	APPointment resources are supposed to be utilized at an average of 60% while the workload peaks vary between 1 requests and response per minute (3 am on a summer vacation day) and 20 requests and responses per minute (6 p.m. Christmas holidays).

requirement category	rating	Explanation
dependability		
availability	4	The probability of failure less than 0.01%. Uptime of at least 99%.
reliability	3	APPointment must continue to store and retrieve appointments even if all partner systems are unavailable. Less than 30 minutes needed to recover from system failure.
Resilience	3	Certain faults in APPointment can be compensated. And these compensations keep MTTR low and/or lead to a slightly degraded system operational state instead of a complete failure.
Consistency	4	Every request enters and responses leaves a database in a consistent state in APPointment system.
Maintainability		
Operability	3	Metrics, logging and tracing should be monitored in APPointment system to keep relevant stakeholders (e.g. admins or support) informed about the current system state and its event history.
Simplicity	4	APPointment will be designed to be understood and easy implemented manner, also take considering of Chunking and Schema Building.
Evolvability	3	APPointment will be designed to implement and evolve towards a change in a system and future requirements expectations.
Security	4	The password should be at least 8 characters, 1 Upper case, 1 lower case and 1 number. Website should use different techniques in order to have secure transfer of data to database. All the user's data can't be sold or distributed to other entities without their previous approval.
Usability	4	Interface elements (e.g. menus) should be easy to understand. The user should be able to learn to use a system in less than 30 minutes. Time required for registration less than 5 minutes. Error messages should explain how to recover from the error. Actions which cannot be undone should ask for confirmation.

Table 2: Quality attributes

4 Quality Attribute Scenarios

We now discuss the four most important system quality attributes (availability, security, usability and performance) for APPointment system, with providing a way to generate specific scenarios for that attribute as shown in the tables below.

Table 4.1 gives elements of one of the scenarios that characterize APPointment performance.

Portion of Scenario	Possible Values
Source	Collection of users (Clients)
Stimulus	The stochastic initiation of 1000 requests and responses per minute.
Artifact	The System
Environment	The system is in normal mode
Response	Processes requests and responses (transactions)
Response Measure	Requests and responses should be processed with an average latency of two seconds.

Table 4.2 gives elements of one of the scenarios that characterize APPointment Security.

Portion of Scenario	Possible Values
Source	Individual, unknown identity, external, not authorized, access to limited resources (database)
Stimulus	Tries to steal credit card numbers.
Artifact	The data
Environment	online, connected, firewalled
Response	Blocks access to data
Response Measure	Probability of detecting attack

Table 4.3 gives elements of one of the scenarios that characterize APPointment usability.

Portion of Scenario	Possible Values
Source	End user
Stimulus	Wants to learn system features; feel comfortable.
Artifact	The system
Environment	At runtime
Response	System provides “learn system features” interface and familiar to user
Response Measure	User satisfaction

Table 4.4 gives elements of one of the scenarios that characterize APPointment availability.

Portion of Scenario	Possible Values
Source	external to the system
Stimulus	Response
Artifact	Processes

Portion of Scenario	**Possible Values**
Environment	degraded mode (i.e., fewer features, a fall back solution)
Response	continue to operate in degraded mode
Response Measure	Time interval in which system can be in degraded mode

5 Constraints

constraint	category	explanation and rationale
Google Cloud	technical	Our application will be a Google cloud-based application. So, we have to agree with the GCP.
Schedule	organizational	Sometimes stakeholders expect a unrealistic schedule for developing a project which may lead to quality degradation.
Cost	organizational	We need more resource to accommodate more user and minimize response time which causes suppression of budget. So, optimal cost-efficient resource management is very important for this project.
Scope	organizational	The scope is one of the significant constraints as the application will accommodate multiple service provider in one platform. It will not be possible to focus on specific goals instead of focus on common goals.
Laws and regulation	Political	The new EU data protection policy applies to our system.

