KAUNAS UNIVERSITY OF TECHNOLOGY

FACULTY OF INFORMATICS

INFORMATICS STUDY PROGRAMME

June 9, 2020

ALCUME

Agapito Gallart Bernat

Interactive Web Technologies Project

Supervisor:

Assoc. Prof. Sigitas Drasutis

Contents

Glossary

Alma Cuerpo Mente (Alcume) Alcume stands for Alma (soul), Cuerpo (body) and Mente (mind) in Spanish. It is a wellness-related business located in La Pobla de Segur (Spain) that offers traditional and modern treatments. These treatments are non-surgical. Also, all the treatments are in compliance with the Spanish health ministry's law.

JavaSript (JS) JavaScript, often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications.

Structured Query Language (SQL) SQL is a standard language for storing, manipulating and retrieving data in databases. SQL deviates in several ways from its theoretical foundation, the relational model and its tuple calculus. In that model, a table is a set of tuples, while in SQL, tables and query results are lists of rows: the same row may occur multiple times, and the order of rows can be employed in queries (e.g. in the LIMIT clause).

Structured Query Language Injection (SQLI) SQL injection is a code injection technique, used to attack data-driven applications, in which malicious SQL statements are inserted into an entry field for execution.

TypeSript (TS) TypeScript is an open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript, and adds optional static typing to the language. TypeScript is designed for development of large applications and transcompiles to JavaScrip.

Unified Modeling Language (UML) The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

User Interface (UI) The user interface, in the industrial design field of human-computer interaction, is the space where interactions between humans and machines occur.

1 Introduction

1.1 Importance and Relevance

Now a days, a business which has no representation on internet is declared to die (except few exceptions). Almost everyone has a smartphone in their pocket, so that it has internet connection within easy reach. This project is committed to visibilize the business called Alma Cuerpo Mente (Alcume), a local health related business.

1.2 Goal and Objectives

The objectives are:

- The web owner has to be able to post any multimedia post.
- The buyers have to be able to comment any post
- The buyers have to be able to sign up.
- The buyers have to be able to review any treatment.

1.3 Document Structure

This document contains all the investigation, design decisions and how to operate the web page. The investigation (see section 2), will be explored similar systems in order to gather a general image of how to develop the web page. Once it is gotten a general image, it's time to explore which features are needed to satisfy each goal (see section 1.2). All this specification is done in section **??**. After that, it is going to explain and endorse all the design decisions that are made.

2 Analysis

2.1 Similar system analysis

To keep the focus on what the project needs, this section is omitting all the technologies which are not considered candidates to implement in the project.

2.1.1 Udemy

Udemy [?] is an e-learning company that sells course created by third parties. According to our goals, its a really close system to look up. It's really close of what the project want to achieve. Instead of selling online course from lots of *"lecturers"*, Alcume will sell treatments from one single person or entity.

Advantages: According to Stackshare [?], Udemy uses for the front-end Angular [?], which is a framework developed and maintained by Google. Angular is really good when the web page needs to be displayed in a desktop computer and a smartphone. Udemy also, Django [?] for the back-end. Django is a really simple framework, which is really helpful to develop fast and consistent code. Django uses MySQL [?] manage the databases. This databases are written in Structured Query Language (SQL), which is really important in order to keep the data tidy. Also Django is developed in Python [?]. Which is a really easy programming language to learn and to develop with it.

Disadvantages: Angular is a framework which the curve of learning is really tilted. It is not used by developing in JavaSript (JS), it is by TypeSript (TS). Which adds more difficulties to use it properly.
Also Django, has a downsizes. The first is its performance, Django does not perform as fast as the other competitors do. But for the use case of Udemy, it is not a really big concern.

2.1.2 Fiverr

Fiverr [?] is an online marketplace for freelance services. The company provides a platform for freelancers to offer services to customers worldwide. What the project will get from Fiverr is how the information is given. It has really accessible information, there is no room for confusion. For the project that degree of clearness is how the User Interface (UI) is designed.

