MINISTRY OF EDUCATION

TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA, ROMANIA

FACULTY OF AUTOMATION AND COMPUTER SCIENCE

System for the analysis of deviations in the behavior of the elderly - Techniques for identifying anomalies in the behavior of the elderly and generating recommendations

LICENSE THESIS

Graduate: **Cosmin - Alexandru CUIBUȘ**

Supervisor: **Associate Professor Dr. Eng. Viorica Rozina CHIFU, Senior Lecturer Dr. Eng. Cristina Bianca POP**

2022

Contents

Chapter 1. Introduction - Project Context

1.1. Introduction

According to WPP[1] 2019, by 2050 1 of 6 people in the world will be over the age of 65. This longevity revolution is affecting all societies throughout the world, with some at an early stage and others at a later level. But everyone will go through this astonishing shift, in which the likelihood of living to 65 years old goes from less than 50% in the 1890s in Sweden to more than 90% today in the countries with the highest life expectancy.

Furthermore, in most industrialized countries, the share of adult life spent beyond the age of 65 has climbed from less than a fifth in the 1960s to a quarter or more today. The population's evolving and aging structure is principally influenced by two reasons. For starters, increased life expectancy means that people are living longer longer. There has also been a drop in fertility, with people having fewer children and having them later in life. A simple chart from ONS[2] [Figure 1.1] can visually demonstrate our point about ageing population in U.K.

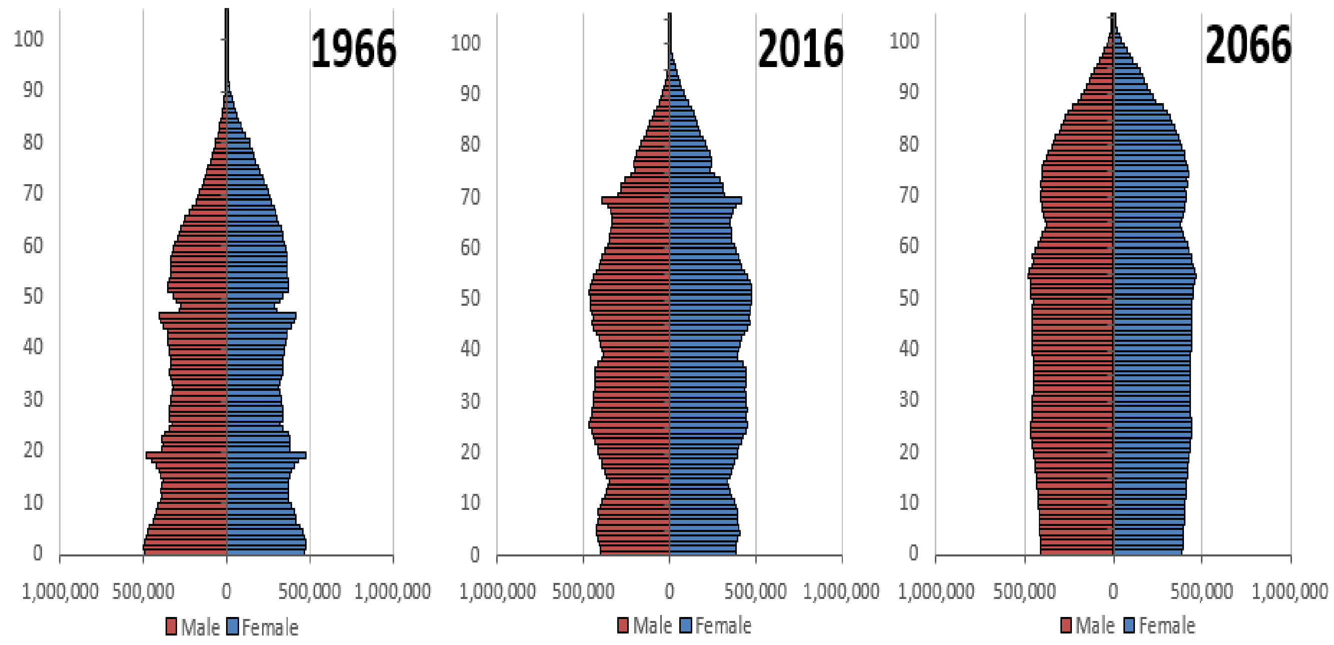


Figure 1.1: Population pyramids, 1966, 2016 and 2066 (principal projection), UK

[1]WPP - World Population Prospects
[2]Office for National Statistics

1.2. Project context

Altough life expectancy has gone up in the latest decades [Figure 1.2] this doesn't imply that the population has an optimal health.

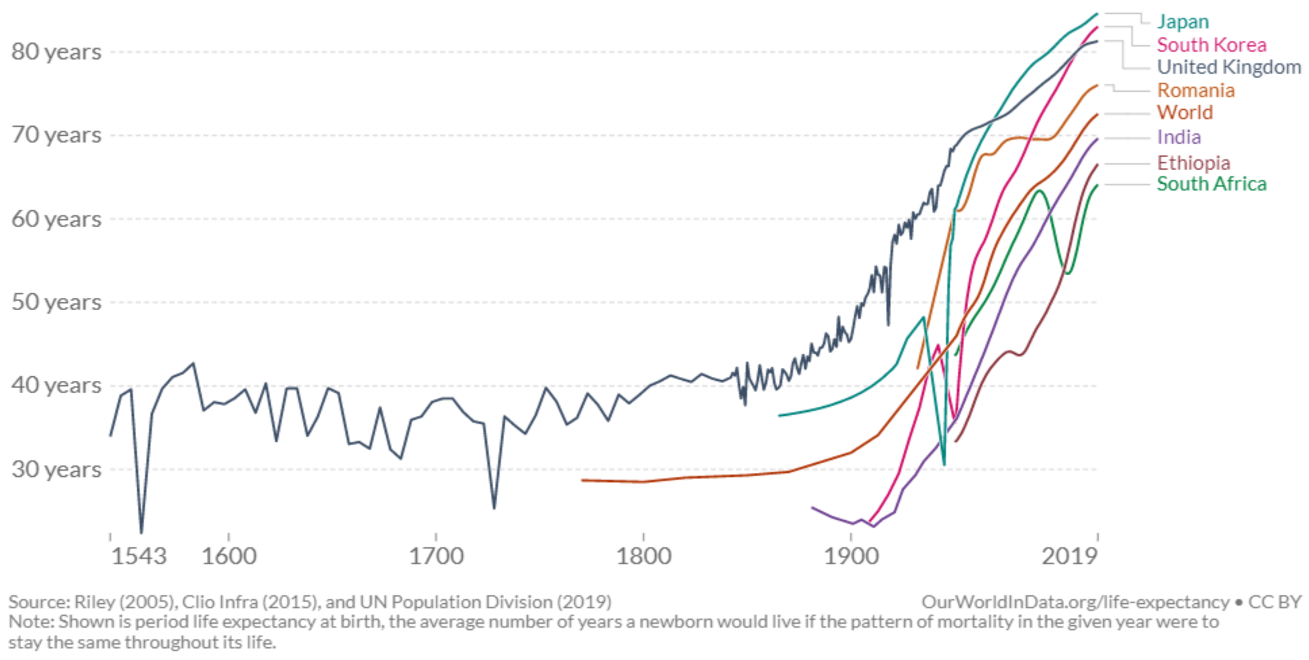


Figure 1.2: Life expectancy between 1543 and 2019 [1]

The public health community is being challenged to be proactive as the population of older adults grows, as does the demand for health and social services as the population with SCD[3] increases [Figure 1.3]. Public health professionals can attempt to limit future implications of SCD, Alzheimer's disease, and related dementias on public health and wellness by working quickly and strategically to encourage needed changes in systems and surroundings. This is especially essential because these concerns can affect not only older persons, but also their caregivers' family and friends.

The main problem we want to eliminate is the early detection of the disease by monitoring the patient without affecting family friends or caregivers but especially the patient.

Looking at the [Figure 1.4] we can see that a big percentage of people with SCD are living alone, so we can monitorize the patient without human resources and without implying the family of the patient [Figure 1.5]. With sensors we can eliminate the need of caregivers for feedback about the mental state of the patients and we can process more data more rapidly. This will change the way we deal with cognitive disorders, early stage detection has a significant impact on the treatment of the disorder. In addition we are just overseeing the data without disturbing daily life living of elder people.

[3]Subjective Cognitive Decline is the self-reported experience of worsening or more frequent confusion or memory loss. It is a form of cognitive impairment and one of the earliest noticeable symptoms of Alzheimer's disease and related dementias.

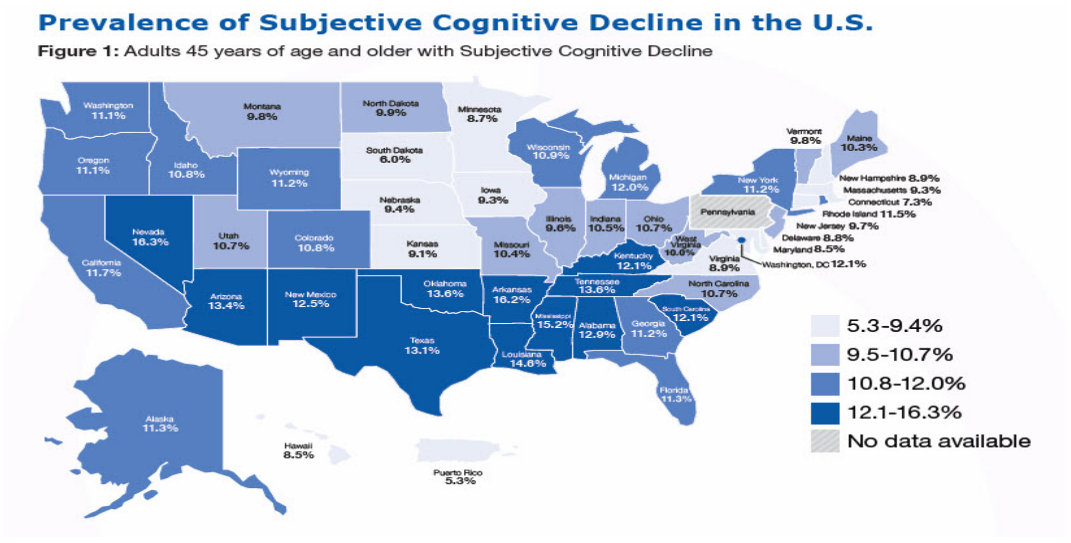


Figure 1.3: Adults 45 years of age and older with SubjectiveCognitive Decline [2]

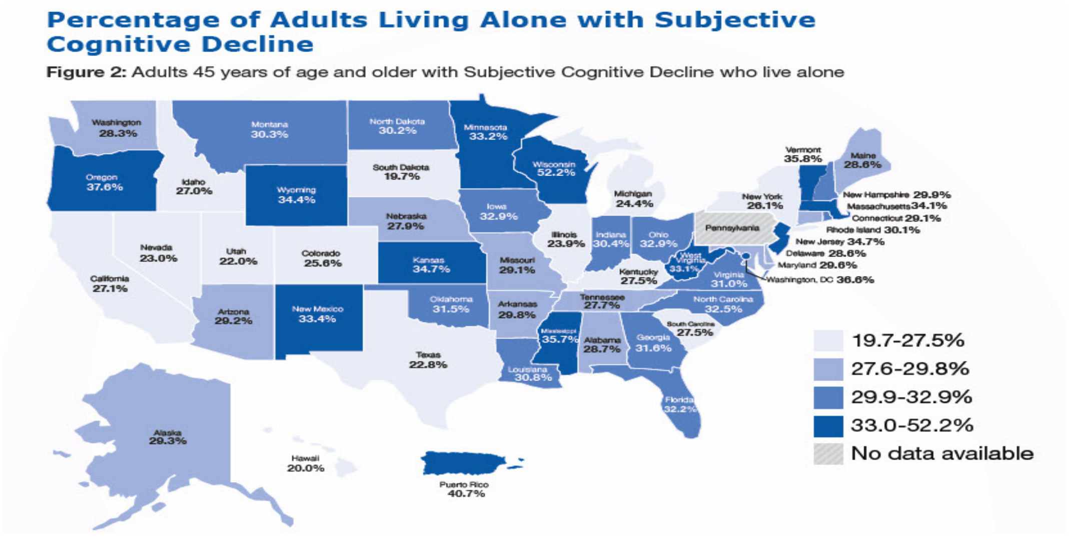


Figure 1.4: Adults 45 years of age and older with SubjectiveCognitive Decline who live alone [2]

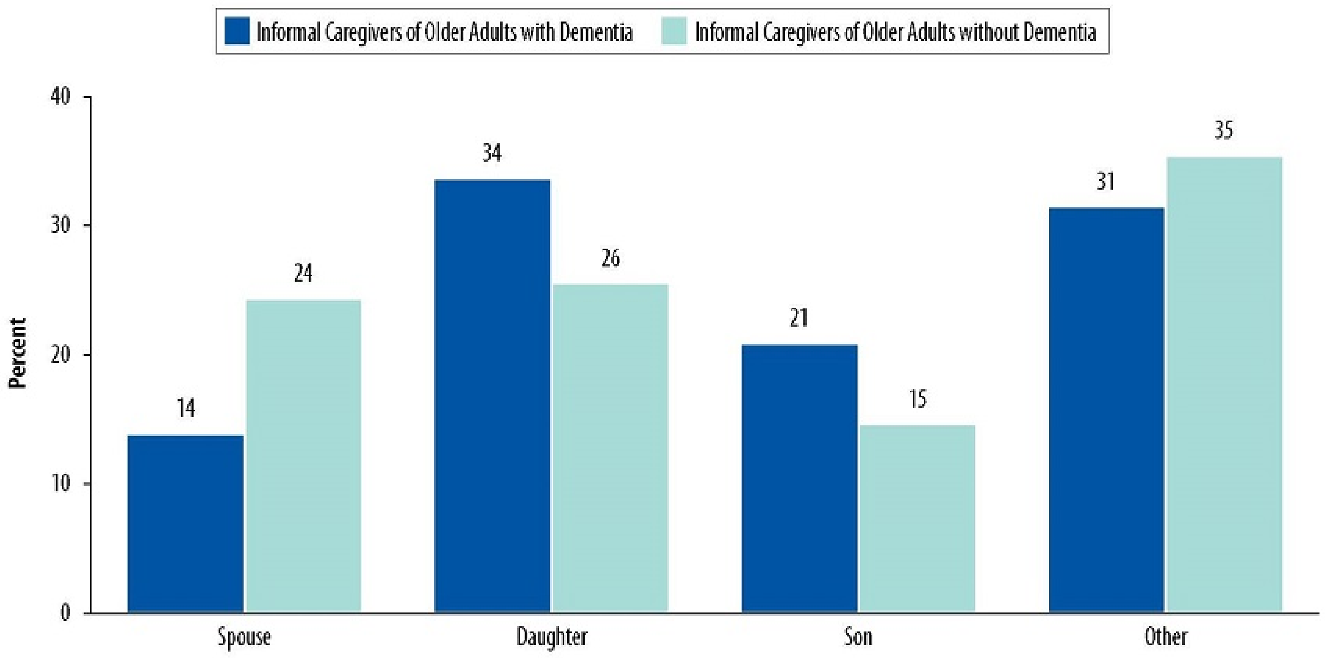


Figure 1.5: Caregivers of patients with and without Dementia [3]

1.3. Contents

In this section we will shortly resume the contents of the paper:

1. **Chapter 1 - Introduction** - The first chapter contains the introduction and and the theme of the paper.
2. **Chapter 2 - Project objectives** - Here we will present the main and secondary objectives of the paper.
3. **Chapter 3 - Bibliographic research** - This chapter will highlight others papapers and other approaches to our theme.
4. **Chapter 4 - Analysis and Theoretical Foundation** - Here we will analyze and set the theoretical foundation needed for the approaches we used.
5. **Chapter 5 - Detailed Design Implementation** - This chapter will put under microscope the design and architecture we approached and also how we implemented the approach.
6. **Chapter 6 - Testing and Validation** - Here we presented some of the result and metrics we kept account of in the papaer.
7. **Chapter 7 - User Manual** - The chapter highlights how to install and start al the components so you can benefit from all the features the application has to offer.
8. **Chapter 8 - Conclusions** - Here we will write the conslusions we have about the paper.

Chapter 2. Project Objectives

In this chapter, we will offer a description of the primary and secondary objectives that are to be reached in the development of the project and which ensure the realization of a system with simple and efficient use.

2.1. Main objective

The main objective of this project is to create a web application that is user friendly and supports the aging of older adults. The application will monitor the daily activities performed by older adults to extract information about their sleep and medication status. Additionally, the platform will incorporate self-reporting features allowing older adults to provide insights upon their mood status with the help of questionnaires. Each questionnaire has a set of well defined questions as well as well defined answers, each answer having a preset score.

The scores of the questionnaires are later reflected in the recommendations the patients are receiving. The system should simulate the collection of data from the sensors by reading from a file with sensor reading at certain intervals. Also the system should generate recommendations for patients that will improve thei daily life and mental health.

2.2. Secondary objectives

As secondary objectives we can have:

1. Studying and comprehending the objectives of daily life analysis, which could prevent and revolutionize the medicine in the field of early combating the cognitive problems in elder people and why not in the entire population.

2. Putting together the web application and combining it with our anomaly detection model is currently being worked on. We are going to create a user-friendly react application for medical professionals, carers, and patients. With nothing more than a web browser and an active internet connection, medical professionals will soon be able to monitor their patients' behavior to detect any unusual deviations from the norm.

3. Building a real time data generator that mimics real life sensor data from elder people / patients households.

4. Making recommendations using machine learning and helping patients based on data we analyze from the sensors in their households.

5. Help medical institutions worldwide to be more efficient about data storage and gathering about patients with cognitive disorders.

6. Let patients fell they are cared about by providing them informations about their lifestyle and help them have a better daily routine.

Chapter 3. Bibliographic Research

The IoT became a part of our life. From smartphones to smartwatches, home assistants and different types of sensors that can make our life easier within the four walls of our home. So why don't we use this to monitor and help elder people. This subject has been deeply researched in the past years, so the doctors can prevent cognitive distrubings in the life of elder people from a really early stage. The concept behind this prevention is detection. As soon as the illness is detected, the doctors can prevent the mental alteration of the patient. But how do we detect this? In this chapter we will present some approached strategies for deviation detectiona and also for recommendations for elder people.

By taking into account sub-activity relations, the authors of paper [4] make use of the Hidden State Conditional Random Field (HCRF) method in order to identify anomalous behaviors. These activities frequently take place in the homes of elderly people. The first step in the process of recognizing activities is for the HCRF to generate a recognition confidence value for each action.

The actions are then evaluated with a system based on thresholds to determine whether or not they are normal or pathological. By utilizing a Markov Logic network, the authors of [5] are able to identify abnormalities associated with moderate cognitive impairment. They employ a technique that combines supervised learning, rule-based reasoning, and probabilistic reasoning to achieve their goals.

Having said that, they begin the process of building their model by specifying the individual steps of each action. It takes a lot of time and effort to define these rules since they are very dependent on the unique home environment, the particular sensors that are utilized, and the individual behaviors of the old people. Furthermore, rules cannot be transferred to other situations.

The same authors suggest a way to automatically learn the rule-based descriptions of behavioral abnormalities [6], which is an effort to address the problem that was presented earlier. They make use of formal rule induction methods in conjunction with a training set of normal and deviant behaviors. However, the authors claim that their proposed rule learning method infers deterministic rules, which are prone to generate anomaly mispredictions in the presence of noise from the sensor infrastructure.

Some studies [7], [8] concentrate on anomalies related to the duration and timing of activities that were performed, but they do not take into account other types of anomalies associated to dementia, such as the repeating of activities. Activity curves are introduced by the authors in the article [7].

These activity curves model an individual's generalized daily activity patterns based on activities that are automatically detected. Comparing activity curves in order to conduct a health evaluation enables the detection of disruptions in the typical patterns of behavior. The authors of utilize a probabilistic model in their work [8], which is based on the position and outing interference of each activity.

The cross-entropy measure is then applied in order to identify anomalies, such as being in bed for an extended period of time or not utilizing the bedroom as a place to sleep during the night. The authors of [9] use HMM in conjunction with fuzzy rules in order to identify duration, time, and frequency related abnormalities.