6 Stakeholders

stakeholder group	class	responsibilities, interests and concerns
Project Manager	Promoters	The project manager is responsible for managing the whole project within the deadline and controls the respective teams. A project manager also communicates with other stakeholders.
Customers	Promoters	The customers will inform about their requirements. The developer team will try to add features according to the elements.
End User	Latents	The opinions of the users will play an essential role in development. The review will help to develop a user-friendly application.
Tester	Apathetics	A tester will test the functionality of the application and will inform the behavior to the develpoer team. The tester team will notify the modified response to the developer team before releasing an update of the software.
Sales Team	Defenders	The sales team is responsible for selling the product to the customers. They will explain the functionality and comparison of the product to the customers. They will collect customer feedback and will inform the concerned team.
Government	Apathetics	The government will also be part of the stakeholders. They will monitor and inform us about the modification of internet policy.
Internet Security	Latents	The internet security team will check the security part and will suggest the developer team about the required modification.

7 Scope and Context

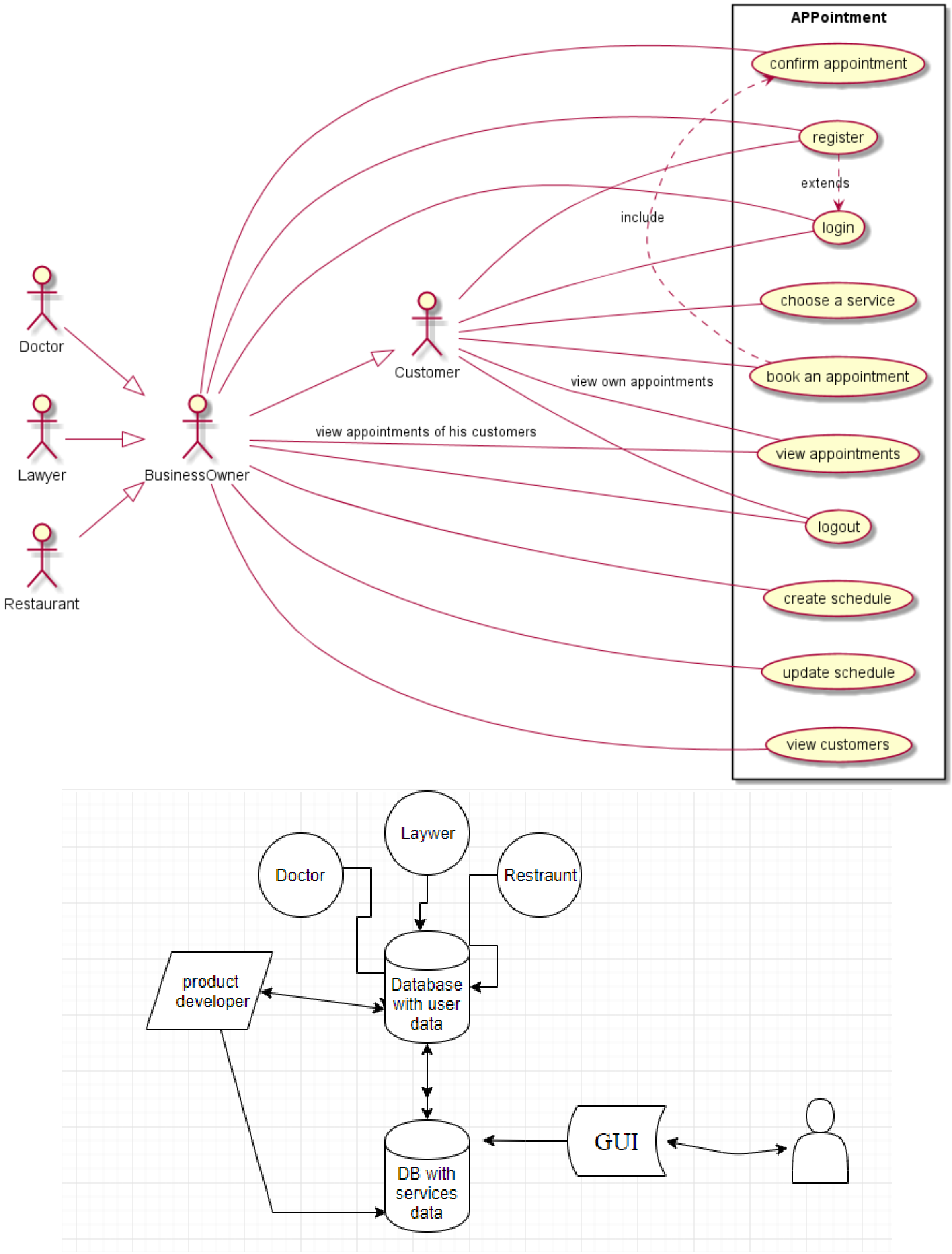

Figure 1 Use case diagram of APPointment

In every use case shown above, there will be an interaction with the backend – Web server and database.