Advantages: According to Stackshare [?], Fiverr uses for the front-end uses React [?]. React is a JS library for building user interfaces. It is really easy to learn and to apply. React stands out for its simplicity and plain learning curve. In the back/end part Fiverr uses Ruby on Rails [?] which allow Fiverr to have good responsive. For the databases Fiverr uses MongoDB [?], a reliable distribuied database. MongoDB is a no SQL database which allows Fiverr a lot of flexibility to safe any information and redundancy in order to recover.

Disadvantages: Fiverr's stack is really solid. The stack is well design to satisfy all the needs of Fiverr. The only thing that is not optimum is the use of React as the main front-end. React is normally used to design the UI of the web page, but in Fiverr is a little bit overload on its responsibilities. But for Fiverr is well achieved.

2.2 System comparison

Both web pages are well design and both companies invested lots of resources to achieve at its finest their own goals. So it is not worth to compare both systems. Both are great solutions for their projects. But according to our requirements and needs, the best solution is to get the best from both. For the front-end it is used Angular, for the back-end, it is developed in NodeJS because it is easier to not change the language between front and back. Also, It performs better than Django and worse than Ruby. But NodeJS has more build-in elements than Ruby on Rails. For the databases, it is used in MongoDB for all multimedia part and it is used MySQL for all user data.

3 Specification

This section includes technical details related to the project. The aim of this section is to expose in all detail the insides of the web and design decisions made.

3.1 Functional requirements

According to the needs of the client, what the client desire is:

- Be able to sell the treatments online.
- The method of payment must be Pay-Pal[1].
- All the patients, can comment any post of the owner.
- The owner wants to post content on the web page.

All the given requirements can be grouped with three groups; Retailing, Media press, Communication.

3.2 Non-functional requirements

The web page needs to be responsive and connection failure tolerance. Because the patient will most of the time reading any post of the client. Also needs to be responsive to calm the patients when they are about to buy any treatment. Because of the client's market it is needed that the web page has to be simple and easy to use. Most of the clients that the client (owner) has, they are over 40 years, so it is recommend a well balance of colors and element sizes.

The colors have to transmit calm so that larger amount of potential buyers become actual buyers. Because there are movements of money between, the web page must have really strong security layers, for the logins and for the payments.

[1] If its not possible, make any alternative.

3.3 Database project

According to the client needs, the database is split into two parts; users profile and post for the MongoDB database, and the user's shopping for MySQL database. The databases' technologies are easily extensible, scalable, and high-available. The SQL database scheme is described in section 3.

The reason why the app stores the post in MongoDB is because if it contains any sort of multimedia content. It is complicated to store in a SQL database. For the users we also save in MongoDB because it hasn't got Structured Query Language Injection (SQLI) [?] problems which can compromise the user's data.
For bookings, treatments and schedules, it is used SQL because the integrity of data is vital.