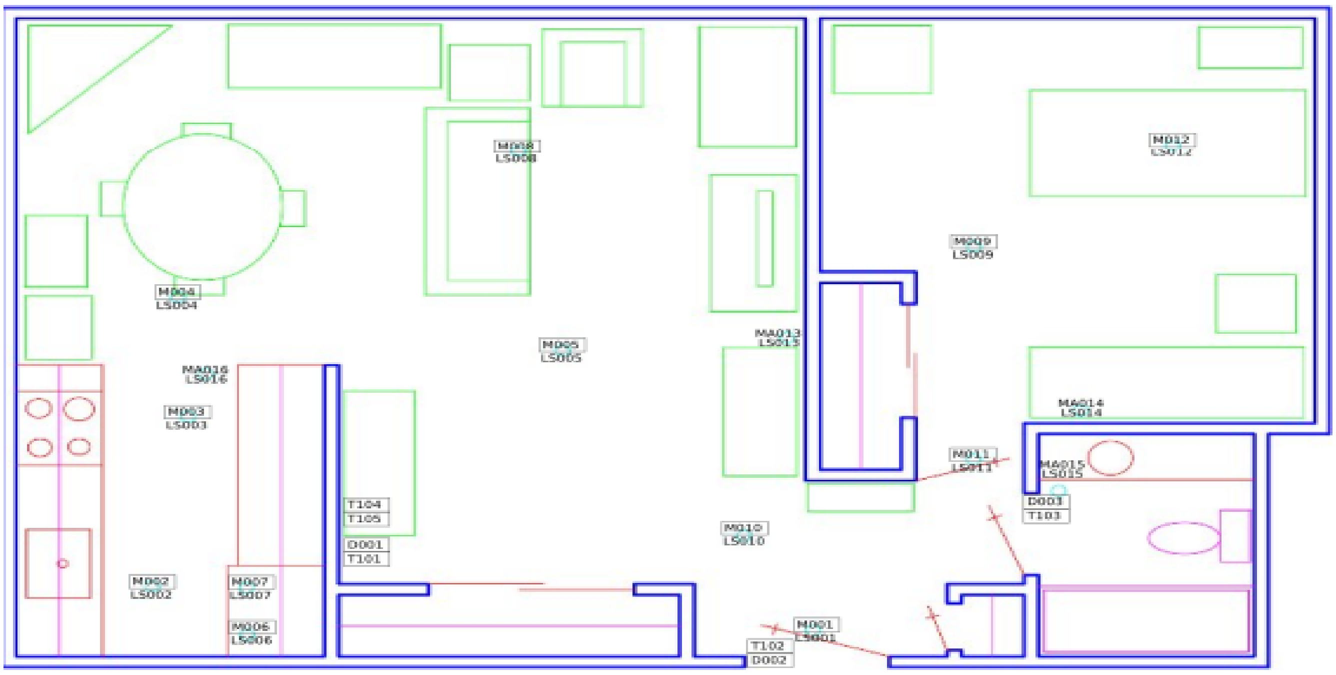


Figure 3.1: House plan and sensor deployment [7]

In several studies [10], [11] determining a patient's cognitive status is accomplished by having the patient carry out a series of predetermined tasks while being given specific instructions (for example, mopping the kitchen). In the end, patients are given scores that are calculated based on a variety of factors including the amount of time spent, the number of times the sensor was triggered, etc.

The cognitive health of elderly persons can be evaluated using these scores. The cognitive deterioration of elderly adults is evaluated in [10] by having them go through a series of programmed instrumental activities of daily living (IADLs). While the volunteers complete activities like making oatmeal on the stove, they are watched by a camera and, at the end of the experiment, they are given ratings by people who have received specialized training. The authors of paper [11] first extract sensor-based data, which include the duration of the activity and the number of sensors that were triggered, and then utilize SVMs, NB, and neural networks to evaluate the activity quality and cognitive status of older persons living in smart homes.

However, participants are given a brief description of each sub-task that they should refer to while engaging in the simulated activities. For example, participants can be asked to plan a bus route or locate a recipe in the recipe book. Since these studies are not conducted in the normal flow of daily life or in situations that actually occur, it is impossible for them to provide an unobtrusive method of assessment. In addition, while employing rule-based systems, an expert is required to manually integrate specific rules into the system. This is due to the fact that every individual possesses their own unique day-to-day routines.

For instance, one person may consider it usual for them to get up and drink water in the middle of the night, while another individual may consider same behavior to be abnormal. Our method, on the other hand, does not necessitate any prior understanding of the subject, as it can automatically determine what is normal and what is odd based on the data that it is trained on. In particular, the purpose of this research is to identify abnormalities in the normal flow of everyday life without the use of any instructions and taking into account not just a certain amount of time gap, but also a typical day's worth of activities.

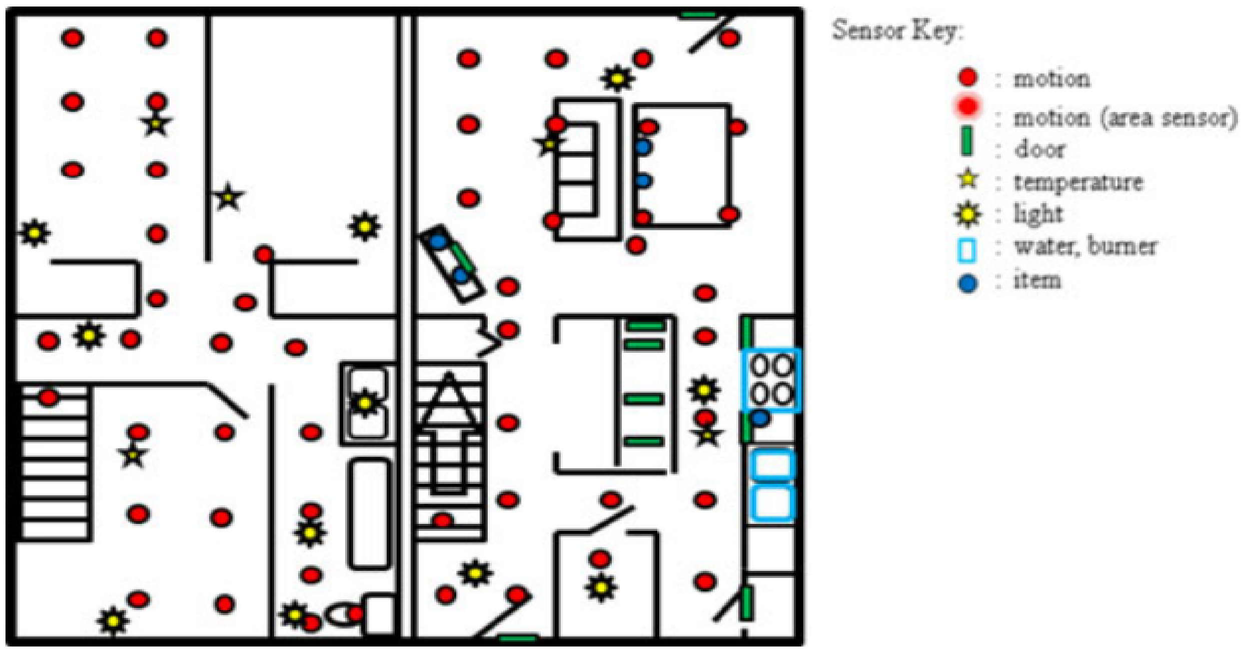


Figure 3.2: House map and legend [11]

The technologies behind recommender systems stretch back to the nineties [12], [13], and its primary purpose is to create customised product suggestions based on previously recorded data on consumers' interests [14]. Recommender system technologies trace back to the nineties. These days, recommender systems are an accepted and widespread technology that are used by many market leaders in a variety of industries (for example, Amazon , Netflix, and Spotify).

The field of recommender systems has had significant development over the years, which has resulted in the production of incredibly complex and specialized approaches that vary according to domain, goal, and level of personalisation [15]. In contrast to more traditional approaches to machine learning, recommender systems are not only able to handle the difficulties connected with the data that has to be processed, but they may also provide insight into the decision-making process, which makes the findings interpretable. It comes as a surprise that recommender systems have not yet found widespread use in the medical field.

Within the realm of recommendation algorithms, a fundamental taxonomy distinguishes between content-based [16], collaborative filtering [14], and hybrid techniques [17]. Each method employs either explicit or implicit past ratings as expressions of preference in order to turn estimates of a user's liking for products into recommendations. This is the one thing that all of these methods have in common. In contrast to the content-based approach, which establishes a connection between a user's preference and the characteristics of an item, the collaborative filtering method takes into account the ratings that other users have given to items within the system in order to generate personalized predictions regarding a user's preference.

Similarity metrics are the foundation of the underlying algorithms for rating prediction and recommendation computation. These metrics are able to process sparse and inhomogeneous data vectors because of their metric structure. In addition, interpretation and explanation of the advice can be offered by displaying the respected data subset as well as the effect factors. Finally, the databases that underpin such systems may be quickly modified and expanded, which allows for the ongoing adaptation of the system to new impact factors and the context in which it is implemented [15]. [18].

Chapter 4. Analysis and Theoretical Foundation

The functional and non-functional needs of the proposed application will be laid out in this chapter from both a functional and a non-functional point of view, respectively. They are also shown the most important use cases.

4.1. Analysis

In this section we will analyze some of the use cases of the application

4.1.1. Use Case

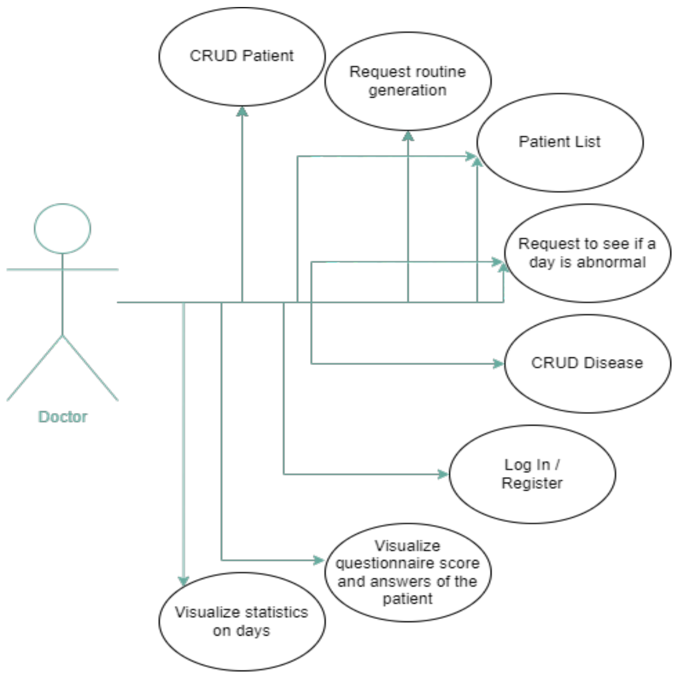

CRUD Patient
Request routine generation
Patient List
Request to see if a day is abnormal
CRUD Disease
Log In / Register
Visualize questionnaire score and answers of the patient
Visualize statistics on days
Doctor

Figure 4.1: Doctor use case

As an example we will choose the use case for the Request to see if a day is abnormal. The primary actor of this use case is the doctor, and the main flow is as follows:

1. The doctor requests a routine generation, based on which the day chosen will be compared and returned as deviated or not.
2. The doctor inputs a valid date for a day that he wants to be checked if it is deviated/abnormal.

3. The doctor sends the request for anomaly detection.
4. The doctor receives some information about the the requested day as well as the information if the day is deviated or not.

An alternate sequence where the doctor does not input a date, would look like this:

- The doctor requests a routine generation, based on which the day chosen will be comapred and returned as deviated or not.
- The doctor does not input a date.
- The doctor tries to send the request for anomaly detection.
- The doctor gets a warning for the empty date input and he returns to step 2.

Another use case example is when the doctor wants to login

1. The doctor enters valid credentials, credentials that he created when he registerd on the application.
2. The doctor click the login button
3. The doctor is allowed to enter the application and he is redirected to a page that he has access to.

An alternate sequence for the above use case can be:

- The docotor enters inavlid credentials.
- The doctor click the login button
- The doctor is notified that something is wrong with the credentials so he returns to step 1.

For the third example we will take the case when the doctor wants to register a new patient in the application

1. The doctor enters valid data about the patient
2. The doctor click the register button so the patient is registered in the database doctor is redirected to the patien view where he can see the newly inserted patient.

An alternate sequence for the third example can be:

- The doctor enters valid data about the patient
- The doctor click the register button so the patient is registered in the database
- The doctor is notified that something is wrong with the resgister of the patient so he returns to step 1.

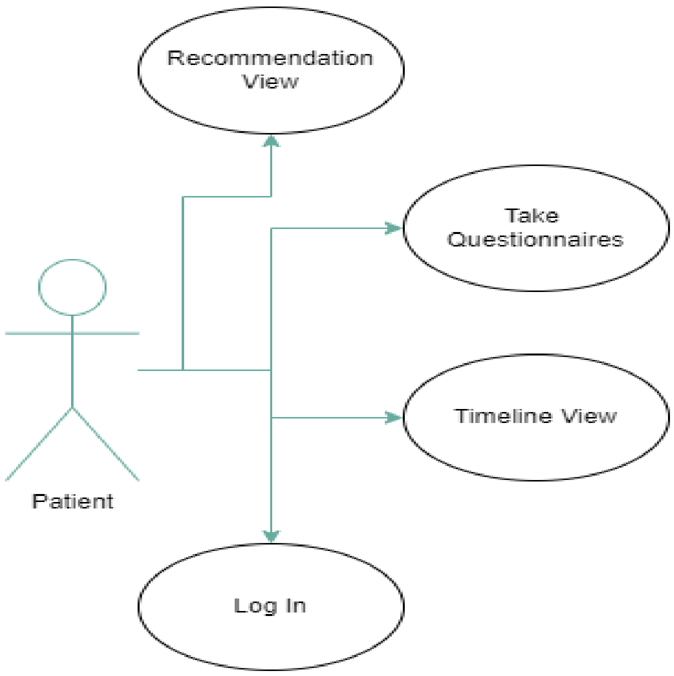


Figure 4.2: Patient use case

As an example for the patient's use case we will take the situation where the patient wants to take a questionnaire. The primary actor of this use case is the patient and the sequence is as follows:

1. The patient goes to the questionnaire page.
2. The patient starts the questionnaire.
3. The patient answers questions.
4. The patient submits the questionnaire.

An alternate sequence where the patient presses submit without responding to a question would be.

- The patient goes to the questionnaire page.
- The patient starts the questionnaire.
- The patient answers questions.
- The patient submits the questionnaire without answering the question, so he gets a warning and goes back to step 3.

Another use case example would be when the patient wants to view the timeline:

1. The patient enters a valid interval of time.
2. The patient clicks the view timeline button.
3. The timeline is populated with activities that took place in the chosen interval.

An alternate flow of the above example is:

- The patient enters a non-valid interval of time.
- The patient clicks the view timeline button.
- The timeline is not populated so the user goes back to step 1

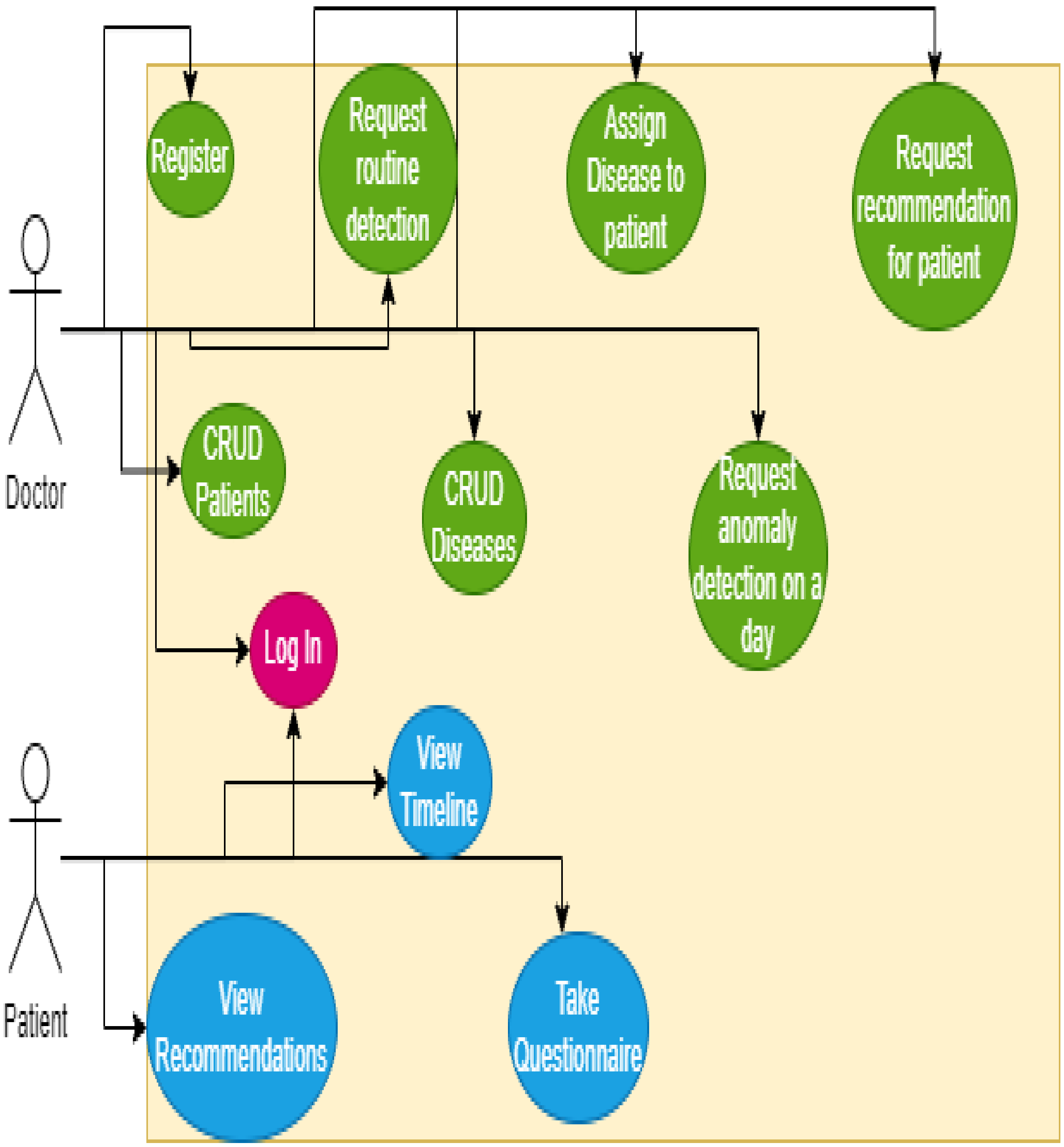

Register
Request routine detection
Assign Disease to patient
Request recommendation for patient
Doctor
CRUD Patients
CRUD Diseases
Request anomaly detection on a day
Log In
View Timeline
Patient
View Recommendations
Take Questionnaire

Figure 4.3: Use Case using both actors

4.1.2. Requirments

Requirments for a software applciation can be split in Functional requirments and Non-Functional Requirments:

Functional Requirments

Something that is needed in order for the system to function properly. In the event that the system does not satisfy a specific functional condition, the evaluation will be deemed unsuccessful. This is owing to the fact that it will be unable to complete a task that it has to do in order for it to work appropriately.

An additional way for acquiring a knowledge of the functional requirement concept is to examine the system in terms of its inputs and outputs. The functional requirements detail not only what the system is anticipated to do in response to a variety of inputs but also what the system is expected to create.

Non-Functional Requirments

In software engineering, the term ”non functional requirements” refers to requirements that define the operation of the system rather than its functions. Requirements that are not functional are concerned with the manner in which the system performs a certain function. Both of these aspects, despite the fact that they can appear to be of less significance at first look in comparison to functional criteria, are necessary for the development of a quality system.

The functionality of the system is unaffected by the non-functional criteria, although the requirements do have an effect on how well the system will work. In a nutshell, non-functional criteria are concerned exclusively with the usability of the system.

For our project we can define some Functional Requirments as well as Non-functional Requirments:

Functional Requirments

- The application should allow the doctor log in.
- The application should allow the doctor to register.
- The application should allow the doctor to insert patients.
- The application should allow the doctor to edit patients.
- The application should allow the doctor to delete patients.
- The application should allow the doctor to view patients.
- The application should allow the doctor to insert diseases.
- The application should allow the doctor to view diseases.
- The application should allow the doctor to to edit diseases.
- The application should allow the doctor to delete diseases.
- The application should allow the doctor to assign diseases to patients.
- The application should allow the doctor to request a routine detection for a specific user.
- The application should allow the doctor to request to see if a certain day is deviated.

- The application should allow the doctor to see statistics about certain days from the dataset of a user.
- The application should allow the doctor to see the evolution of a patient through questionnaires.
- The application should allow the patient to log in.
- The application should allow the patient to see the recommendations for multiple days.
- The application should allow the patient to complete questionnaires regarding his mental health.
- The application should allow the patient to view his timeline on a certain period of time.

Non-Functional Requirments

- The application should be intuitive and easy to use by both the doctor and the patient.
    - The program ought to have the highest potential level of user friendliness. It is essential that it be developed in such a way that its use does not call for any specialized knowledge or skills relating to computers. Abstracting the algorithms that detect anomalies and hiding them behind a user interface is recommended.
- The security of the application.
    - Because of the nature of the research, concerns relating to individuals' right to privacy should be anticipated. Therefore, the system ought to provide increased security so as to defend against the possibility of a malicious actor collecting personal information about a patient's day-to-day activities. In addition to that, the authentication and authorisation procedures need to be brought up to date with the most recent security requirements.