8 Key Design Decisions

decision	category	explanation and rationale
RESTful web API	technology	Appointment Booking App will offer a RESTful web API, in order to support non-browser clients to access the App … (Mobile App, Desktop App, …)
No SOAP-based web API	organizational	The number of potential non-browser clients that rely on SOAP instead of a RESTful API is neglect able. Therefore, Appointment App does not offer a SOAP-based web API due to cost reasons.
Stand-alone web UI	technology	Appointment App offers a web UI that is rendered exclusively in a browser tab. This stand-alone web UI, e.g. Edit and manage the calendar for the customer (Doctor, Lawyer, …)
Iframe	technology	Appointment App offers a web UI which can be integrated via iframes into web pages of third-party websites.
Smartphone-App	organizational	We expect users to install a dedicated smartphone app for occasional rating. Therefore, Appointment smartphone app is not in scope. (feature work)
distributed NoSQL data-base	quality attribute	Since performance beats consistency in Appointment App, we will use a distributed NoSQL database.

9 Deployment View

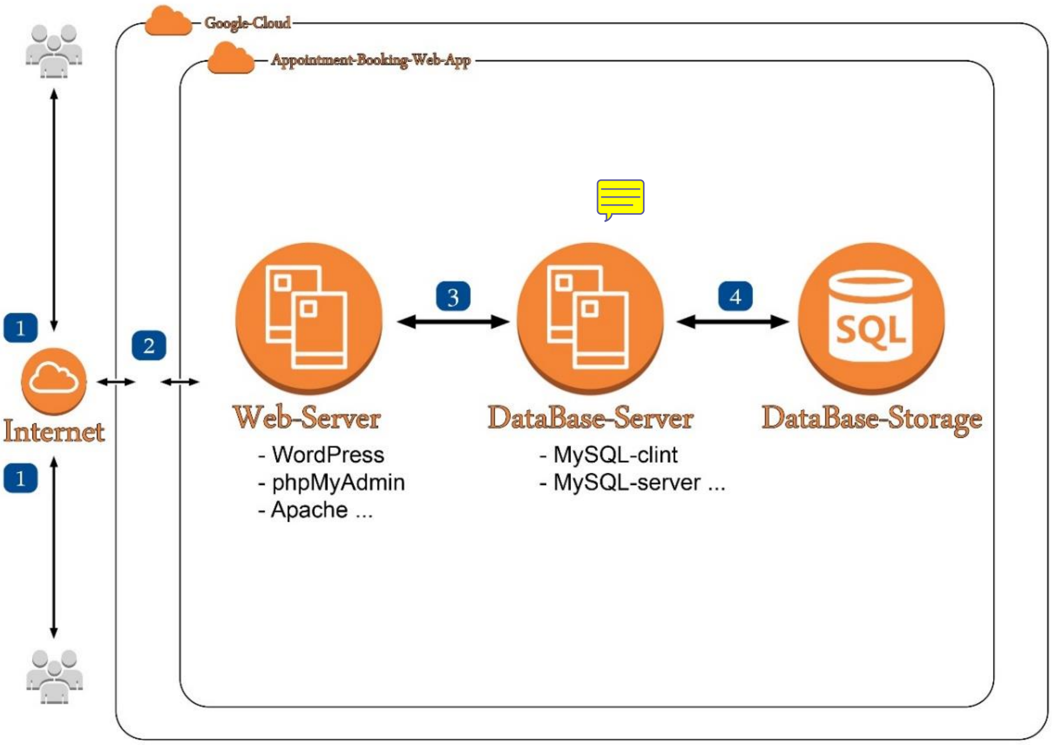

Figure 2 Deployment view

component(s)	explanation
Dataflow between clients (1) and google gateway (2)	Inbound traffic from clients is https. http requests will be redirected to https. Dataflow will be fully encrypted.
Web Application Server (2 and 3)	The Web Application servers are GCP virtual machine instances (Linux Ubuntu). The web Application Server for the website and pages (web-application). All connection between clients and Database will be through Webserver.
Dataflow from Application Server (3) to Database Server (4)	The Database Server is an instance of GCP's "Cloud SQL" using the MySQL-option with failover and replication. The connection between the application servers and the database is encrypted using GCP's encryption mechanisms.