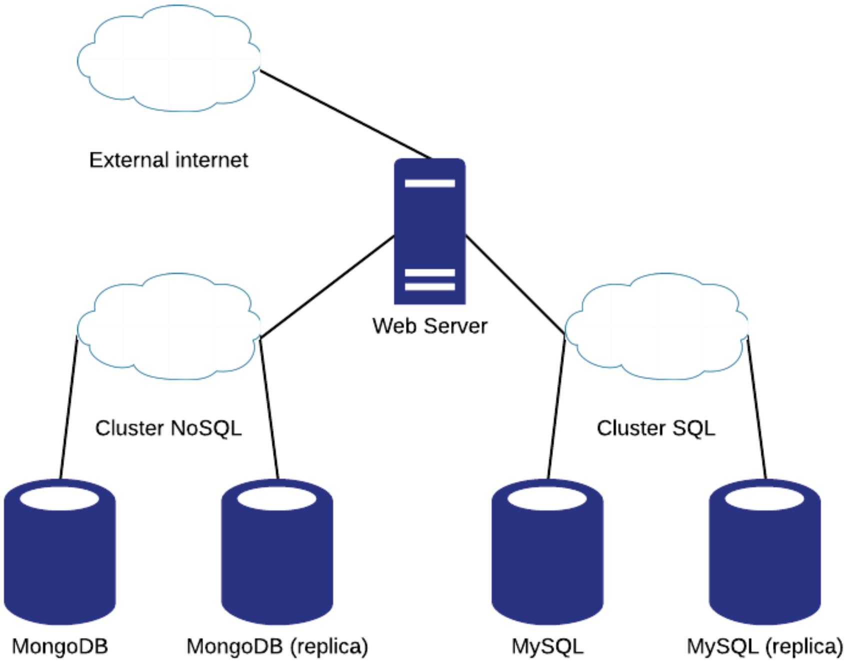

Figure 1: Database diagram

3.4 Context diagram

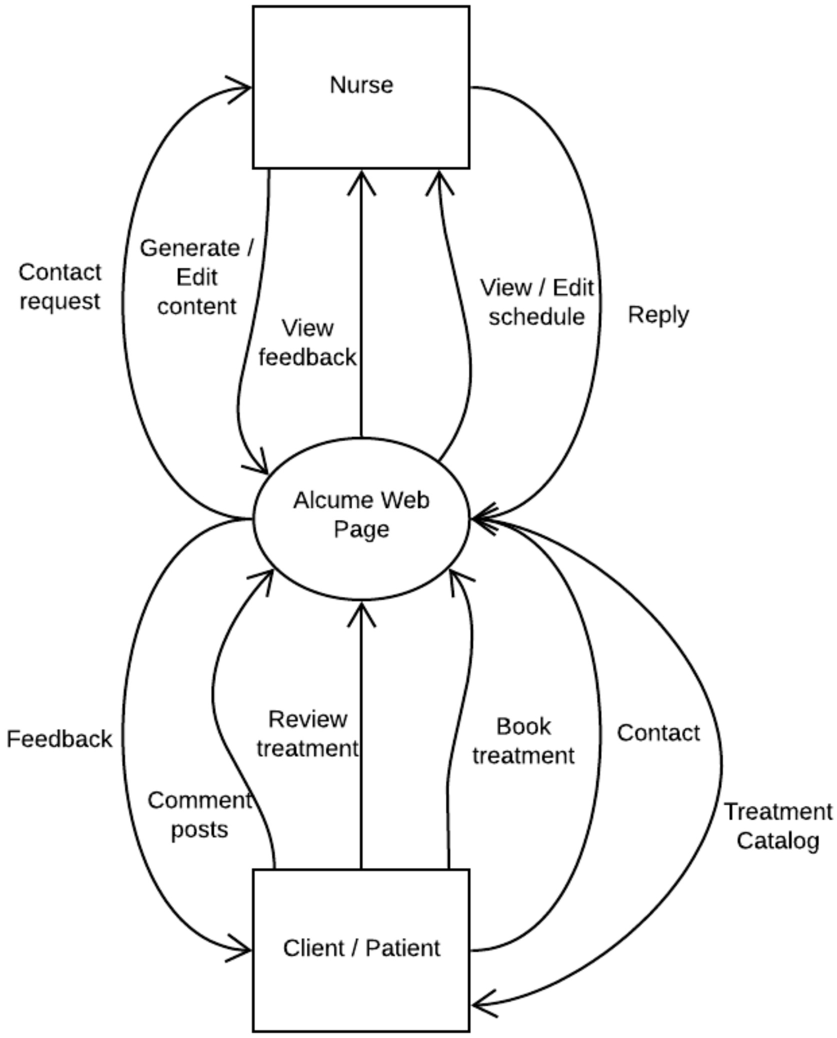

Figure 2: Context diagram

3.5 UML diagram

The design of the MySQL database is designed according to the needs of the client. The client wants to store all the data from the bookings. The bookings give points to users (patients). The reason to do is the wish of rewarding those clients who are loyal to the business. The client(business owner) desires to not store the gender of their patients due to business policies. The owner also wishes to have control of the schedule, so it can manipulate the bookings for any event. The business logic of the database is in the figure3, written in Unified Modeling Language (UML).