4.1.3. Sequence Diagram

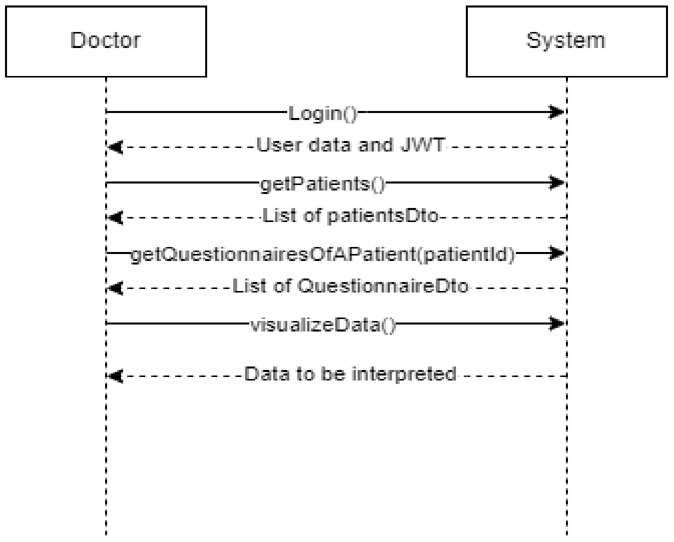


Figure 4.4: Doctor sequence diagram

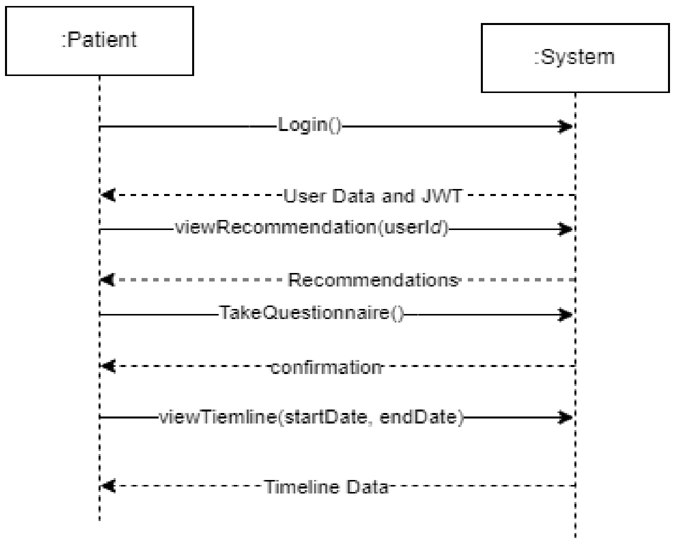


Figure 4.5: Patient sequence diagram

4.2. Theoretical Foundation

Machine learning is a branch of Artificial Intelligence that permits software to be more precise at forecasting output despite having been specifically designed to act in a certain manner.

Machine learning can be of multiple types as follows:

- Supervised or unsupervised
  - Classification
  - Regression

Supervised and unsupervised machine learning are working using different methods. The supervised machine learning can be used when we have a target field opposite from the unsupervised one that uses the data to explore by itself and find some type of structure. As a simple example for supervised ML we can have a labels that categorize one type of object that was labeled by human input, emails can be labeled by human attendence with social or promotions.

In the unsupervised category it is frequently used to group clients whose features or behaviors are comparable to those in segmented marketing campaigns. Another common application of this technique is to analyze customer data.

The subfield of machine learning known as Supervised Learning encompasses both the classification and regression types. One of them is making a prognosis regarding a category or label, while the other one is making a prediction regarding a number.

The example with emails that are labeled so they can be included in one of the two categories: social or promotions. On the contrary the regression machine learning predicts a number for example how much a flight will be in the month of July next year.

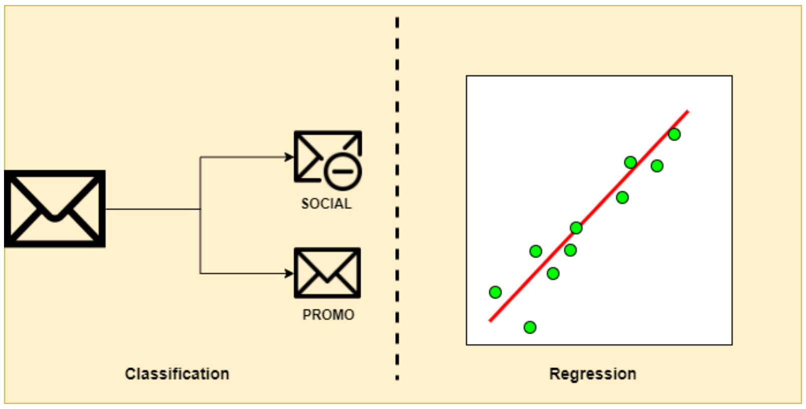


Figure 4.6: Classification vs Regression

4.2.1. Decison Trees

In the given section we will talk about the concept of Decision Trees also called The Classification and Regression Trees. They function by learning the answers to a series of if/else questions that lead to a choice.

The name comes from the fact that these questions take the form of a tree. Let's take the situation where we want to forecast if someone will order meals or not. For this, we can see the decision tree in the figure 4.2:

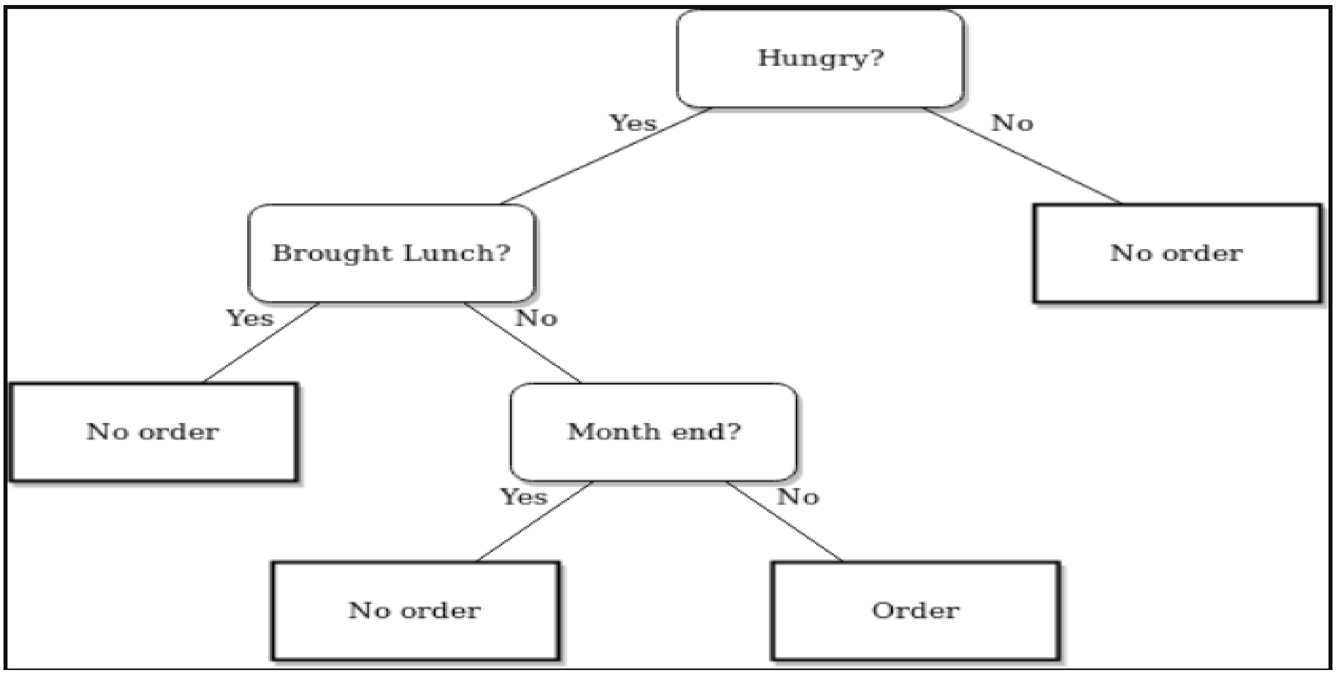


Figure 4.7: Simple decision tree [19]

A question produced from the characteristics in your dataset is represented by each node in the tree. Up until the maximum depth of the tree is achieved, your dataset is divided depending on these questions. The last node indicates to which class the value belongs rather than posing a query.

- The Decision tree's uppermost node is referred to as the Root Node
- The lowest node is referred as as the Leaf Node
- A node with sub-nodes is referred to as a Parent Node
- The name of the sub-nodes is Child Nodes

Based on the goal variables, decision trees are divided into two categories

1. Decision Trees for Categorical Variables: This approach uses a categorical target variable. Consider the scenario when you are asked to choose between the low, medium, or high pricing range for a car. Features might include HP, Acceleration, and speaker quality in addition to manufacturer. After sending each data point through each node of the decision tree, which will learn from these attributes, it will finally arrive at a leaf node of one of the three category objectives, low, medium, or high.
2. Continuous Variable Decision Trees: In this scenario, a continuous output will be predicted using the features provided to the decision tree (such as characteristics of

a vacation) (e.g. the price of that vacation).

Decision trees present some advantages as well as some disadvantages as follows:

Advantages:

- Easy to comprehend, interpret, and visualize.
- Decision trees automatically screen variables or choose features.
- Can manage data that is both numerical and categorical. may also manage issues with many outputs.

Disadvantages:

- Overly complicated trees that poorly generalize the input can be produced by decision-tree learners. Overfitting is the term for this.
- If some classes predominate, decision tree learners will produce biased trees. As a result, it is advised to balance the data set before fitting it to the decision tree.

4.2.2. Naive Bayes Classifier

For classification tasks, a probabilistic machine learning model known as a Naive Bayes classifier is typically utilized. The Bayes theorem is the foundation of the classification system's most important component.

The Bayes theorem states: $P = (A|B) = \frac{P(B|A)P(A)}{P(B)}$
The Bayes Theorem is a mathematical formula that determines the likelihood of one event happening given the likelihood of another event that has already taken place.

Several advanteges of this classifier are:

- This method has a very rapid running time and is able to accurately determine the category of a test dataset.

- You can put it to use in the resolution of problems involving multiple classes of prediction because it is highly useful in those contexts.

- If the assumption that features are independent is correct, the Naive Bayes classifier has superior performance to that of other models that require less data for training.

A number of disadvantages of this classifier are:

- The Naive Bayes model will assign it zero probability and won't be able to make any predictions in this regard if your test data set contains a categorical variable that belongs to a category that wasn't present in the training data set. This is because the training data set didn't contain this variable. This occurrence is referred to as "Zero Frequency," and in order to rectify the situation, you will need to implement a smoothing strategy.

- It makes the assumption that each of the characteristics can exist independently. Although it may be appealing to consider in concept, in practice, there is rarely such a thing as a collection of stand-alone characteristics.
  Some applications of this classifier are as follows:
- You are able to utilize this algorithm to generate predictions in real time because it is both quick and effective.

- This technique is often used for making predictions across multiple classes. Utilizing this approach makes it simple to determine the probabilities of a number of different target classes.

- This algorithm is utilized by email services to determine whether or not a certain email is considered spam. This algorithm does quite well when it comes to screening out spam.

Naive Bayes classifiers are of different types:

- Bernoulli Naive Bayes
- Gaussian Naive Bayes
- Multinomial Naive Bayes

4.2.3. Deviation detection and recommendation generator

Using Decision Trees to generate recommendations for the patients

In order to make a recommendation regarding the actions that must or mustn't be taken by a certain patient we will use Decision Trees to solve the problem. This problem is raised when multiple parameters must be taken into account are added to the decision tree.

The categorical variables type of tree will be very useful in this approach we will use, since we have inout parameters that we need to keep account of. After the decision tree has reached the leafs node (the final node of the tree) we will apply a naive bayes classifier on the results.

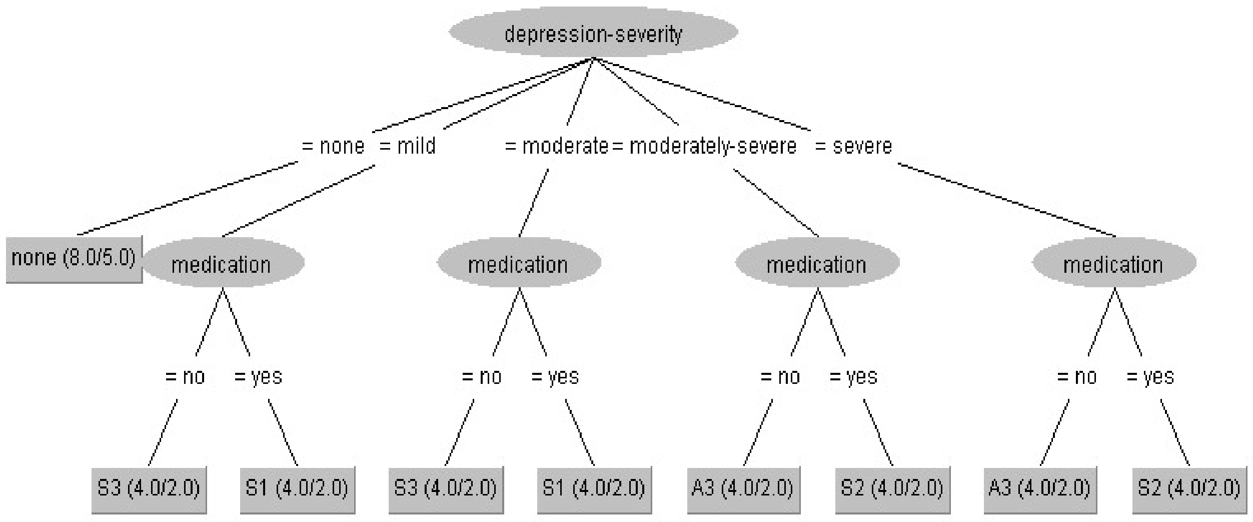


Figure 4.8: Simple decision tree

```
@relation recommendation

@attribute depression-severity
{'none', 'mild', 'moderate', 'moderately-severe', 'severe'}
@attribute sleep                { '0-2', '3-5', '6-8', '9+'}
@attribute medication    { 'no', 'yes'}
@attribute recommendation
{   'S1', 'S2', 'S3', 'A1', 'A2', 'A3', 'A4','none'}

@data
'none', '0-2', 'no', 'S2'
'none', '0-2', 'yes', 'S1'
'none', '3-5', 'no', 'A1'
'none', '3-5', 'yes', 'S1'
'none', '6-8', 'no', 'S3'
'none', '6-8', 'yes', 'none'
'none', '9+', 'no', 'S3'
'none', '9+', 'yes', 'none'
'mild', '0-2', 'no', 'A1'
'mild', '0-2', 'yes', 'S1'
'mild', '3-5', 'no', 'A1'
'mild', '3-5', 'yes', 'S1'
'mild', '6-8', 'no', 'S3'
'mild', '6-8', 'yes', 'none'
'mild', '9+', 'no', 'S3'
'mild', '9+', 'yes', 'none'
'moderate', '0-2', 'no', 'A1'
'moderate', '0-2', 'yes', 'S1'
'moderate', '3-5', 'no', 'A1'
'moderate', '3-5', 'yes', 'S1'
'moderate', '6-8', 'no', 'S3'
'moderate', '6-8', 'yes', 'none'
'moderate', '9+', 'no', 'S3'
'moderate', '9+', 'yes', 'none'
'moderately-severe', '0-2', 'no', 'A3'
'moderately-severe', '0-2', 'yes', 'A2'
'moderately-severe', '3-5', 'no', 'A3'
'moderately-severe', '3-5', 'yes', 'A2'
'moderately-severe', '6-8', 'no', 'A4'
'moderately-severe', '6-8', 'yes', 'S2'
'moderately-severe', '9+', 'no', 'A4'
'moderately-severe', '9+', 'yes', 'S2'
'severe', '0-2', 'no', 'A3'
'severe', '0-2', 'yes', 'A2'
'severe', '3-5', 'no', 'A3'
'severe', '3-5', 'yes', 'A2'
'severe', '6-8', 'no', 'A4'
'severe', '6-8', 'yes', 'S2'
'severe', '9+', 'no', 'A4'
'severe', '9+', 'yes', 'S2'
```

We determine the type of advice that should be given to a patient by classifying them based on the variables depression, sleep hours, and medicine. If we look at the first row of data in the dataset, we can see that the patient does not suffer from any form of depression, that they fall into the category of sleeping 0-2 hours a day, and that they do not use any form of medicine. Given this information, the prediction ought to be the label. S2.

In order to illustrate the point more clearly, the data set consists of a feature matrix as well as a response vector. Depression, the number of hours that were slept, and the presence or absence of medication are the three components that make up the feature matrix in our dataset. The value of the class is contained within the response vector.

X = (none, 0-2, no) => y = S2

Cosine similarity

The cosinusoidal similarity index assesses the degree to which two vectors that belong to the same space are similar to one another. It is determined whether two vectors indicate about the same direction by measuring the cosine of the angle formed by the two vectors and comparing the results. In the field of text analysis, it is frequently applied to the task of measuring the degree to which different papers are alike. A document may be represented by hundreds of attributes, each of which records the number of times a specific term (such a keyword) or phrase appears in the text.

Therefore, each document is an object represented by what is called a term-frequency vector. For example, in Table 4.1, we see that Document 1 contains five instances of the word ”sleep,” while ”breakfast” takes place only once. The word ”picnic” is absent from the entire document, as indicated by a count value of 0. Such data can be very asymmetrical. Term frequency vectors are usually very long and rare (ie they have many values of 0).

Document	sleep	picninc	breakfast
Document 1	5	0	1
Document 2	2	1	1
Document 3	4	0	1

Table 4.1: Example of a word-based frequency vector

Applications such as information retrieval, the grouping of text documents, biological taxonomy, and gene feature mapping all make use of such structures. For instance, two term-frequency vectors may have many 0 values in common, which indicates that the associated texts do not share many words; however, this does not mean that the papers are similar. We require a metric that focuses on the words that are shared by the two texts as well as the frequency with which those words appear in both of the documents. To put it another way, we require a measurement for numeric data that disregards zero-to-zero matches. The following is the formal definition of the similarity function:

$$sim(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}$$

where ||x|| is the Euclidean norm of the vector x = (x1, x2,..., xp), defined as $\sqrt{x_1^2 + x_2^2 + ... + x_p^2}$. Conceptually, it represents the length of the vector. Similarly, ||y|| is the Euclidean norm of the vector y.

The function calculates the cosine of the angle between the vectors x and y. A cosine value of 0 means that the two vectors are at 90 degrees and have no match. The closer the cosine value is to 1, the smaller it is the angle and therefore the fit between the vectors is greater.

Standard deviation The standard deviation can be thought of as a measurement of "dispersion" or "spread." This one is utilized as a standard overview of the whole range of scores linked with it. To calculate it, first add up all of the square values of the deviation that each observation had on average, then divide that number by the entire number of observations, and finally take the positive square root of the result. For instance, we have several different metrics.:

13, 6, 12, 10, 11, 9, 10, 8, 12, 9

Their sum is 100 and therefore their median is 10. The median deviations are:

3, 4, 2, 0, 1, 1, 0, 2, 2, 1

In order to calculate the standard deviation, we must first square the deviations, which causes them to become positive values. This then leads to the:

9, 16, 4, 0, 1, 1, 0, 4, 4, 1

The total square deviations add up to 40, and the average of them (calculated by dividing by the total number of measurements) is 4. This is referred to as the "Variance" of the initial numbers, and the "standard deviation" is the square root of the positive number, which is equal to 2. By taking the square root of the numbers, we are able to arrive at values that are on the same scale as the initial entries. According to the findings of an analysis, the average of these ten numbers is 10, and their standard deviation is 2. This latter statistic provides us with a sense of how dispersed the original statistics are and, as a result, how representative the average is. The primary reason why the standard deviation was calculated in this manner was so that it would be easier to work with algebraically than if one were to simply ignore the sign of negative deviations. This was accomplished by using the square, which gets rid of any negative deviations that may have occurred.

An alternative to the standard deviation as a measure of dispersion is the absolute mean deviation. This is merely an average of the absolute disparities that each score possesses in comparison to the overall average. The following is a list of the individual metrics:

13, 6, 12, 10, 11, 9, 10, 8, 12, 9

Their sum is 100 and therefore their average is 10. Their deviations on average are:

3, 4, 2, 0, 1, 1, 0, 2, 2, 1

In order to calculate the average deviation, we must first disregard the minus signs next to the negative integers that represent the deviations. This allows us to get rid of the negative values, which results in:

3, 4, 2, 0, 1, 1, 0, 2, 2, 1

These numbers now indicate the distance between each observation and the average, despite the fact that the differences can go either way. Their total is 16, and their average (calculated by dividing the total by the number of readings) is 1.6.

The average deviation is actually more useful than the standard deviation in situations that are more realistic and in which some of the measurements are incorrect. It is also more useful for other distributions besides the perfectly normal distribution, is closely related to a number of other helpful analytical techniques, and is easier to understand.

Chapter 5. Detailed Design and Implementation

This chapter will evaluate the system's design, focusing on the web application's architecture and the integration of the web application and recommendation generating system. The technologies that were chosen will be presented in the second section, with an emphasis on the key implementation choices.