10 Software Component View

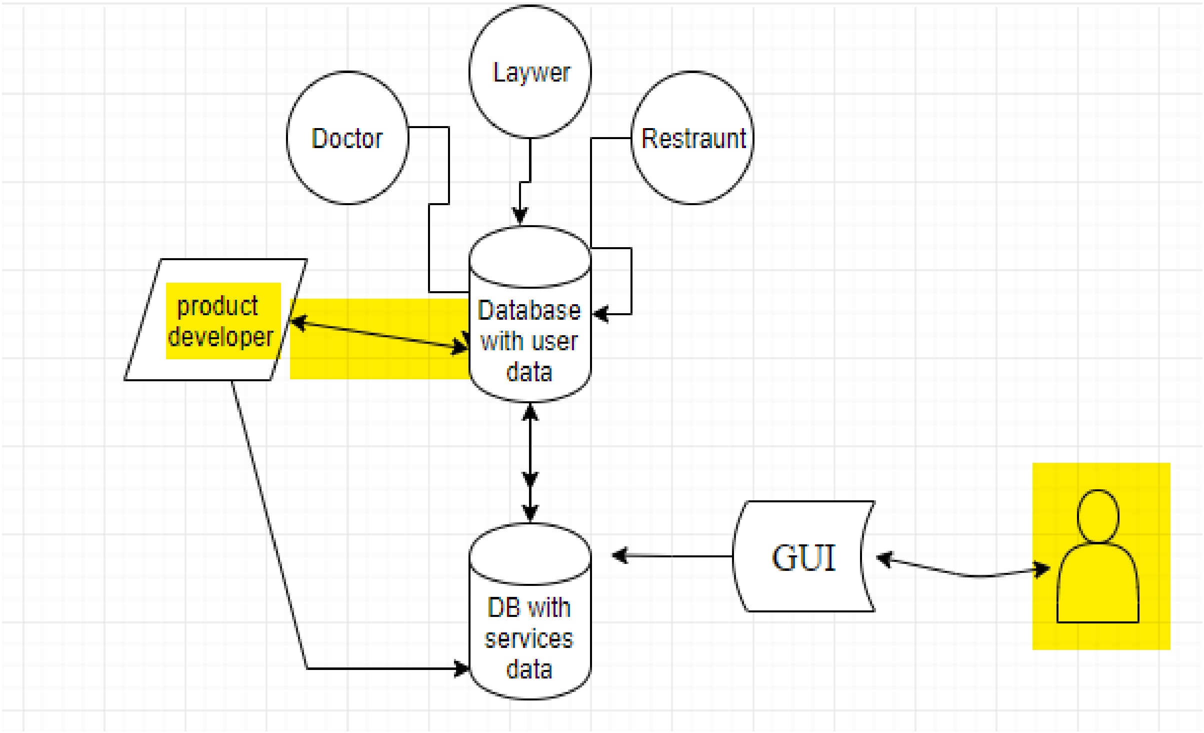

Figure 3 System Architecture of APPointment

11 Data Schema View

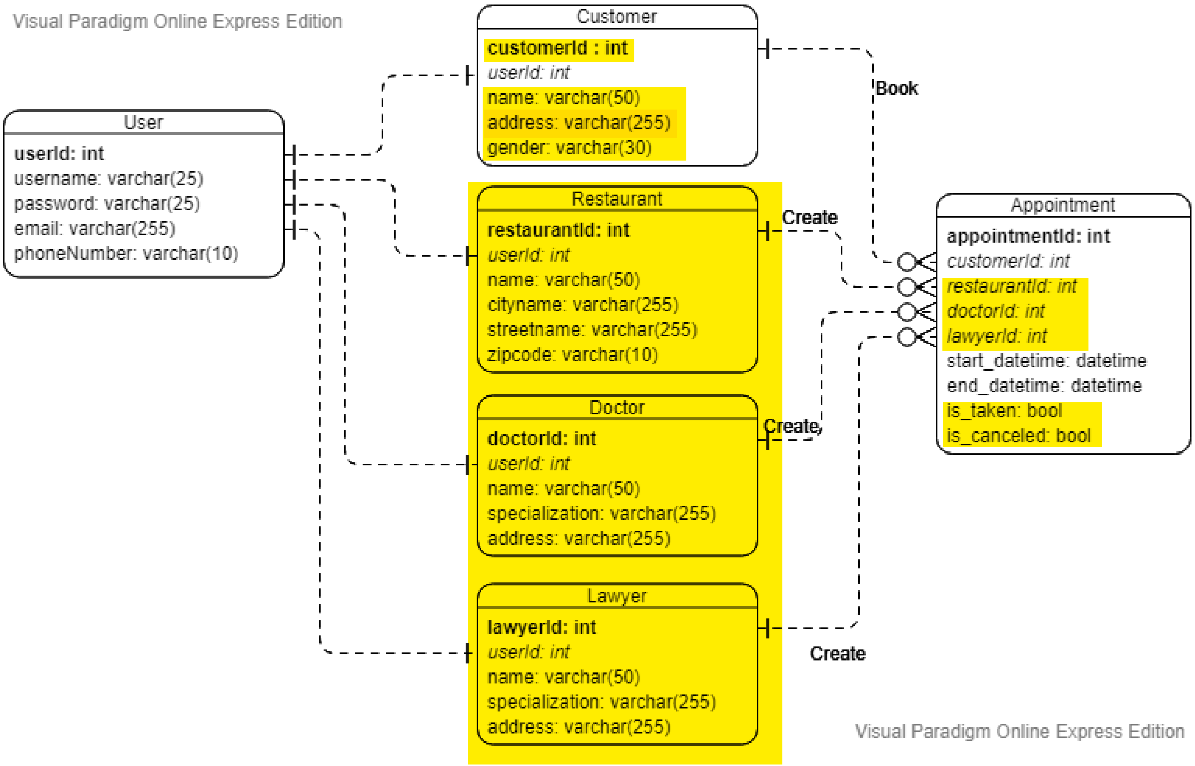

Figure 4 ER-Diagram for the APPointment web-app

Figure 4 above represents the database schema of the *APPointment*web application.

The *User* table is used to store information about registered users to be used for logging in. One user can login as customer or as one (and only one) of the rest of the actors; either a restaurant, doctor, or a lawyer. A separate table for each actor is created. A *Customer* can book multiple appointments, but a *Restaurant, Doctor,* or a *Lawyer* can create appointments which can be used as their schedule. Any appointment taken will set the “is_taken” field in the *Appointment* table to “true”. If the appointment is cancelled, the “is_taken” field will be set to “false” and “is_canceled” field will be set to “true” for later reference.

12 Runtime View

In the following figures, two scenarios are shown. In Figure 5, a user registers as a customer, logs in, chooses a doctor as a service and request to book an appointment. In Figure 6, a user registers as a Business Owner (Doctor, Lawyer, or Restaurant), logs in, and adds, deletes, or updates his available appointments.