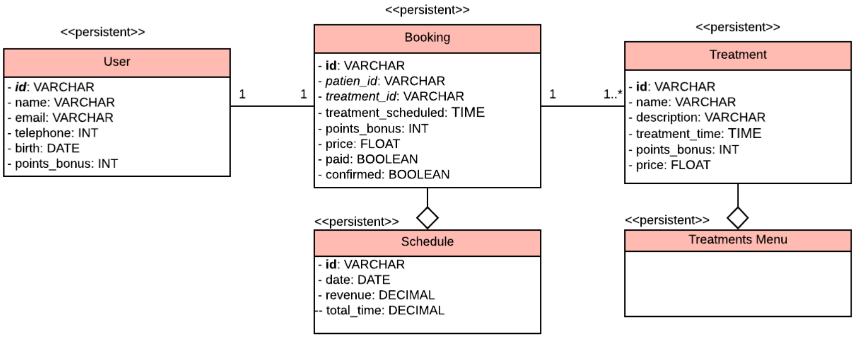

Figure 3: UML[?] diagram

The bookings are automatically saved in every step of the purchase. In an eventual disconnection, the web app will resume the booking so that the user can continue with the purchase anytime. If the user doesn't confirm[2], the purchase will be destroyed. If the booking is confirmed, it is optional if it is paid online or in cash.

[2] The user has entered all the required data but hasn't paid yet.

3.6 Use case diagram

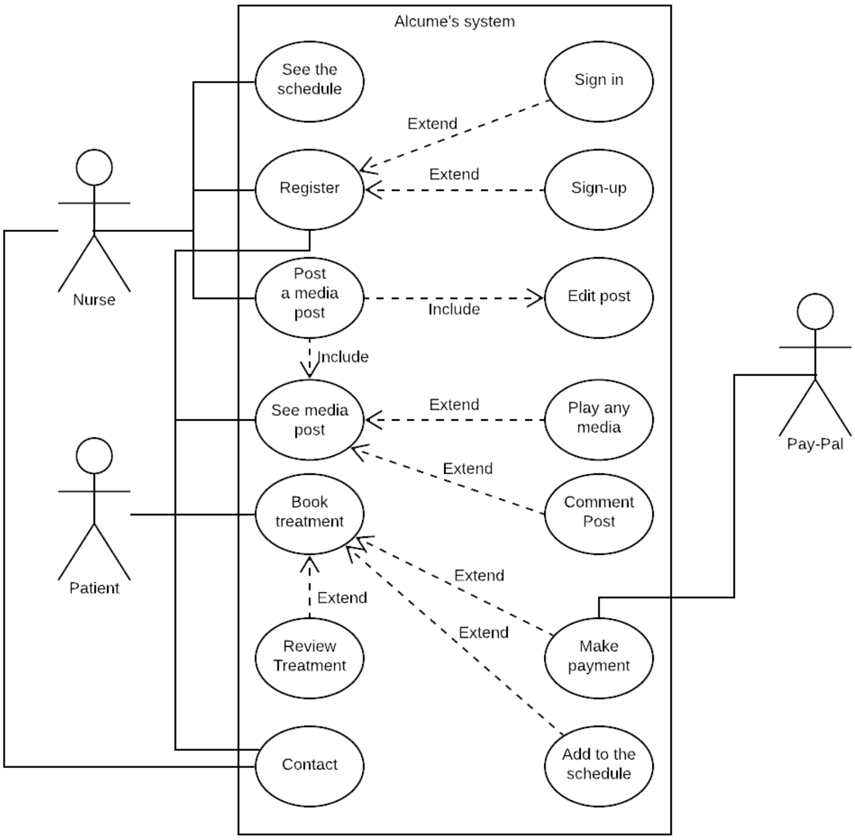

Figure 4: Use case diagram

To clarify the diagram, the nurse is our client that needs the web page. Because the nature of the business there is and will be only one nurse running the business. All the diagram hold all the needs of the owner.