5.1. Design

We'll give a broad overview of the web application design in this part. The conceptual architecture of the complete system is shown first. The design of each component is then examined in greater detail.

5.1.1. Architecture

The whole project is composed from 5 different parts: Front-End, Back-End, DataBase, Recommendation Generator and Producer. The communication between the components include HTTP requests using a REST API and also asynchronous communication through CloudMQ.

From the architecture point of view, we used a Client - Server architecture where the logic is included in the Server side. By using an interface, such as the Front-End, the system's complexity is abstracted away.

Advantages of the Client-Server architecture

- The centralized network has total control over the activities and operations.
- Data administration is centralized.
- Only in response to a request is data made available.
- Information exchange is the goal.
- Both small and large networks can use it.
- Security provided by JWT (JSON Web Token)

Because of the full-stack impelmentation, we need to store information and data that we will use later. The necesity of relationships between the entitites, different privilege users and recommendations we resorted to a relational database.

In the next subsections we will take a closer look to each component, and spotlight the decisions and architecture of each one of them.

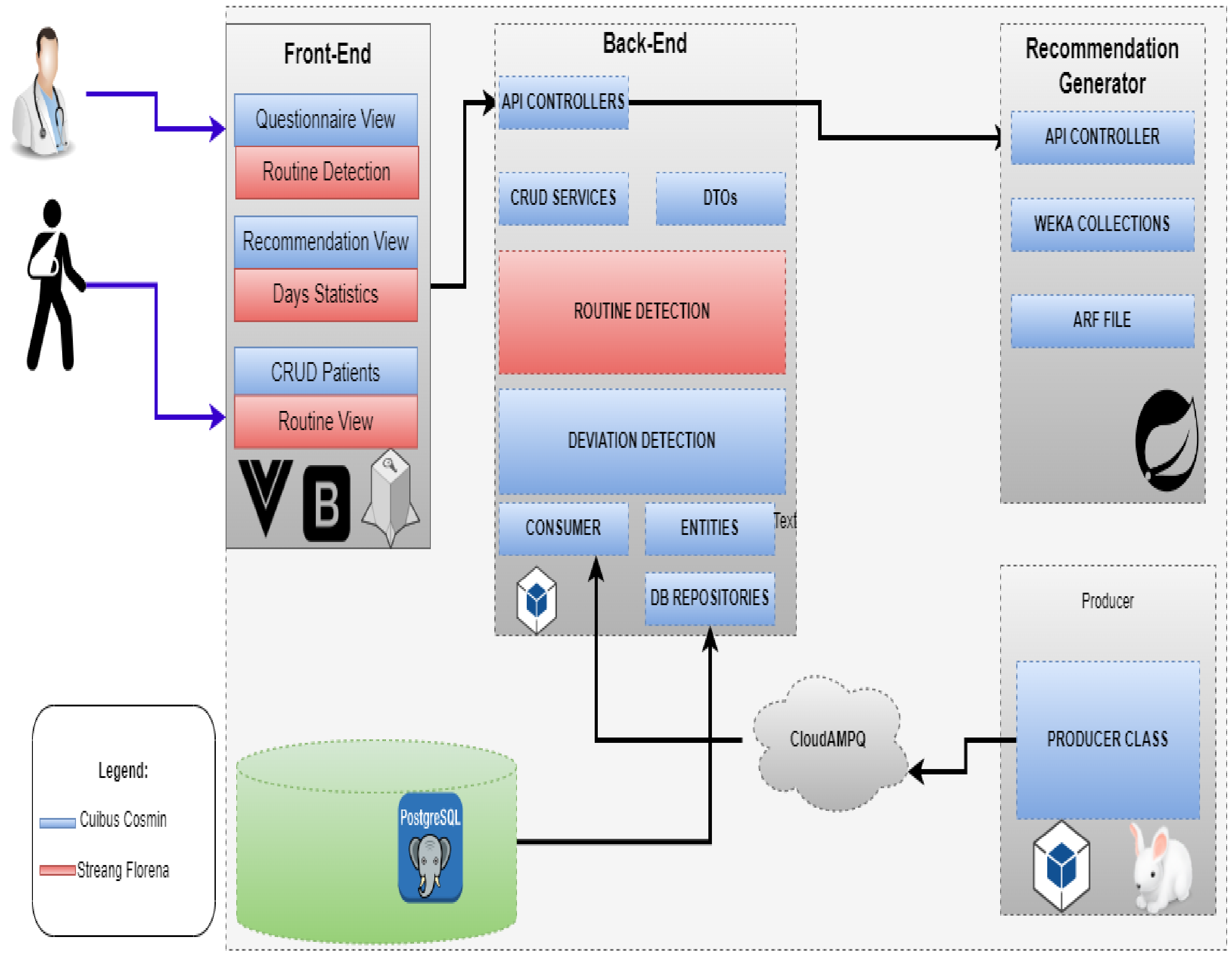


Figure 5.1: System Architecture

5.1.2. Front-End

Regarding the Front-End architecture we chose a simple SPA (Single Page Architecture). With no page reloads or added wait times, SPAs strive to provide an excellent UX by mimicking a "natural" environment in the browser. You only need to visit one web page, and JavaScript—which they strongly rely on—will then be used to load all subsequent information.

SPA renders pages directly in the browser after separately requesting the markup and data. Thanks to cutting-edge JavaScript frameworks like AngularJS, VueJs, React, we are able to do this. Single-page websites assist in keeping the user in a single, pleasant digital environment where content is presented in a straightforward, user-friendly, and practical way.

Disadvantage

- Because large client frameworks must be loaded onto the client, the download is slow.
- JavaScript must be present and turned on. If a user disables JavaScript in their

browser, the program and its activities cannot be displayed correctly.
- SPA is less secure than the ”conventional” application. Attackers can inject client-side scripts into online applications used by other users thanks to cross-site scripting (XSS).

An SPA uses the full capability of AXIOS and JavaScript to build the page, HTML is injected with data resulted from the communication with server side.

Advantages of SPA
- SPAs load quickly because the majority of resources (HTML, CSS, and scripts) are only loaded once during the course of the application. There is no other communication besides data.
- No code needed to render pages on the server.
- Easily debugging with Chrome devtools and also framework related tools.
- Any local storage can be successfully cached using SPA. An program sends a single request, stores all the data, and then uses that data to function even when offline.
- There are two independent apps the communicate through HTTP.

The security concern of making unauthorized requests to the server side is solved by using a JWT in each request header that authorizes or not the request from the server side.

5.1.3. Back-End

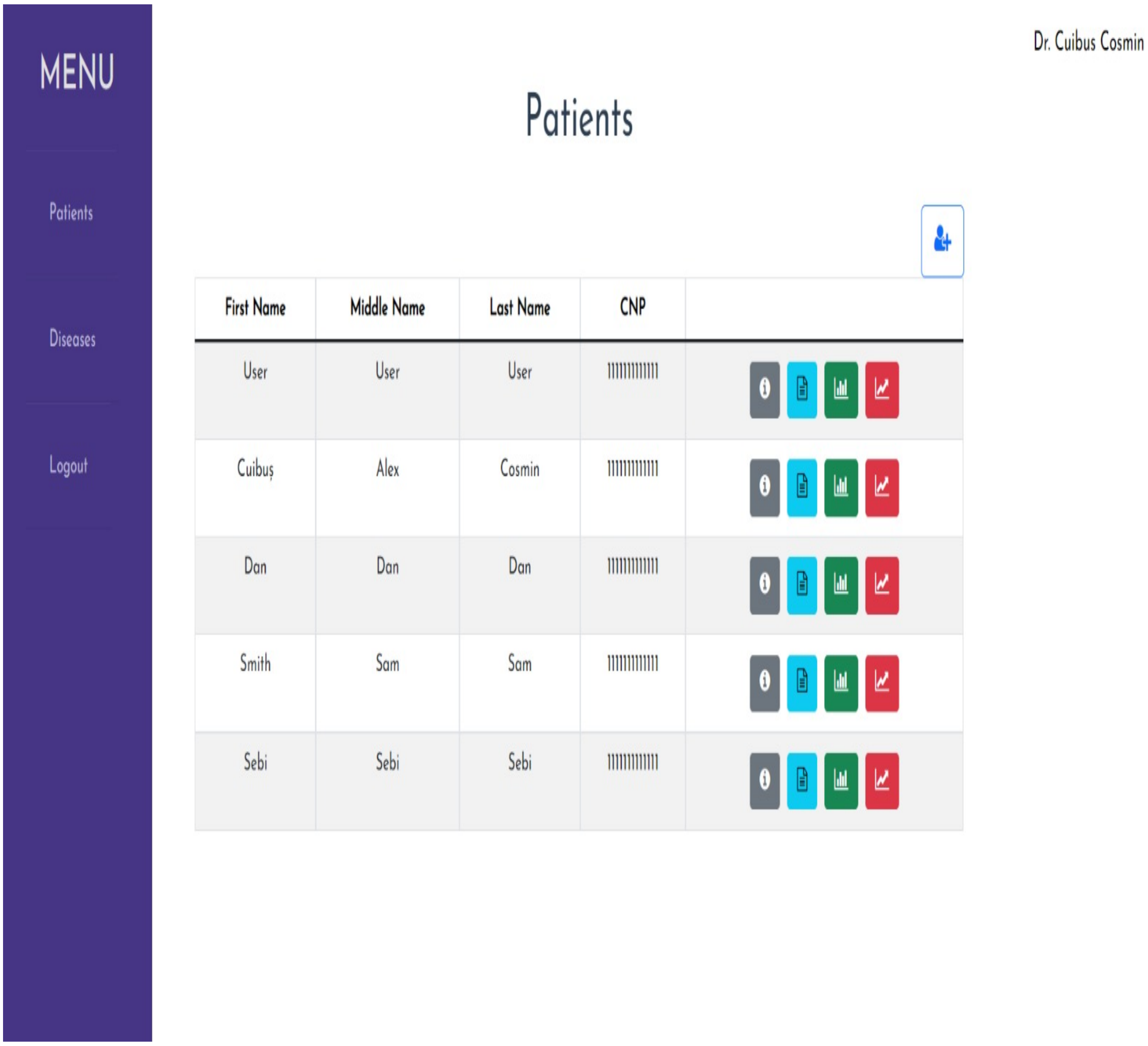
5.1.3.1 App Design

All the logic and data manipulation is operated by the back-end application, from this rule is excepted the Recommendation Generator and the Producer. The app handles the data saving for users also the necesarry data for routine detection and recommendations generation.

A layered architecture is used in the back-end, with the presentation layer, business logic layer, and data access layer being the three main layers. The controllers are the application’s entry point; they receive HTTP requests, parse the arguments and headers, and then forward the request to the business logic to be processed further. They are located in the presentation layer.

The business logic layer manipulates all the data and operations. To improve the architecture we used a facade service design pattern.

The data layer has two types of classes, the entities that are used between the repositories to interogate the Database and the DTO’s (Data Transfer Objects) that tranfer data from the respository to the services and back to the controller so it can be sent in the client side. The ORM(Object Relational Mapper) builds a mapping between the classes from the application and the tables of Database.



Patients

First Name	Middle Name	Last Name	CNP	
User	User	User	1111111111111	
Cuibuș	Alex	Cosmin	1111111111111	
Dan	Dan	Dan	1111111111111	
Smith	Sam	Sam	1111111111111	
Sebi	Sebi	Sebi	1111111111111	

Figure 5.2: Doctor view on patients

Because of its ease of management and simplicity, we chose to employ a layered design. Furthermore, by avoiding a direct connection between the presentation layer and the data access layer, we substantially strengthened the security of the application by using this architectural approach. This makes it possible to prevent common attacks like SQL injections.

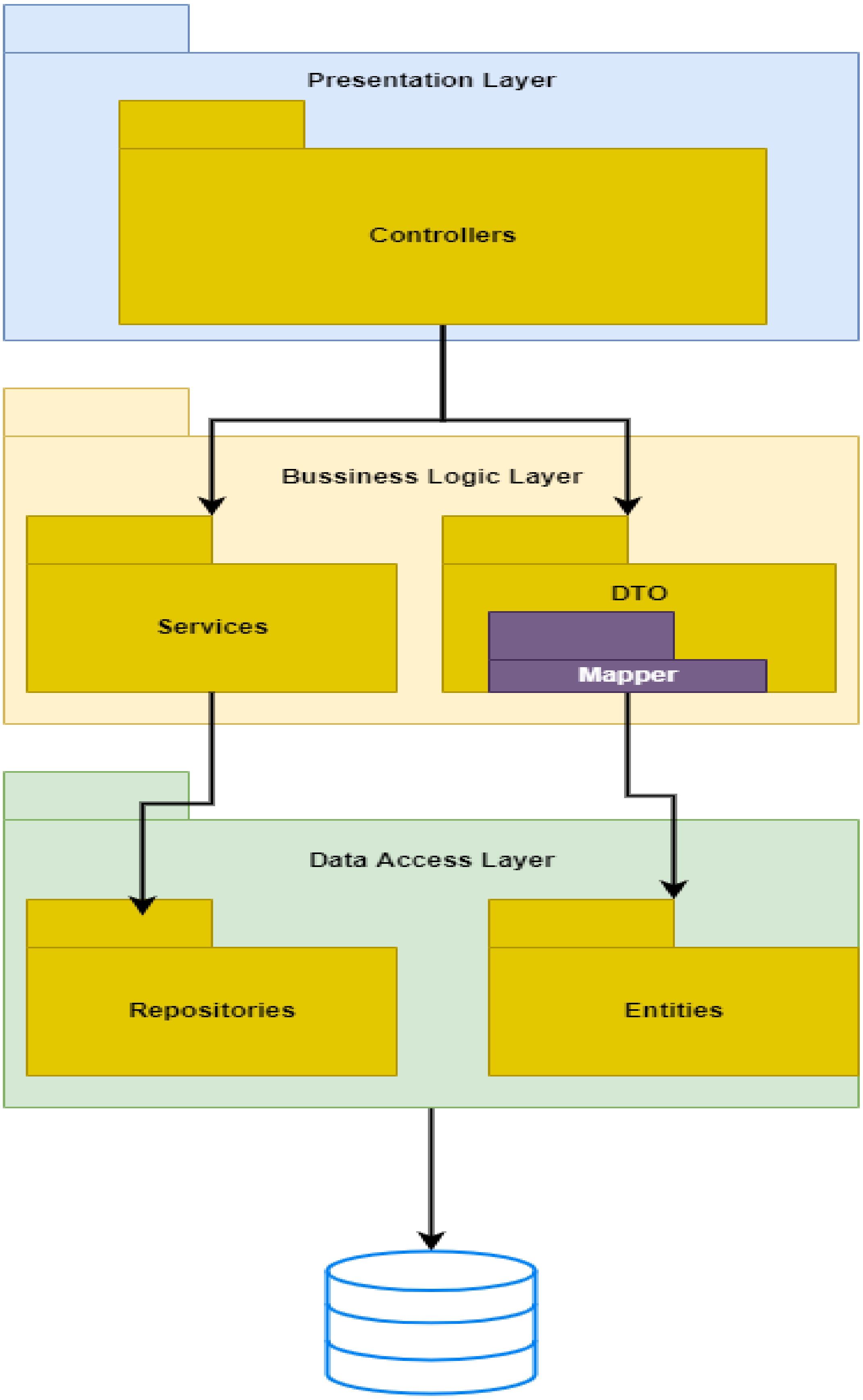


Figure 5.3: Layered architecture

5.1.3.2 Database

For the database we chose a relational database. The information about our two types of users (doctors and patients) is stored permanently in the dabase. For a better optimization of the database we link the user with the required information of each one of them through a foreign key.

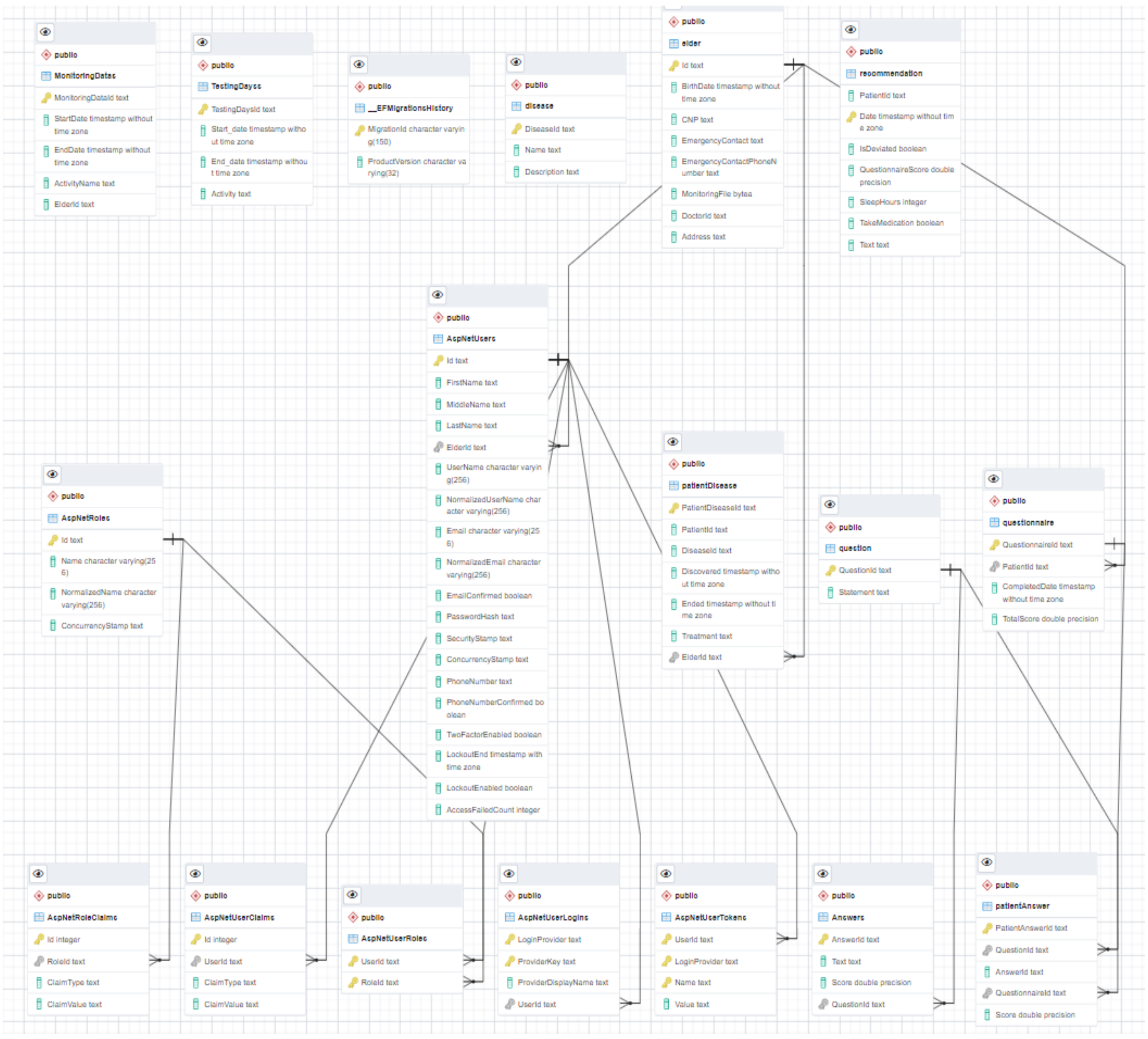


Figure 5.4: Database ERD

For each type of user the role he has defines the privileges and acces to certain pages and requests. This approach boosts the security on both server and client side. The complexity point of view highlights that the user has the most complex table beeing linked with multiple data from the database.

Users have acces to recommendation through the Recommendation generator that uses the data processed from tha anomaly detection component that is stored in the Recommendation table.

The test days table is necessary for saving the data that is coming in real time from the sensors and that are mandatory for routine comparing and recommendation generations.

5.1.3.3 Recommendation Generator

The Recommendation generator is a separate application that comunciates with the main back-end through HTTP requests this application beeing the same REST API as the back-end. Generating the recommendations is possible with the processing part from the anomaly detection component and storing the results in the Database. The information for a specific user is taken from the database and sent over to the generator where the processing of the data takes place.

When the recommendation is generated based on the data that is sent from the back-end, it returns to the back-end application that returns the recommendation in the client side.

5.1.3.4 Producer

The producer is an external application that mimics the real time data generation from the sensors, data that is processed by the back-end for multiple operations as well as anomaly detection and recommendation generation.

The data is sent over from the producer to the back-end through a queue, the back-end consumes all the information that is on the queue after that data is saved in the Database for further operations and processing on it. The frequency of sending the data to the queue from the producer can be modified so we can better mimic the real time data generation.

5.2. Implementation

The most significant implementation choices will be covered in this paragraph. The following are the primary technologies employed in the system's implementation:

- **VueJS.** We made the choice to go with the Vue JavaScript framework because of the simple understanding of the framework and big community that has raised beeing between the most used JavaScript frameworks. The front end implementation is made using Vue.
- **Spring.** The implementation of Java corporate applications often uses the Spring framework. The recommendation generator was implemented using the spring boot web server and dependency injection framework.
- **.NetCore** is a free and open source web framework developed by Microsoft and provides an easy REST API Controller architecture that was used to implement the Back-End application.
- **.NetCore IdentityUser** enables you to include login capabilities in your application through a membership system.