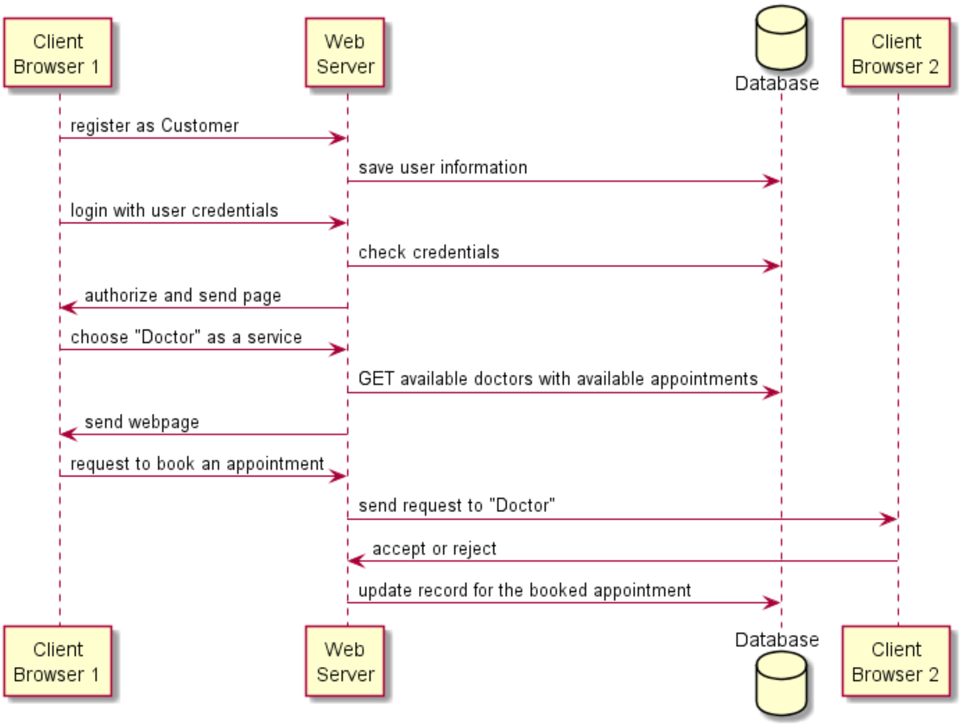

Figure 5 A sequence diagram of the web application in case the user registers as customer

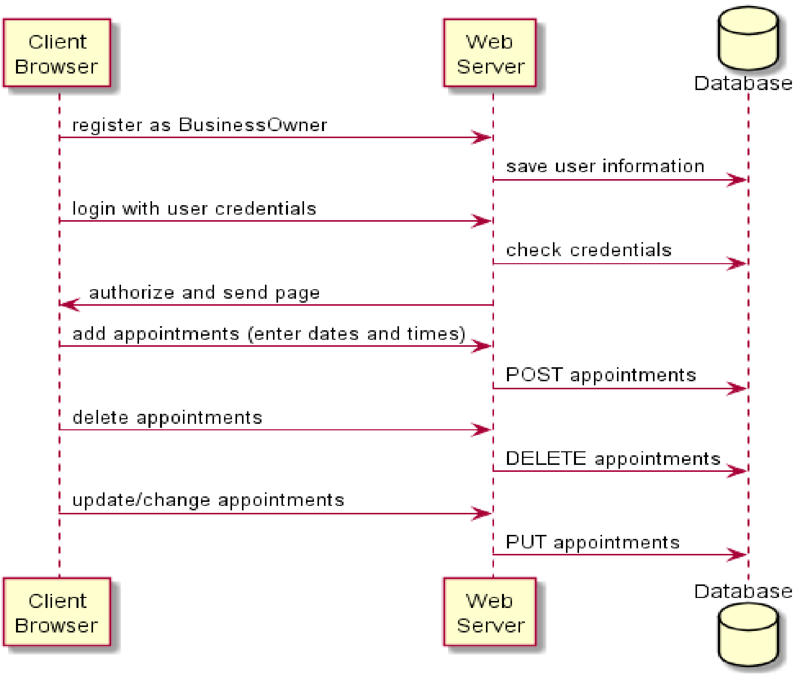

Figure 6 A sequence diagram of the web application in case the user registers as Business Owner

13 Crosscutting Concepts

crosscutting concept	**category**	**explanation and rationale**
text changes	configurability	APPointment supports minor text changes (e.g. correction of typos) to be made by the product owner without a whole redeployment of APPointment and without downtimes of any servers due to partial redeployments of configuration artifacts.
Argon2	security	We implement the Argon2 protocol and allows securing password by hashing.
web UI test	test	Tests for APPointment web UI are tested manually.
no sticky sessions	session handling	APPointment avoids server-side sessions. If necessary, session states are read and written per request from the database. Requests from a single client-side session can therefore be balanced to any application server for better scalability.
source code repository	development	APPointment source code is versioned in a single Git repository hosted on the internal GitLab xyz server.
graphic	User interface	APPointment user interface changes are made accordingly upon interest of product developer and demand of user.
advertisements	Business rules	APPointment, we have advertisements for promotions.
migration	development	APPointment, code can be migrated from one repository to other.
Disaster-recovery	operation	APPointment, we maintain regular back-ups as planned to recover from loss of data (if any).
clustering	operation	APPointment, we make the application distributable.