3.7 Sequence diagram for Sign up

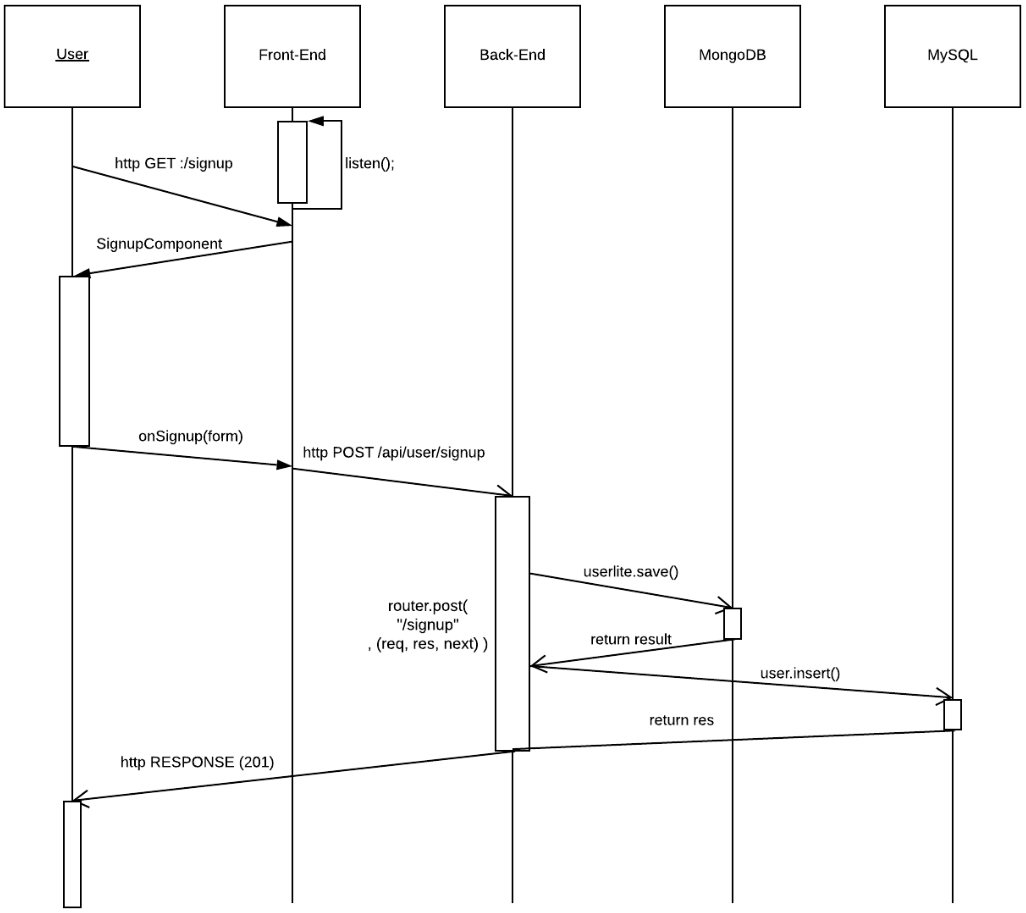

Figure 5: Sequence diagram

The reason that the app is registering in MongoDB and MySQL. In MySQL is stored the data like telephone number, birthday, etc. While in MongoDB it is stored the email and password. The id of both entries are the same. The MongoDB generates the id and its replicated in MySQL.

4 Conclusions

According to the needs of the client, what the client desire is:

- Be able to sell the treatments online.
- The method of payment must be Pay-Pal[3].
- All the patients, can comment any post of the owner.
- The owner wants to post content on the web page.

The project has fulfilled the requirements, but it is needed further development in order to make the web-page worth deploying. While the project was developing, the main problems arose were:

- Lack of skills in designing the web-page.
- Poor planning and management.
- Underestimate the learning curve in order to implement a full-stack web.
- Too big for only one developer in the given time.

[3] If its not possible, make any alternative.

5 Appendix

All the diagrams were made with `https://app.lucidchart.com`

6 Work report

Activity	Time
Topic decision	8h
Similar system investigation	9h
Finding the right technologies	8h
UI design paper	8h
UI testing paper	8h
Initial Report	4h
Semi-final report	12h
Setting up enviroments	8h
Creating front-end server	8h
Creating Back-end server	8h
Connectind Databases	8h
Creating models + routes for Front-end	8h
Creating models + routes for Back-end	8h
Authetication sevice	8h
Rating system implementation	10h
Developing services front-end routes	10h
Developing services back-end routes	10h
Cleaning code	8h
Final report	2h
TOTAL	145h

Table 1: Work time sheet