- **CloudAMQP** provides a messaging transport and delivery service for applications that is hosted in cloud.
- **Entity Framework ORM** Entity Framework is an object relational mapper (orm), a tool that makes it easier to map software objects to the tables and columns of a relational database, it also has it's own query language that makes the interaction with the Database easier.
- **PostgreSQL** is an open source relational database that can warehouse all the data needed for our application.
- **VUEX** Acts like a centralized store for all the components in the application.
- **Bootstrap** A powerful front-end framework used to build contemporary websites and online apps.
- **Google Charts** Google charts is a free tool from google that helped us to visualize the data better.

5.2.1. Front-End implementation

The Front-End implementation is achieved through multiple VueJs files that incorporate JavaScript for functionalities, CSS for styling and also HTML for rendering, the last one beeing injected with data from the server side.

Different roles of the user is set when the Login button is pressed through the JWT that is returned from the server side. The JWT carries the role and other usefull informations about the logged in user. All the data about the user is stored with the help of vuex, this way the data is persistent even after the login and can be used for accessing different pages that the role of the user allows him and also make requests.

The number of accesible screen for the doctor is seven as follows:

- Welcoming screen for the doctor
- Patients list, where the doctor has access to all the data of all the users that are using the application.
- Add patient page where the doctor add a patient to the application, along with the personal information required to register in the application a csv file is added with resgistered data from the sensors that are placed in the patient house. This data is used later on by the doctor to detect a routine, deviations and also prepare data for the Recommendation Generator.
- A patient personal file where the doctor can edit the personal informations of the patient, delete the patient and add a disease record for the patient.
- A disease page where the doctor can perform CRUD on diseases.
- A page where the doctor can visualize on a chart the progress or regress of a patient based on the answers given on a questionnaire also here it can visualize the score for each questionnaire and the answers for each questionnaire.
- A page where the doctor can request for a certain patient to compute the routine

based on the data that was inserted at patient creation after that the doctor can request to see if a day choosen by him is deviated. On the same page but on a different tab he can see statistics about the activities duration in each part of the day and at last he can see the results about the requested day that contains the numbers of slept hours, if the medication has been taken by the patient, the last questionnaire score and also if the day is deviated.

The number of accesible screen for the patient is four as follows:

- Welcoming screen for the patient with some usefull informations and where he can see the last recommendation.
- A page where the patient can see his recommendations for multiple days
- Questionnaire page where the patient can complete a questionnaire regarding his mental health and his lifestyle.
- A timeline page where he can see the activities he did and their duration by selecting an interval of dates.

The routing between pages is implemented using a router that also checks the privileges of a user before entering a certain page eliminating the case where the user can see a doctor page or vice versa. This thing also boosts security of the application in the client side.

Along with the Bootstrap that provides components for a more friendly UI, a notification library is used for notifying the users with different warnings or information. At this we added the Google chart library so we can visualize data better and easily present it to the doctor.

```
<GChart
    type="ColumnChart"
    :data="chartData"
    :options="chartOptions"
/>
```

The component that we use is the GChart, the type of the chart is given to the type attribute, the data we want to represent on the chart is passed through the data attribute and at last we have the options of the chart.

- **Type**
  - Pie
  - Area
  - Line
  - Column
  - Tree map

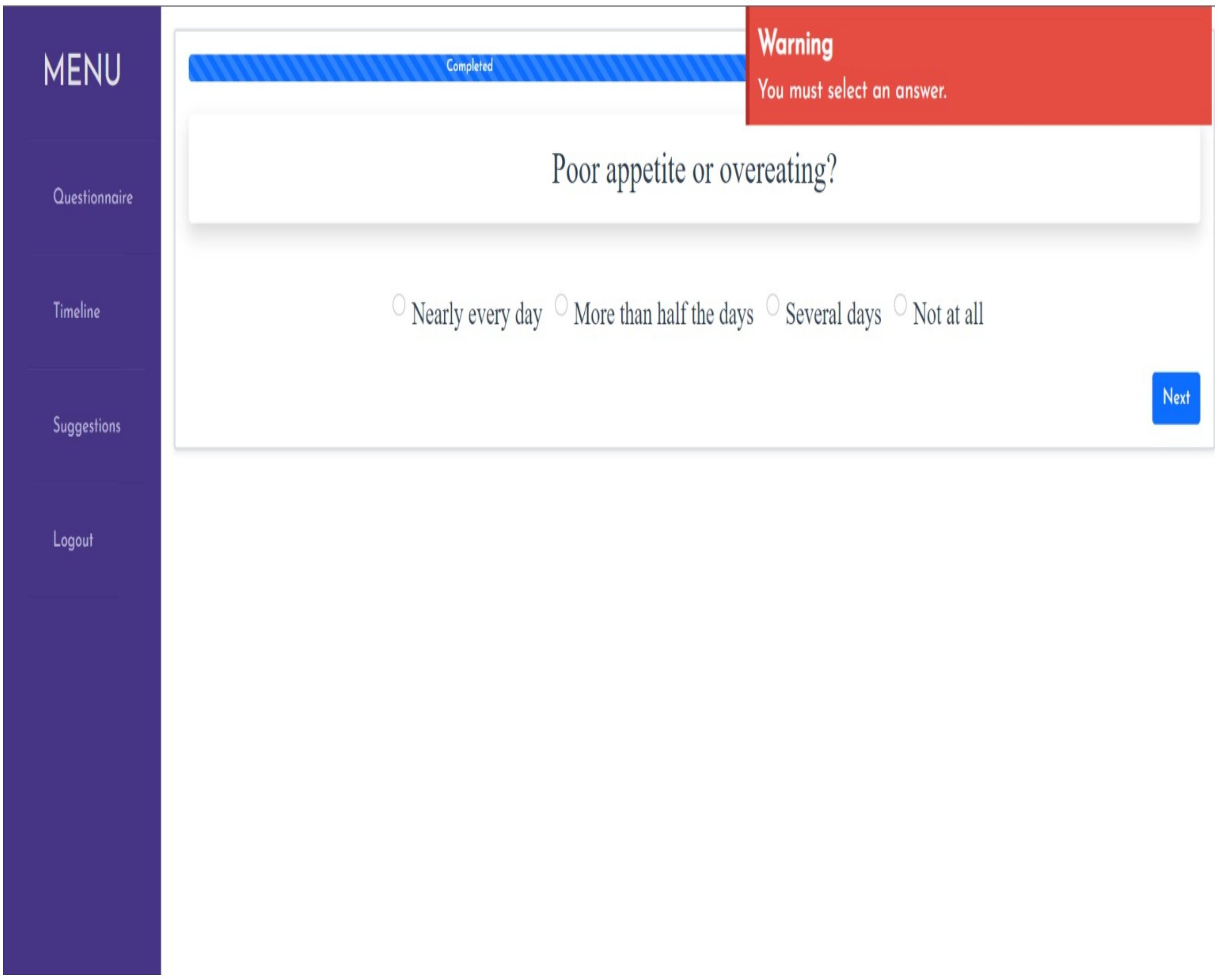
Chart options attribute can be modified at will so the chart can have different specifications like:

- Height
- Width

- Title
- SubTitle

```
chartOptions2: {
        width:1300,
        chart: {
          title: "Activities",
          subtitle: "",
          legend: { position: "top", maxLines: 3 },
          isStacked: true,
        },
      },
```

The chart that allows the doctor watch the performance of the patient regarding it's mental health through questionnaires that are scored along with the timeline that the patient can see and the questionnaire page where the user takes questionnaires look like this:



Warning
You must select an answer.

Poor appetite or overeating?

Nearly every day More than half the days Several days Not at all

Figure 5.5: Questionnaire with invalid next button press

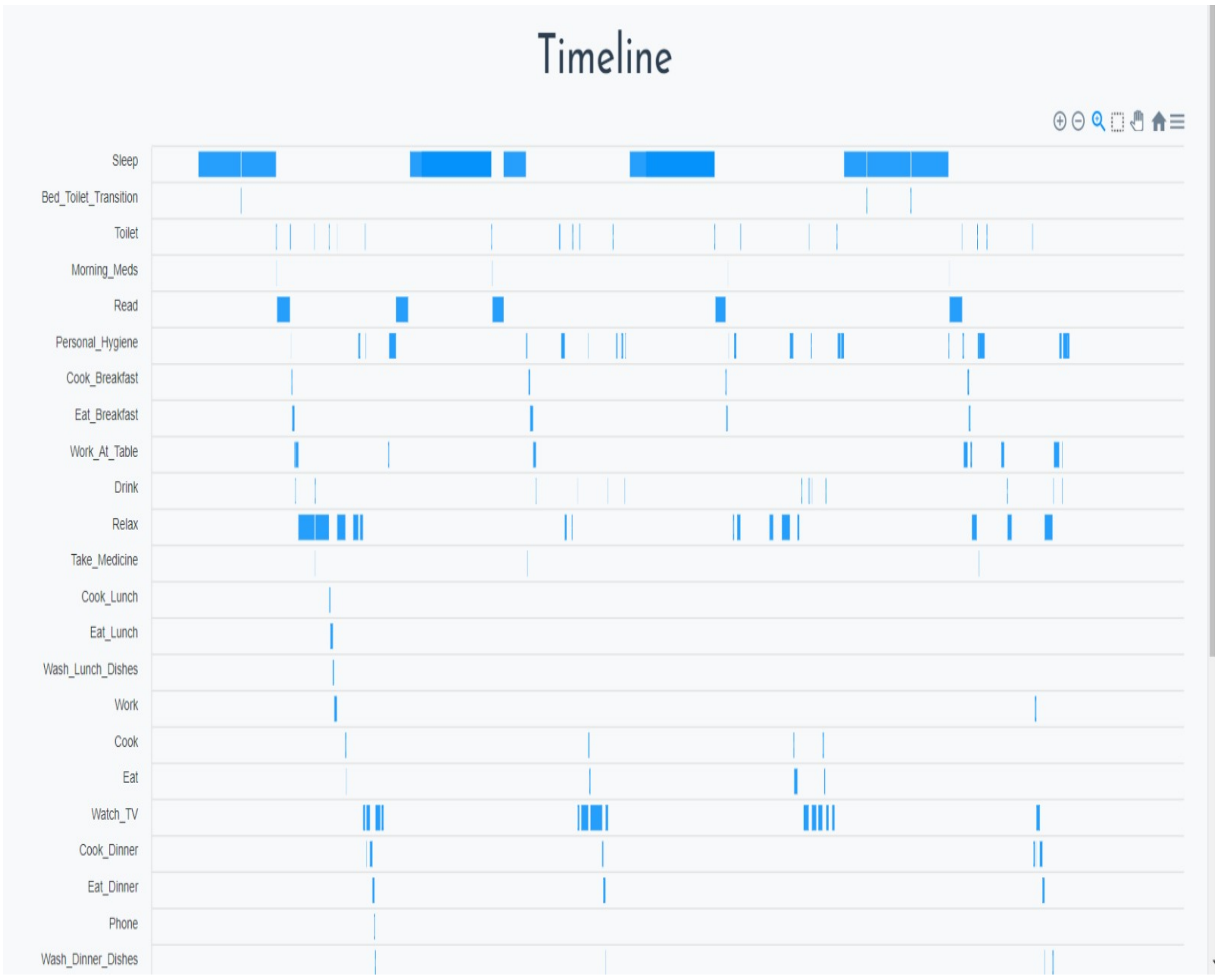
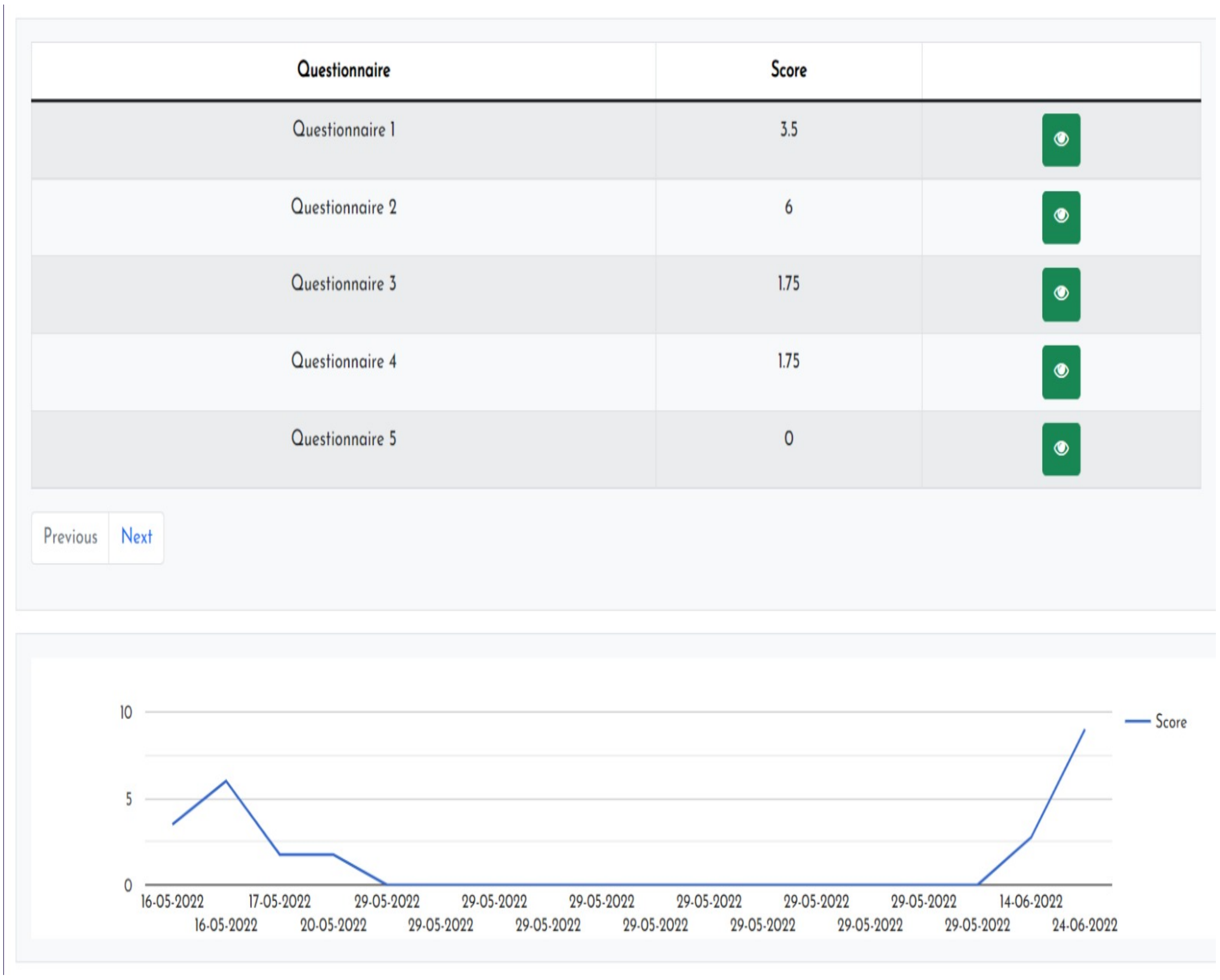


Figure 5.6: Tiemline view



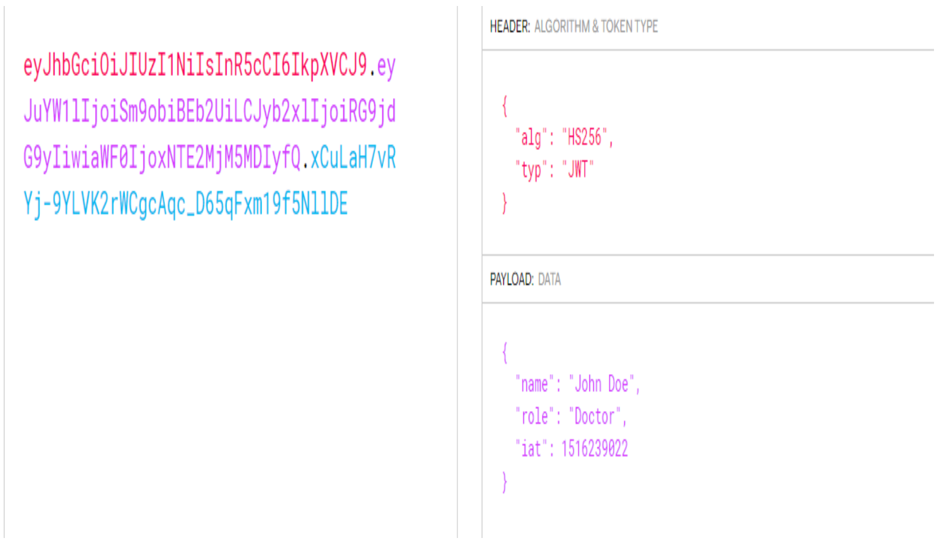
Questionnaire	Score	
Questionnaire 1	3.5	👁
Questionnaire 2	6	👁
Questionnaire 3	1.75	👁
Questionnaire 4	1.75	👁
Questionnaire 5	0	👁

Figure 5.7: Questionnaire score chart

5.2.2. Back-End implementation

As a consequence of the complexity that the back-end application has we will highlight the most vital parts.

Let's start with the security of the application, where the most important part is the JWTAuthenticationBearer classes. This enables us to use a JWT (JSON Web Token), for communication between client side and server side. The request from the client side has a header where the JWT is placed. Server side receives the request and parses the JWT that has a payload where the information about the user that made the request. Depending on the information and validity of the token the server side decides if it will send response or will execute the given request.



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JuYW1lIjoiSm9obiBEb2UiLCJyb2xlIjoiRG9jd
G9yIiwiaWF0IjoxNTE2MjM5MDIyfQ.xCuLaH7vR
Yj-9YLVK2rWCgcAqc_D65qFxm19f5NlIDE
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

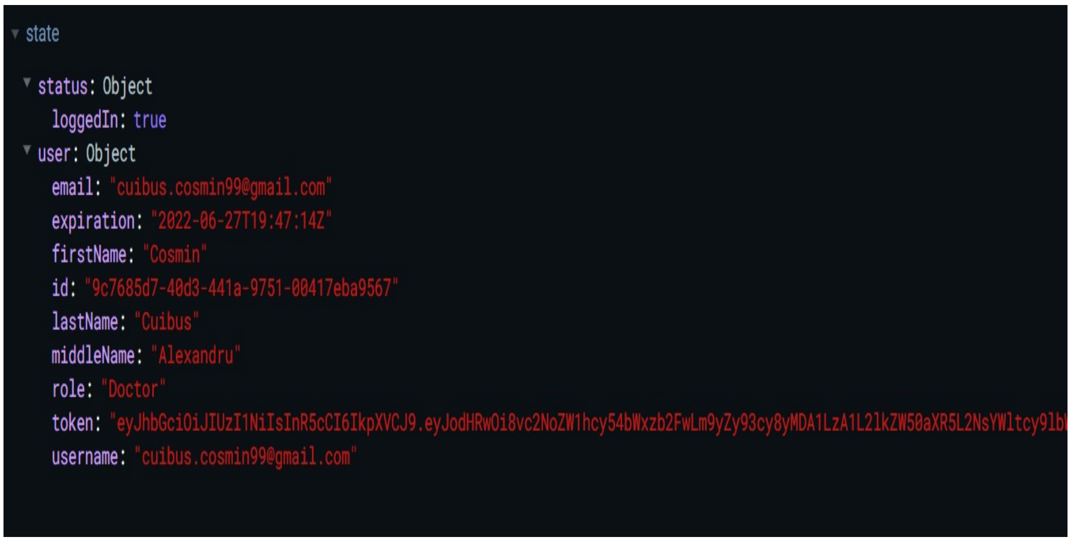
PAYLOAD: DATA

```
{
  "name": "John Doe",
  "role": "Doctor",
  "iat": 1516239022
}
```

Figure 5.8: Structure of a JWT token [20]

For the creation of the JWT a user must login with a valid email and password and as a return to the login request, the server side sends a newly generated JWT token containing in the payload multiple data regarding the user like: Name, id, generation timestamp and expiring timestamp. Also the server side checks if the JWT is send from a valid audience to prevent even more unauthorised requests in case of JWT leak. This boosts even more the security of the application and secures the communication channel between the client side and the server side.

One of the most important parts of the back-end application is build up by the REST Controllers. Without the REST Controllers the application can't communicate with other components that build our system. The Controllers are adnotated with [ApiController] and the path is set using [Route("api/[controller]")] adnotation



```
▾ state
  ▾ status: Object
      loggedIn: true
  ▾ user: Object
      email: "cuibus.cosmin99@gmail.com"
      expiration: "2022-06-27T19:47:14Z"
      firstName: "Cosmin"
      id: "9c7685d7-40d3-441a-9751-00417eba9567"
      lastName: "Cuibus"
      middleName: "Alexandru"
      role: "Doctor"
      token: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lkZW50aXR5L2NsYWltcy9lb
      username: "cuibus.cosmin99@gmail.com"
```

Figure 5.9: Structure of the object created and stored in the VUEX from the back-end after login.

```
[Route("api/[controller]")]
    [ApiController]

    // POST api/<DiseaseController>
        [HttpPost]
        public void Post([FromBody] DiseaseDto disease)
        {
            disease.DiseaseId = Guid.NewGuid().ToString();
            _diseaseRepository.Create(disease);
        }
```

Each method from a controller is adnotated with the type of HTTP request it maps. Also the variables that are coming from body (POST) or from URL (GET) are given as input parameters in the header of the function. The controller calls a service that at their turn will call a repository. The Controllers receives variables that are mapped to DTO's so the transfer from Controller to Service and Repository is made by the DTO. Also when there are returning variables/objects that need to be sent as a response to a request, the entity is mapped to a DTO.

Another vital part of the back-end application is the communication with the Recommendation generator. We use a HTTP request between the back-end application and the Recommendation Generator component. The data is sent over a POST request that has in the body the necessary data for generation a recommendation based on the received data Fig 5.10.

Another important part of the Back-End application is the Producer component and also the consumer one. The producer component is an application that mimics a real time data generator and uses CloudAMQP, a RabbitMQ service hosted in cloud. This

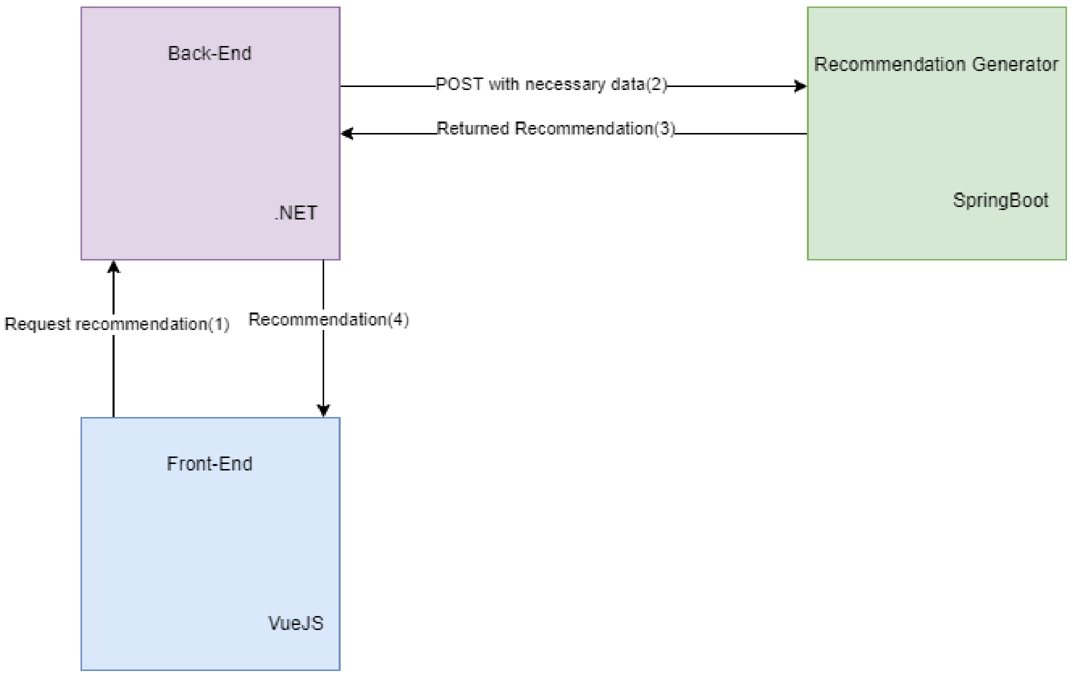
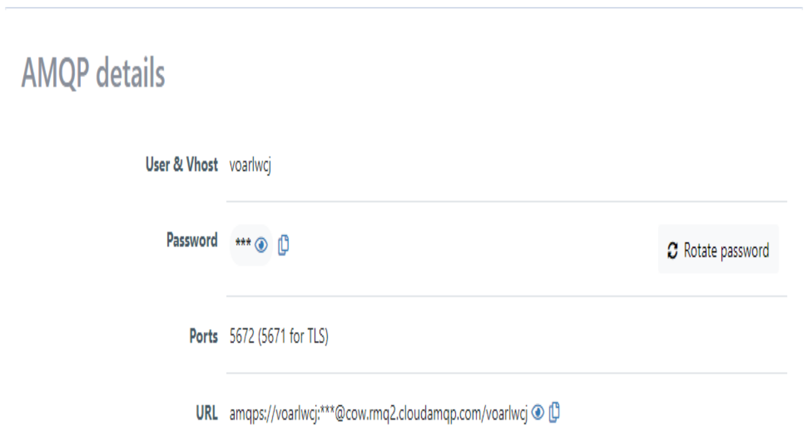

Back-End
.NET
POST with necessary data(2)
Returned Recommendation(3)
Recommendation Generator
SpringBoot
Request recommendation(1)
Recommendation(4)
Front-End
VueJS

Figure 5.10: Simple understanding of the communication between Front-End, Back-end and Recommendation Generator

producer reads data from a csv and at a set interval of time sends data to the queue. The configurations necesary to use the cloud hosted queue are given by the site so we can simply set the password and the URL of the queue Fig 5.11.



AMQP details

User & Vhost	voarlwcj
Password	***
Ports	5672 (5671 for TLS)
URL	amqps://voarlwcj:***@cow.rmq2.cloudamqp.com/voarlwcj

Figure 5.11: CloudAMQP config to connect

The last but not the least important part of our back-end is the database. The database communicates with the main back-end application through EntityFramework ORM. This Object Relational Mapper integrates perfectly with the .NET application, due to Microsoft NuGet Package Manager. This ORM comes as a simple additional package that is installed in tha application. The Mapping between our classes and data tables is very easy and very fast, the ORM taking the responsability to make migrations to the database and also update it. Handling the data and interogating the database is very easy due to the simple non query language that the ORM is using. A query with this ORM lokks something like this.

```
public List<ElderByDoctorDto> AllEldersOfADoctor(string id)
{
    var elders = dbContext.Elders.
    Join(dbContext.Users, elder =>
    elder.Id, user => user.Elder.Id,
    (elder, user) => new ElderByDoctorDto
    {Id = elder.Id,FirstName = user.FirstName,
    MiddleName = user.MiddleName, LastName = user.LastName,
    CNP = elder.CNP, DoctorId = elder.DoctorId })
    .Where(e => e.DoctorId == id).ToList();

    return elders;
}
```

The connection between the main back-end application is made through a connection string and a NuGet package that must be installed for communication between .NET and PostgreSQL. The connection string shoul include the user of the database, the password, the host where the database is located, the port number on which the database is running and the name of the database.

5.2.3. Recommendation Generator

For this component of the application we used SpringBoot, together with Weka that is an open source collection of machine learning algorithms that also contains tools for preprocessing the data, classification and visualization.

The SpringBoot framework helped us to create a REST API type of application so we can make a request from the main application through an POST HTTP request. The request carries the data necessary for generating the recommendation.

The data necessary for generatin recommendations are taken from the database by the main back-end application for the patient that the request is made. Scoring of the questionnaires that help to keep track with the depression mental and phisical health, the number of slept hours that are taken for each day individually from the dataset that is registered alongside the patient and also from the dataset if the patient has taken his/her medication.

Each patient can take a questionnaire when they log in, each answer choosen by the patient has a score, at the end when the patient submits the questionnaire a copy of the completed questionnaire is available on the doctor side. The overall score of the questionnaire is computed and also shown on a chart in the doctor view. For recommendation generation the last completed questionnaire of the patien is taken in consideration, partial answers not beeing saved and so not beeing eligible as a completed questionnaire.

This information is received by the Controller that creates a RecommendationDto object from them. Every values is than mapped to a Recommendation class, where they are mapped to be useful atributes in our .arff file. Every attribute is feed to the weka decision tree and after the final nodes of the trees have been created a naive bayes classifier is applied on the results.

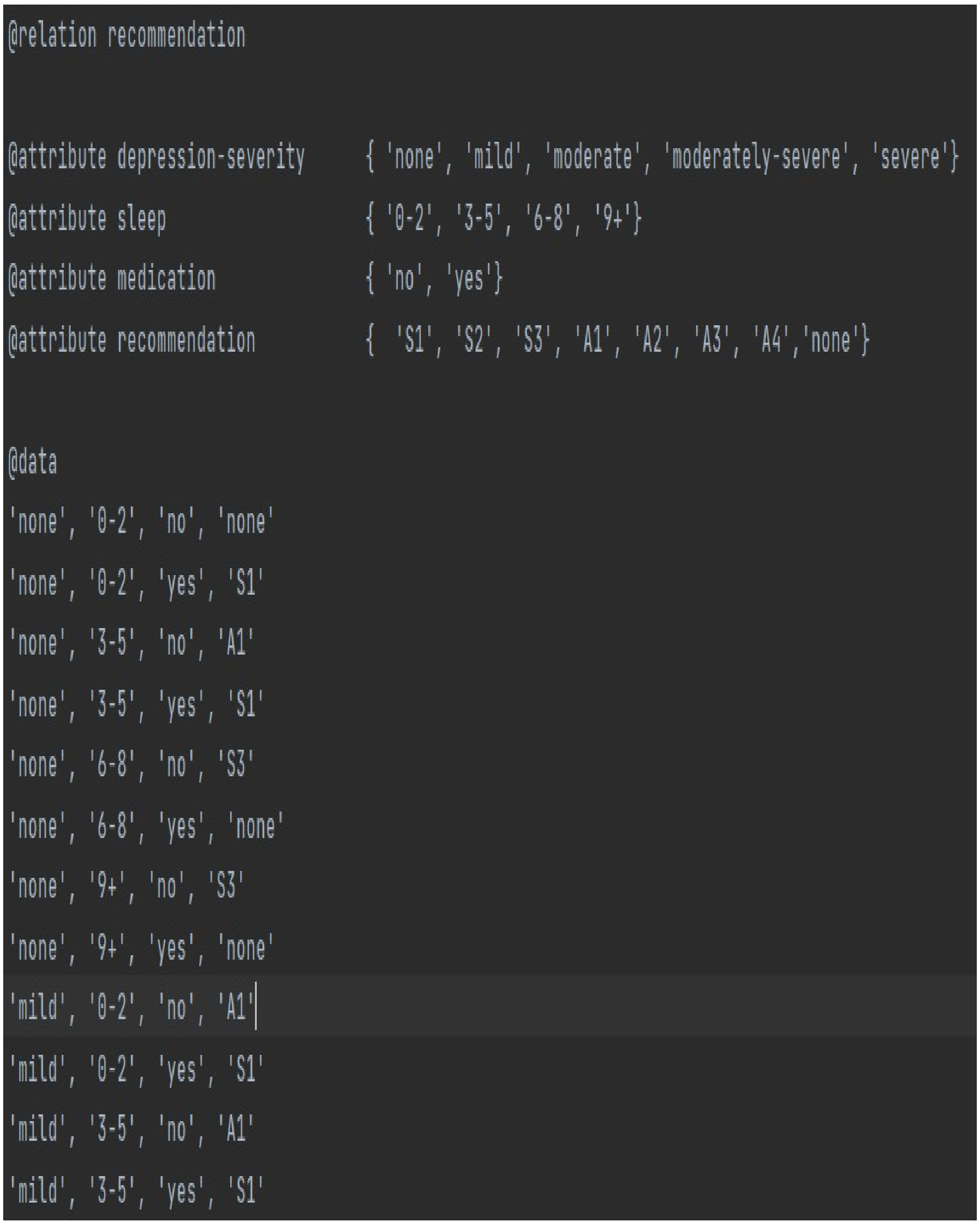
```
ConverterUtils.DataSource source =
new ConverterUtils.DataSource("ADL.arff");

Instances train = source.getDataSet();

train.setClassIndex(train.numAttributes() - 1);

NaiveBayes tree = new NaiveBayes();

tree.buildClassifier(train);
```

The data is readed from the .ARFF file with the Instance() class and then with the help of the setClassIndex(int) we set the class from where the prediction should be part of. The last index from the .ARFF file is the recommendation we want to get.

Each label that is predicted is taking in consideration the data send over from the back-end application and their meaning is as follows:

1. **S1** We strongly recommend you to get more sleep.
2. **S2** You should take in consideration talking to someone about your state.
3. **S3** We recommend to set an alarm, so you don't forget to take your medication.
4. **A1** We strongly recommend you to get more sleep and set an alarm so you don't forget about medication.
5. **A2** We strongly recommend you to get more sleep and you should take in consideration talking to someone about your state.
6. **A3** We strongly recommend you to get more sleep, you should take in consideration talking to someone about your state and don't forget to set an alarm for medication.
7. **A4** You should take in consideration talking to someone about your state and don't forget to set an alarm for medication.
8. **none** None

The structure of the ARFF file is as follows. All the atributes have a relation of recommendation, all atributes have multiple labels that can be represented by and all the data adnotation is all the combination between the atributes. The combinations are made using each attribute's labels and at the end of combination there is always a label for the recommendation that should be given for that combination.

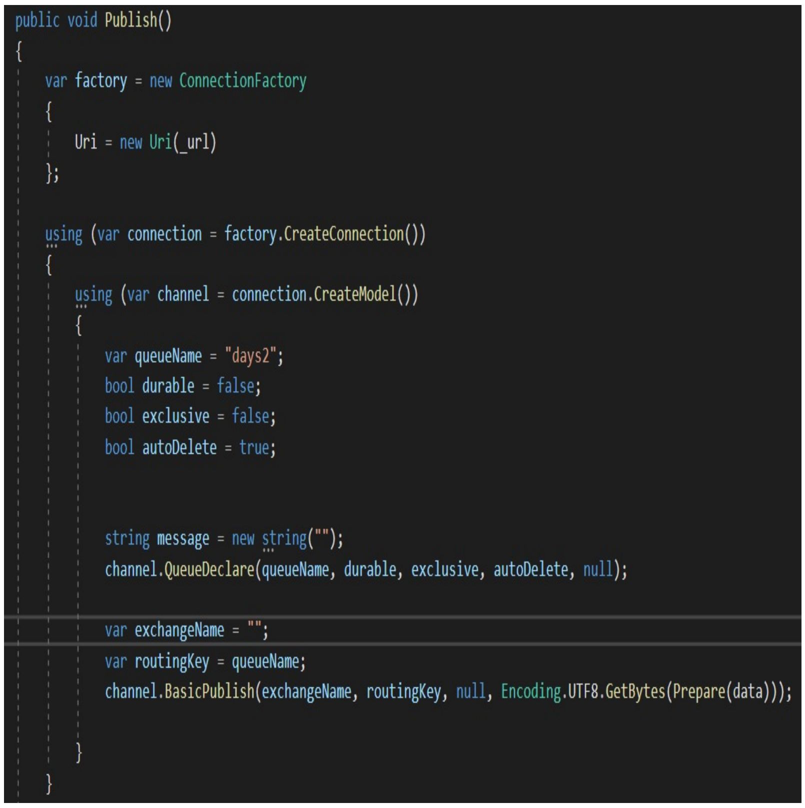


```
@relation recommendation


@attribute depression-severity     { 'none', 'mild', 'moderate', 'moderately-severe', 'severe'}
@attribute sleep                   { '0-2', '3-5', '6-8', '9+'}
@attribute medication              { 'no', 'yes'}
@attribute recommendation          {  'S1', 'S2', 'S3', 'A1', 'A2', 'A3', 'A4','none'}


@data
'none', '0-2', 'no', 'none'
'none', '0-2', 'yes', 'S1'
'none', '3-5', 'no', 'A1'
'none', '3-5', 'yes', 'S1'
'none', '6-8', 'no', 'S3'
'none', '6-8', 'yes', 'none'
'none', '9+', 'no', 'S3'
'none', '9+', 'yes', 'none'
'mild', '0-2', 'no', 'A1'
'mild', '0-2', 'yes', 'S1'
'mild', '3-5', 'no', 'A1'
'mild', '3-5', 'yes', 'S1'
```

Figure 5.12: Attributes in the arff file

5.2.4. Producer (real time data generator)

The producer is a .NET Desktop Application that must be turned on, and also set the timer for sending data. It simulates an entire set of data at an interval that we can set. The communication between the producer and the cloud is realised with an URL, password and queue name. Data is read from a csv file and processed so an entire list of activities with their start and end date are sent, an entire day. The Publish() method is responsible to create the connection with cloud queue, and send the data.



```
public void Publish()
{
    var factory = new ConnectionFactory
    {
        Uri = new Uri(_url)
    };

    using (var connection = factory.CreateConnection())
    {
        using (var channel = connection.CreateModel())
        {
            var queueName = "days2";
            bool durable = false;
            bool exclusive = false;
            bool autoDelete = true;


            string message = new string("");
            channel.QueueDeclare(queueName, durable, exclusive, autoDelete, null);

            var exchangeName = "";
            var routingKey = queueName;
            channel.BasicPublish(exchangeName, routingKey, null, Encoding.UTF8.GetBytes(Prepare(data)));

        }
    }
```

Figure 5.13: Publish method

The consumer is located in the main back-end application and it is using the same connection parameters to make the link with the cloud queue. After consuming the data in the queue it creates a list of Day objects, that contains the same attributes, activity name, start and end date.

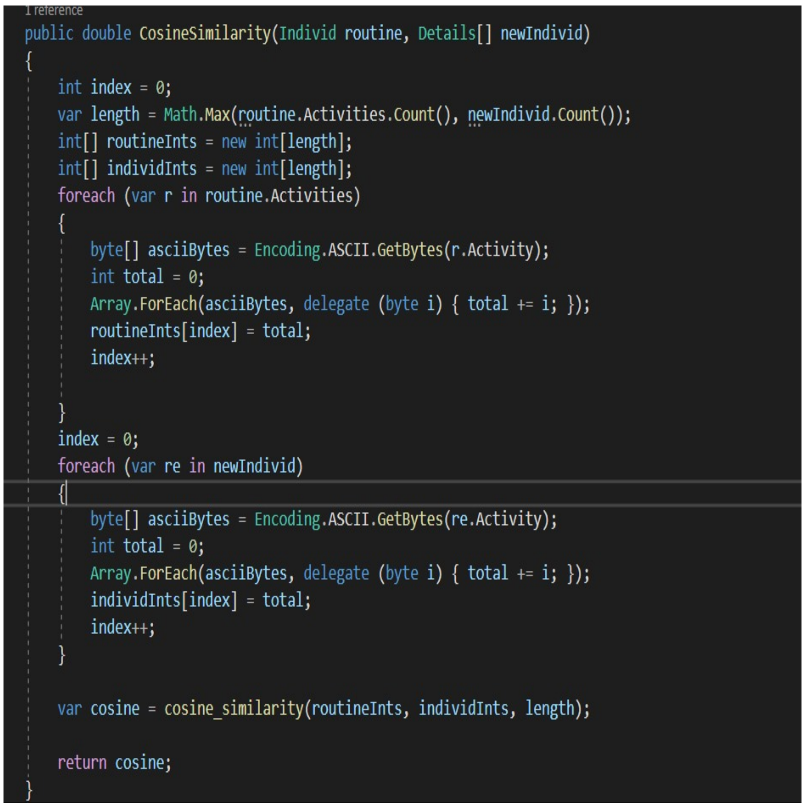
After building the list, the entire data is sent to the database, where is stored for later use. The days that are saved at this step are used for testing the anomaly detection component and also for recommendation generator.

5.2.5. Deviation detection component

The deviation detection component is included in the main back-end application. This component is used along the routine that is found on an interval of days from the dataset. The deviation detecction has two main parts.

The first part of this compoennt is to see if the day is deviated from the activities order point of view. For a better understanding imagine we have two vectors with activities, the routine that was detected and the day we want to know if is deviated. For this we will calculate the cosine similarity of the vectors. The higher the cosine similarity the higher the similarity between activities order. After thinking how to compare strings and implement the cosine similarity formula in two arrays of strings, we decided to convert each entry from our two arrays in their ASCII value, that way we can work with numbers not strings.

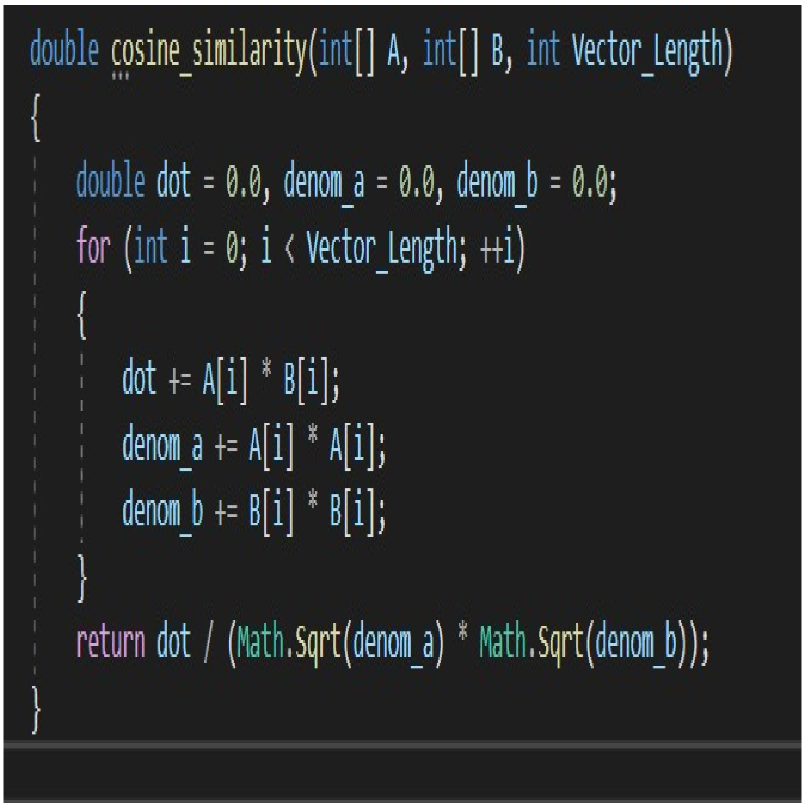
As an example take the activity ”sleep” that is one entry in our array, either in routine or in day, and we convert the activity name into ASCII code ”sleep” => 115 108 101 101 112. After we converted the string and we have the values for each individual letter we will sum the values so we can have the value of the entire word. The value of ”sleep” is 537. We will use this value for the cosine similarity formula.



```
1 reference
public double CosineSimilarity(Individ routine, Details[] newIndivid)
{
    int index = 0;
    var length = Math.Max(routine.Activities.Count(), newIndivid.Count());
    int[] routineInts = new int[length];
    int[] individInts = new int[length];
    foreach (var r in routine.Activities)
    {
        byte[] asciiBytes = Encoding.ASCII.GetBytes(r.Activity);
        int total = 0;
        Array.ForEach(asciiBytes, delegate (byte i) { total += i; });
        routineInts[index] = total;
        index++;

    }
    index = 0;
    foreach (var re in newIndivid)
    {
        byte[] asciiBytes = Encoding.ASCII.GetBytes(re.Activity);
        int total = 0;
        Array.ForEach(asciiBytes, delegate (byte i) { total += i; });
        individInts[index] = total;
        index++;
    }

    var cosine = cosine_similarity(routineInts, individInts, length);

    return cosine;
}
```

Figure 5.14: Main Cosine Function

We consider a day deviated if the cosine similarity value that is returned is under a certain threshold, this threshold can be modified to be more permissive with the results or



```
double cosine_similarity(int[] A, int[] B, int Vector_Length)
{
    double dot = 0.0, denom_a = 0.0, denom_b = 0.0;
    for (int i = 0; i < Vector_Length; ++i)
    {
        dot += A[i] * B[i];
        denom_a += A[i] * A[i];
        denom_b += B[i] * B[i];
    }
    return dot / (Math.Sqrt(denom_a) * Math.Sqrt(denom_b));
}
```

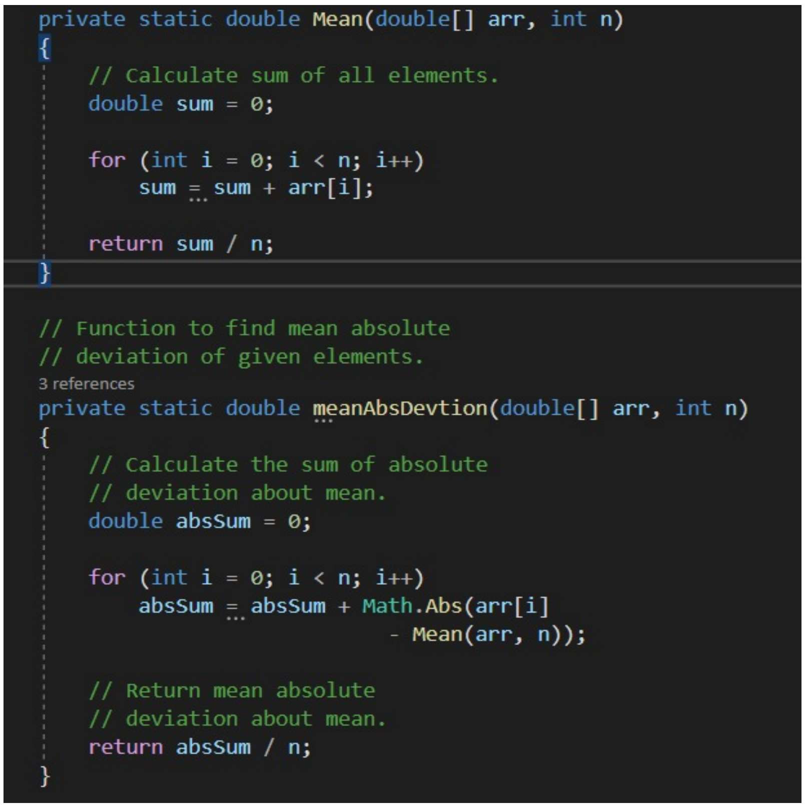
Figure 5.15: Cosine Formula

more strict. Only if the day is not deviated from the order point of view we can proceed to the second part of this component.

The second part of this component checks if the duration of activities from the selected day that we want to see if is deviated are longer or shorter than the durations from the found routine. Depending on the type of activity we tolerate a deviation bigger or smaller.

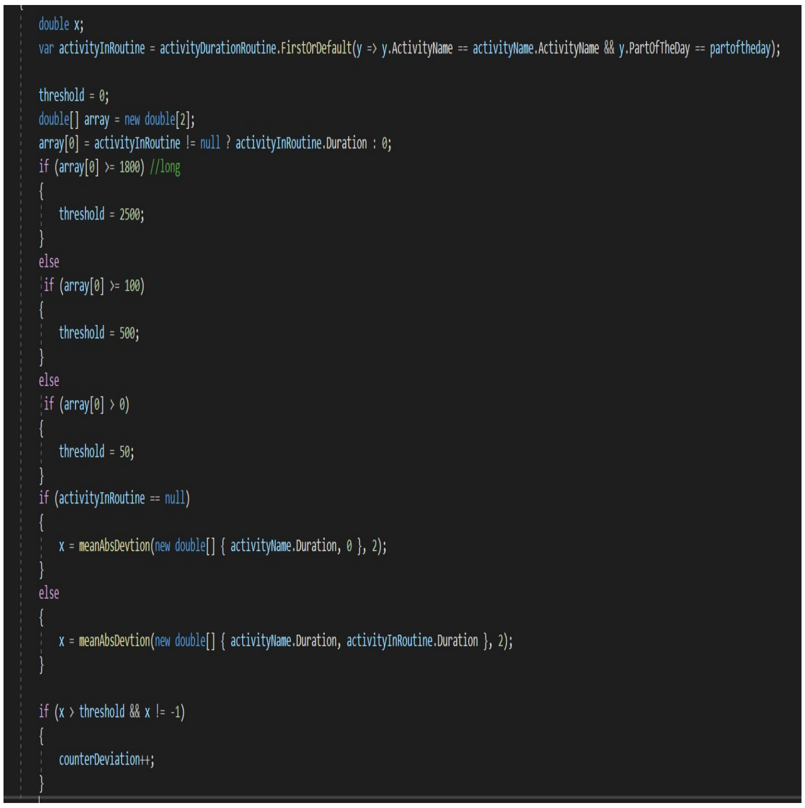
The types of activities can be long, short or medium. Considering that we copmpute the mean of the routine and we will compare that to the duration of the activities from the chosen day taking in consideration the type of activity. If the duration of the activity exceeds the threshold we count hat activity as deviated. If more than percent that we set as threshold from the activities of the day are deviated than we consider that day to be deviated.

If the activity is not considered deviated after passing the second part of the deviation detection component than the day is not deviated.



```
private static double Mean(double[] arr, int n)
{
    // Calculate sum of all elements.
    double sum = 0;

    for (int i = 0; i < n; i++)
        sum = sum + arr[i];

    return sum / n;
}

// Function to find mean absolute
// deviation of given elements.
3 references
private static double meanAbsDevtion(double[] arr, int n)
{
    // Calculate the sum of absolute
    // deviation about mean.
    double absSum = 0;

    for (int i = 0; i < n; i++)
        absSum = absSum + Math.Abs(arr[i]
                                - Mean(arr, n));

    // Return mean absolute
    // deviation about mean.
    return absSum / n;
}
```

Figure 5.16: The mean and mean deviation functions



```
double x;
var activityInRoutine = activityDurationRoutine.FirstOrDefault(y => y.ActivityName == activityName.ActivityName && y.PartOfTheDay == partoftheday);

threshold = 0;
double[] array = new double[2];
array[0] = activityInRoutine != null ? activityInRoutine.Duration : 0;
if (array[0] >= 1800) //long
{
    threshold = 2500;
}
else
 if (array[0] >= 100)
{
    threshold = 500;
}
else
 if (array[0] > 0)
{
    threshold = 50;
}
if (activityInRoutine == null)
{
    x = meanAbsDevtion(new double[] { activityName.Duration, 0 }, 2);
}
else
{
    x = meanAbsDevtion(new double[] { activityName.Duration, activityInRoutine.Duration }, 2);
}

if (x > threshold && x != -1)
{
    counterDeviation++;
}
```

Figure 5.17: The main function for comparing the duration for the days

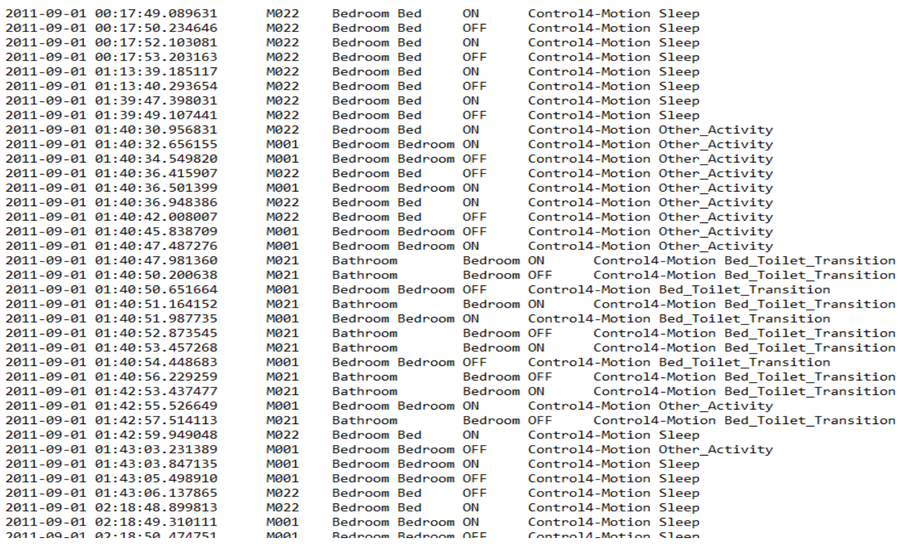
Chapter 6.   Testing and Validation

In the following section we will evaluate the results of our approach regarding anomaly detection and recommendation generation.

6.1.   Dataset

Tha dataset we are using is from CASAS (Center of Advanced Studies In Adaptive Systems), the dataset we have chosen has as a main actor a single elder that lives alone in his/her apartment. Data is collected continuously while residents perform their normal routines.

Ambient PIR motion sensors, door/temperature sensors, and light switch sensors are placed throughout the home of the volunteer. The sensors are placed in locations throughout the home that are related to specific activites of daily living that we wish to capture. The classification task is to predict the activity that is occurring in the smart home and being observed by the ambient sensors. The data set is received in text file and it look like this:



```
2011-09-01 00:17:49.089631      M022    Bedroom Bed     ON      Control4-Motion Sleep
2011-09-01 00:17:50.234646      M022    Bedroom Bed     OFF     Control4-Motion Sleep
2011-09-01 00:17:52.103081      M022    Bedroom Bed     ON      Control4-Motion Sleep
2011-09-01 00:17:53.203163      M022    Bedroom Bed     OFF     Control4-Motion Sleep
2011-09-01 01:13:39.185117      M022    Bedroom Bed     ON      Control4-Motion Sleep
2011-09-01 01:13:40.293654      M022    Bedroom Bed     OFF     Control4-Motion Sleep
2011-09-01 01:39:47.398031      M022    Bedroom Bed     ON      Control4-Motion Sleep
2011-09-01 01:39:49.107441      M022    Bedroom Bed     OFF     Control4-Motion Sleep
2011-09-01 01:40:30.956831      M022    Bedroom Bed     ON      Control4-Motion Other_Activity
2011-09-01 01:40:32.656155      M001    Bedroom Bedroom ON      Control4-Motion Other_Activity
2011-09-01 01:40:34.549820      M001    Bedroom Bedroom OFF     Control4-Motion Other_Activity
2011-09-01 01:40:36.415907      M022    Bedroom Bed     OFF     Control4-Motion Other_Activity
2011-09-01 01:40:36.501399      M001    Bedroom Bedroom ON      Control4-Motion Other_Activity
2011-09-01 01:40:36.948386      M022    Bedroom Bed     ON      Control4-Motion Other_Activity
2011-09-01 01:40:42.008007      M022    Bedroom Bed     OFF     Control4-Motion Other_Activity
2011-09-01 01:40:45.838709      M001    Bedroom Bedroom OFF     Control4-Motion Other_Activity
2011-09-01 01:40:47.487276      M001    Bedroom Bedroom ON      Control4-Motion Other_Activity
2011-09-01 01:40:47.981360      M021    Bathroom        Bedroom ON      Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:50.200638      M021    Bathroom        Bedroom OFF     Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:50.651664      M001    Bedroom Bedroom OFF     Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:51.164152      M021    Bathroom        Bedroom ON      Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:51.987735      M001    Bedroom Bedroom ON      Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:52.873545      M021    Bathroom        Bedroom OFF     Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:53.457268      M021    Bathroom        Bedroom ON      Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:54.448683      M001    Bedroom Bedroom OFF     Control4-Motion Bed_Toilet_Transition
2011-09-01 01:40:56.229259      M021    Bathroom        Bedroom OFF     Control4-Motion Bed_Toilet_Transition
2011-09-01 01:42:53.437477      M021    Bathroom        Bedroom ON      Control4-Motion Bed_Toilet_Transition
2011-09-01 01:42:55.526649      M001    Bedroom Bedroom ON      Control4-Motion Other_Activity
2011-09-01 01:42:57.514113      M021    Bathroom        Bedroom OFF     Control4-Motion Bed_Toilet_Transition
2011-09-01 01:42:59.949048      M022    Bedroom Bed     ON      Control4-Motion Sleep
2011-09-01 01:43:03.231389      M001    Bedroom Bedroom OFF     Control4-Motion Other_Activity
2011-09-01 01:43:03.847135      M001    Bedroom Bedroom ON      Control4-Motion Sleep
2011-09-01 01:43:05.498910      M001    Bedroom Bedroom OFF     Control4-Motion Sleep
2011-09-01 01:43:06.137865      M022    Bedroom Bed     OFF     Control4-Motion Sleep
2011-09-01 02:18:48.899813      M022    Bedroom Bed     ON      Control4-Motion Sleep
2011-09-01 02:18:49.310111      M001    Bedroom Bedroom ON      Control4-Motion Sleep
2011-09-01 02:18:50.474751      M001    Bedroom Bedroom OFF     Control4-Motion Sleep
```

Figure 6.1: Casas Dataset [21]

6.2. Deviation Detection and Recommendations

The doctor logs in and selects a patient, he goes to the routine generation page, because without the routine we could not detect deviations, the routine beeing the reference point for a day. After the doctor generates the routine by pressing the ”Load Data” buttom he has the possibility to see the routine and select a date to request an anomaly detection on the selected day.

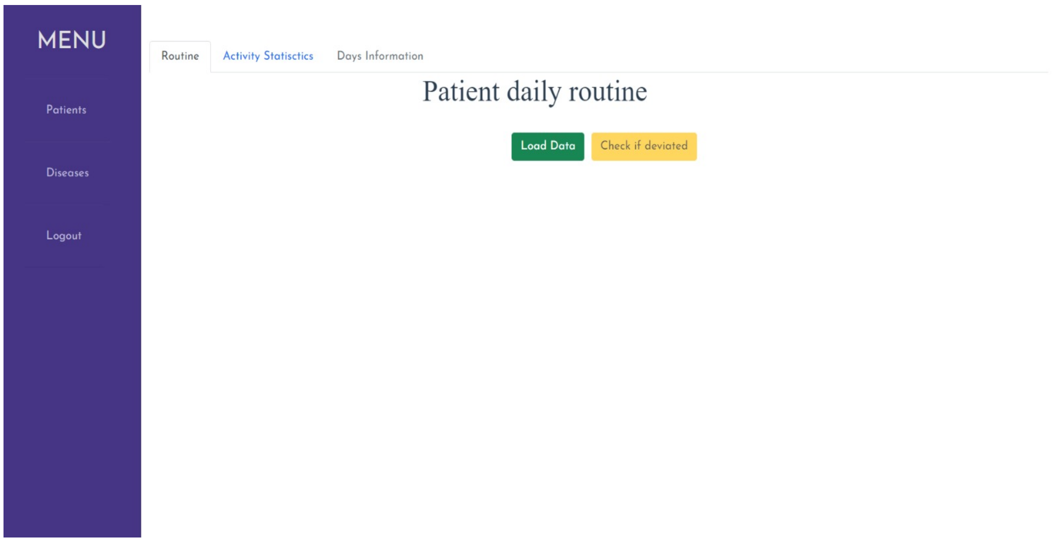


Figure 6.2: Routine Page

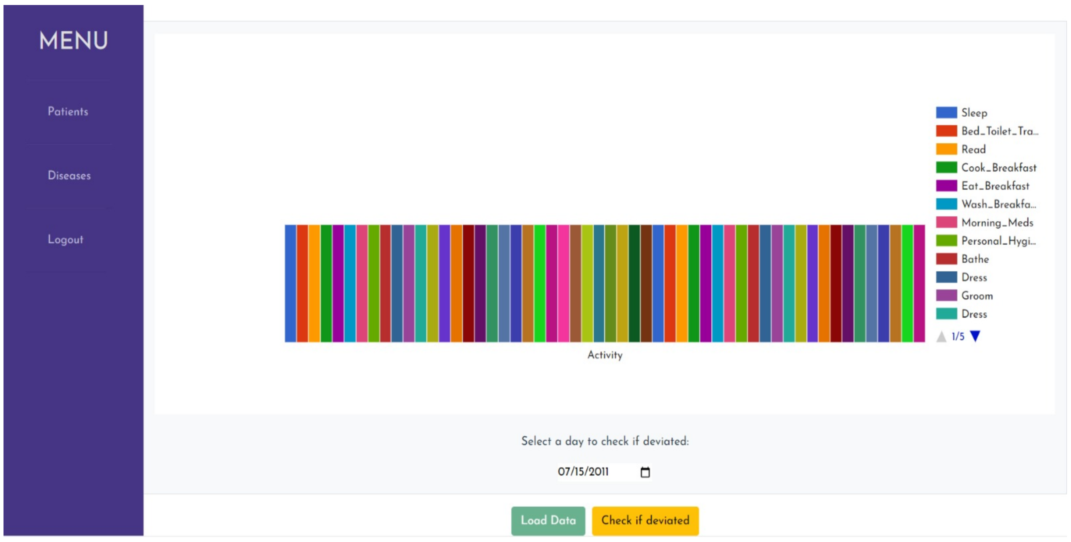


Figure 6.3: Routine Generated

After the a date is selected the doctor presses the "Check if deviated" button, that will make a request to the deviation detection component and will return an alert with an information message that will announce the doctor that the Days Information Tab from the page is now available.

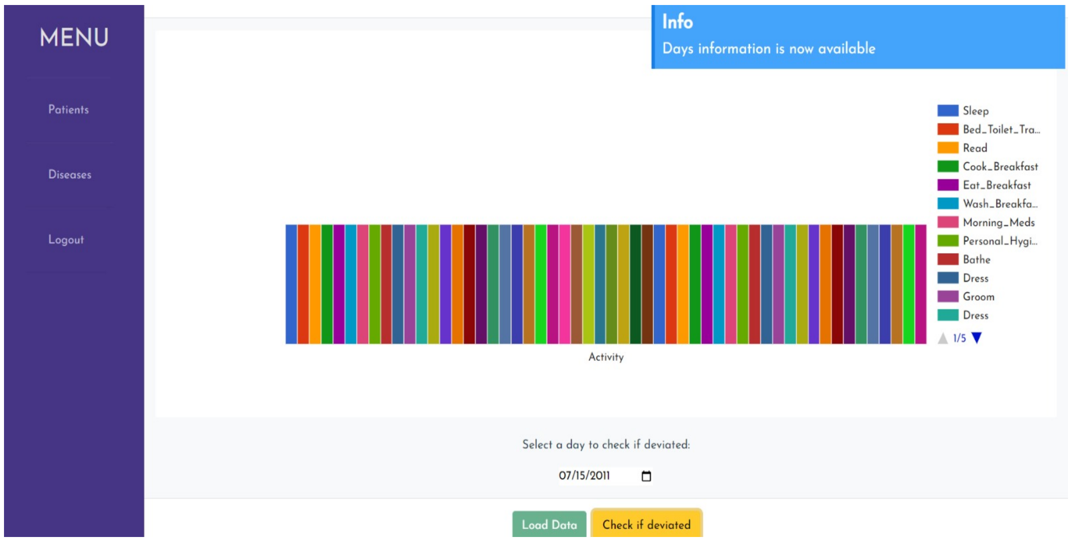
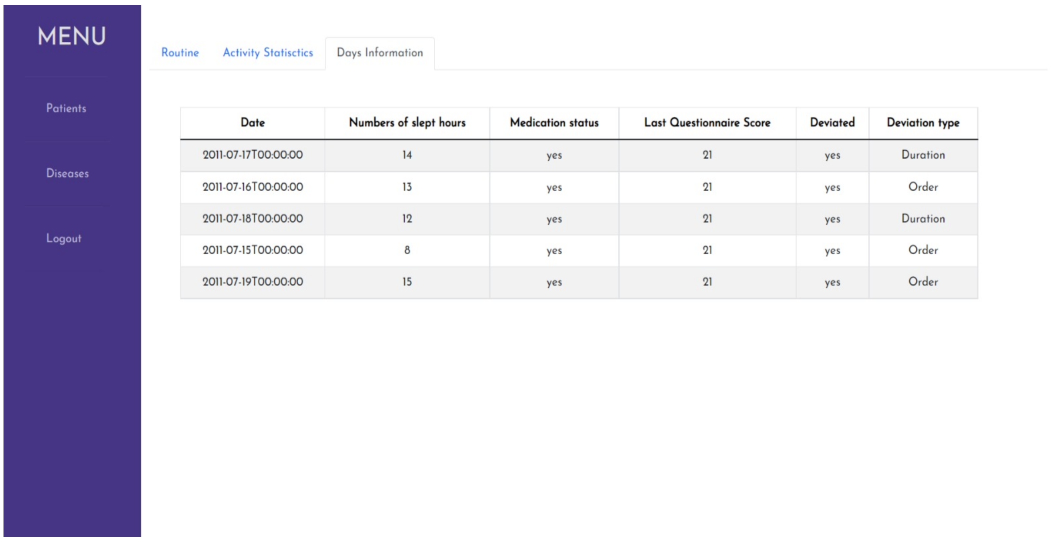


Figure 6.4: Routine Page

After the message the doctor can navigate by pressing the Days Information Tab to see a table with 6 columns and the number of rows equal to the days he requested to see if are deviated. The table contains the next columns:

1. **Date**
   - To see the date of the day he requested to see if it's deviated
2. **Numbers of slept hours**
   - To see the numbers of hours the patient has slept in that specific day, number that will help us to generate a recommendation for the patient.
3. **Medication Status**
   - Another variable that will help us generate recommendations but this time this column indicates if the patient took his medication.
4. **Last Questionnaire Score**
   - Along the two variables from above this one is the last that helps us generate a recommendation, this column indicating the score from the newest questionnaire taken by the patient regarding his mental state.
5. **Deviated**
   - This column indicates if the given day is deviated or not
6. **Deviation type**
   - The last column is filled only if the one before is true, this one is informing the doctor about the type of deviation that the day has, "Order" deviation means that the activities had an unusual chain of events in comparison with the routine. "Duration" means that the activities in the day had an unusual duration, short or long in comparison with activities in the routine.

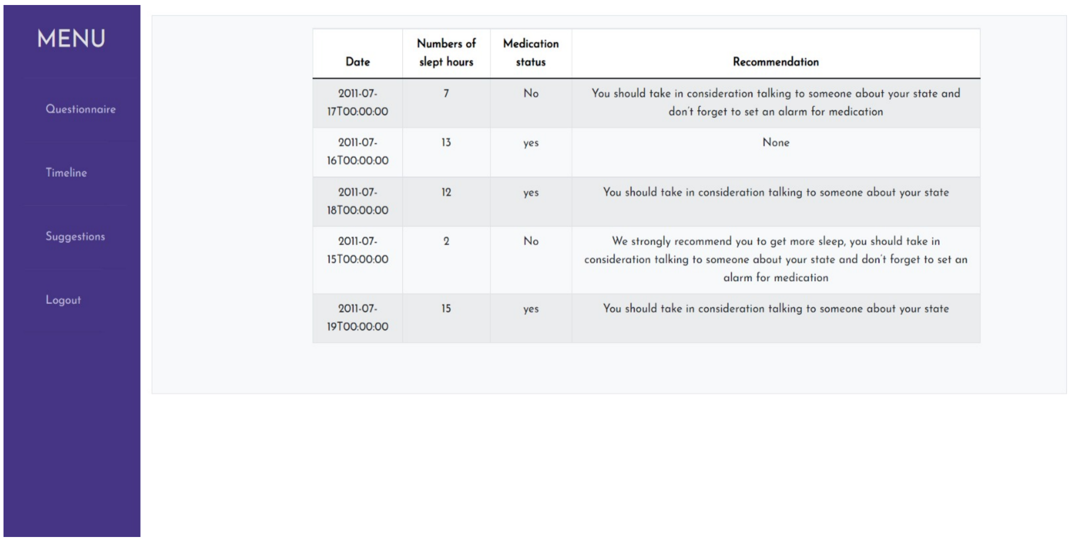


Date	Numbers of slept hours	Medication status	Last Questionnaire Score	Deviated	Deviation type
2011-07-17T00:00:00	14	yes	21	yes	Duration
2011-07-16T00:00:00	13	yes	21	yes	Order
2011-07-18T00:00:00	12	yes	21	yes	Duration
2011-07-15T00:00:00	8	yes	21	yes	Order
2011-07-19T00:00:00	15	yes	21	yes	Order

Figure 6.5: Days Information

For each day the doctor requested to see if it is deviated the patient receives a recommendation for that certain day. The request to the recommendation generator is made when the doctor wants to view the ”Days Information” tab or when the patient acceses the ”Sugestions” from the navigation menu. After the recommendations are returned from the component the deals with them, a table generates that contains the following columns:

1. **Date**
2. **Numbers of slept hours**
3. **Medication status**
4. **The recommendation generated**



Date	Numbers of slept hours	Medication status	Recommendation
2011-07-17T00:00:00	7	No	You should take in consideration talking to someone about your state and don't forget to set an alarm for medication
2011-07-16T00:00:00	13	yes	None
2011-07-18T00:00:00	12	yes	You should take in consideration talking to someone about your state
2011-07-15T00:00:00	2	No	We strongly recommend you to get more sleep, you should take in consideration talking to someone about your state and don't forget to set an alarm for medication
2011-07-19T00:00:00	15	yes	You should take in consideration talking to someone about your state

Figure 6.6: Days Information

The recommendations are generated by keeping in mind the variables that we keep account of from each day.

6.3. Metrics

The metrics we are keeping account of in the deviation detection component and also in the recommendation generator are as follows:

For deviation detection:

1. **Cosine similarity**
   - This real number that is contained in the [0,1] interval, shows us if the two vectors we are comparing point in the same direction. The closer the value to one, the closer the direction the two vectors point and this translates into the fact that the distance between the two vectors is smaller. This shows us that the similarity between the vectors is higher.
2. **Mean Deviation**
   - We caclulate the mean deviation of the day we want chack if it is deviated or not. The mean is coming from the routine and we want to check how far are the activities duration from the selected day are from the mean of the routine.

For recommendation generator:

1. **Slept hours**
   - This metric show us the numbers of slept hours of the patient that helps us generate recommendations, keeping track of this and the next two metrics.
2. **Medication status**
   - This boolean metric shows us if the patient has taken his/her medication
3. **Questionnaire score**
   - This number is placing the patient into one category of depression severity and by keeping track of it we can generate better recommendations

Chapter 7. User's manual

This chapter will provide a list of step-by-step instructions for installing and using the application, as well as a description of the minimal resources that are necessary for the application to function correctly.

7.1. Install Guide

For the instalation and runing the application we need a computer with network connection so we can install the necessary tools to start the app.

1. Microsoft Visual Studio 2019 or higher
   - The visual studio IDE is a powerful tool from Microsoft that will allow us to install NuGet Packages that are used in our applications as well as running the main back-end application. This app must be of type REST API CONTROLLER.
   - The Producer application is as well made in Visual Studio, it is a desktop application configured to be a producer of data.
2. Microsoft Visual Studio Code
   - We use visual code to edit and start the Front-End code

3. Node js and Vue Js
   - The main core of the Front-End application is Node.js this can be installed from the website.
   - Alongside this install we have used multiple libraries for front end like Bootstrap and Google Charts this libraries can be installed with npm comand and "-i" and the name of the library
4. PostgreSQL
   - The database can be installed from the website, it is free and open source.
5. IntelliJ Idea
   - The IntelliJ IDE helps us create the maven project that has WEKA dependencies. The project contains the Recommendation Generator API. We create a single controller that communicates with main app.

7.2. Start the application

For this part we will let you know how to configure all the components that must be started in order for the app to run smoothly and take advantage of all it's features.

1. The connection to the database is made by the connection string that has the **user:root** and **password:root**

2. The main application is easy to start since the Microsoft Visual Studio IDE has a run button.

3. The same thing applies to the producer app.

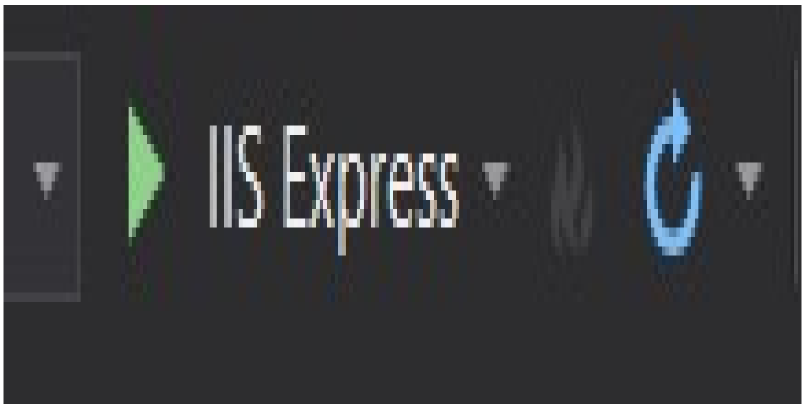
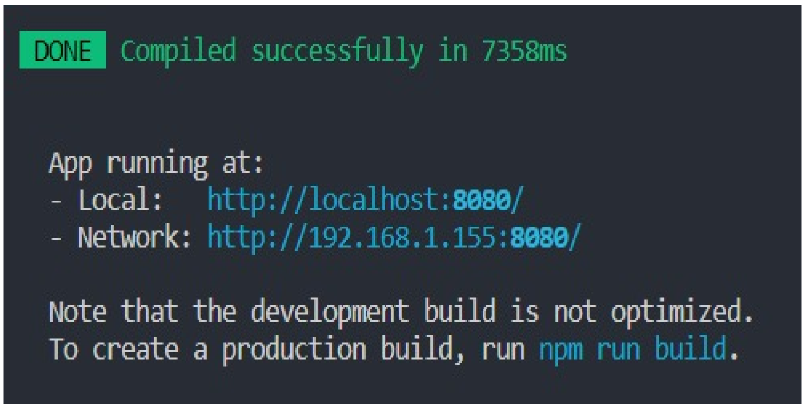


Figure 7.1: Run Back-end

4. The front end application is started via a terminal that points to the root folder of the app. The command to start the application is **npm run serve**.



```
 DONE  Compiled successfully in 7358ms


  App running at:
  - Local:   http://localhost:8080/
  - Network: http://192.168.1.155:8080/

  Note that the development build is not optimized.
  To create a production build, run npm run build.
```

Figure 7.2: Run Front End

5. As well as the Microsoft Visual Studio intelliJ Idea has a start button to start the REST API that waits to be called, but in the case of this application we need to add to the application.properties file the configuration **server.port = 8082** because the default port of the app is 8081 and conflicts with tha maien backend application.

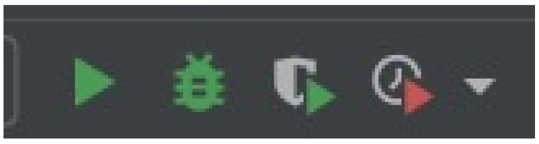
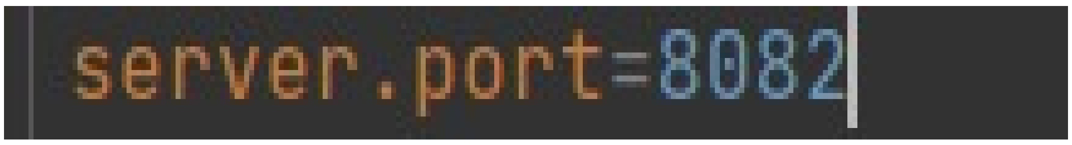


Figure 7.3: Run Recommendation generator



```
server.port=8082
```

Figure 7.4: Port setting Recommendation Generator

Chapter 8. Conclusions

In this chapter we will talk about our contributions as well as further development. The paper pursues to detect anomalies in daily life activities of elder people. The approach that the papers implements is to compare the routine that is generated from a dataset of daily life activities data with the day we want to classify as deviated or not. This approach uses algorithms to compare the days from two points of view.

The order of the activities and the duration of the activities. The order problem is solved using cosine similarity and keepin account of a threshold. Duration problem is solved with the help of the median deviation. Beside the deviation detection the paper implements the genration of recommendation based on healthcare data from the patient.

8.1. Contributions

- **Deviation detection using cosine similarity and median deviation**. Comparing the routine with the day we want to classify.
- **Recommendation Generator API**. The API component can be used to generate recommendations based on simple variables extracted from the dataset and patient input
- **Medical platform**. A medical platform that can be easily used and modified because of the layered architecture. This application can be integrated for a clinic or hospital web page.

8.2. Further development

For further development the system created can be integrated in much bigger systems of hospitals and also be modified to fit the requirments of the hospital/clinic. The front end can be modified to implement more requirments for patient. This requirments can inclunde a chat where the patient can contact an emergecny person or can contact a doctor. Also videos of healthy lifestyle and conferences about health in general can be added to the platform.

The deployment of the system in cloud can be a great further development for the application. Also the recommendation generator can be upgraded to a bigger recommendation system that keep account of multiple variables about the patient and adapted to the need of each hospital or clinic. This can be modified to give recommendations for healthy eating, daily activities and drinking water for a complete set of recommendations that induce a healthy lifestyle.

Along the changes at the recommendations, a further development that can help the patients would be to implement webscokets that will notify users in real time based on live data that have been colected from wearable devices or smartphones. The notification could be sent to a sibling or doctor if soimething is wrong with the vitals of the patient. Or to the patient smartphone if the notification is just a warning.

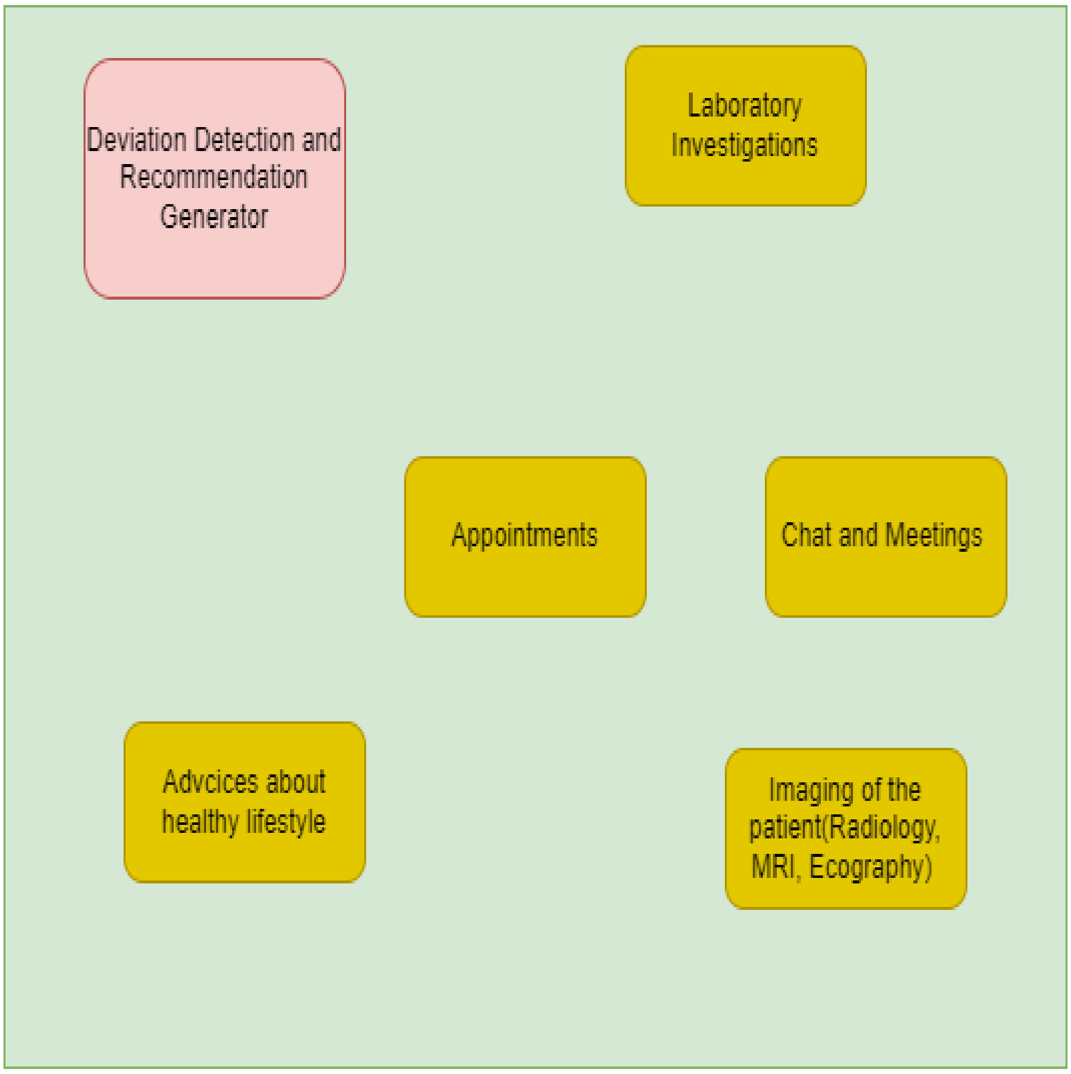

Deviation Detection and Recommendation Generator
Laboratory Investigations
Appointments
Chat and Meetings
Advcices about healthy lifestyle
Imaging of the patient(Radiology, MRI, Ecography)

Figure 8.1: Further development: Integrating our system in a bigger hospital system

Bibliography

[1] "World data," https://ourworldindata.org/life-expectancy. [Online]. Available: https://ourworldindata.org/life-expectancy.

[2] "Cdc data," https://www.cdc.gov/aging/data/subjective-cognitive-decline-brief.html. [Online]. Available: https://www.cdc.gov/aging/data/subjective-cognitive-decline-brief.html

[3] "Caregivers data," https://aspe.hhs.gov/reports/profile-older-adults-dementia-their-caregivers-issue-brief-0. [Online]. Available: https://aspe.hhs.gov/reports/profile-older-adults-dementia-their-caregivers-issue-brief-0

[4] R. Y.Tong and J.Gao., "Hidden state conditional random field for abnormal activity recognition in smart homes." *Entropy*, vol. 17, no. 3, p. 1358, 2015.

[5] G. Z. D.Riboni, C.Bettini and R.Helaoui, "Fine-grained recognition of abnormal behaviours for early detection of mild cognitive impairment." *Entropy*, vol. 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), p. 149–154, 2015.

[6] D. R. Zaffar Haider Janjua and C. Bettini., "Towards automatic induction of abnormal behavioral patterns for recognizing mild cognitive impairment." *Entropy*, vol. In Proceedings of the 31st Annual ACM Symposium on Applied Computing, p. 143–148, 2016.

[7] D. J. Prafulla N.Dawadi and M. Schmitter-Edgecombe., "Modeling patterns of activities using activity curves." *Pervasive Mob. Comput*, vol. 28, no. C, pp. 51–58, 2016.

[8] M.-T. D. Oya Aran, Dairazalia Sanchez-Cortes and D. Gatica-Perez., "Anomaly detection in elderly daily behaviour in ambient sensing environments." *Human Behaviour Understanding: 7 th International Workshop*, p. 51–67, 2016.

[9] Z. T. S. F. Abdur Rahim Mohammad Forkan, Ibrahim Khalil and A. Bouras., "A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living." *Pattern Recognition,*, vol. 48, no. 3, p. 628 – 641, 2015.

[10] M.-E. D.J.Cook A.Crandall Adriana M.Seelye, "Naturalistic assessment of everyday activities and prompting technologies in mild cognitive impairment." *Journal International Neuropsychologly Soc*, vol. 4, p. 442–52, 2013.

[11] D. P.N.Dawadi and M.Schmitter-Edgecombe., “Automated cognitive health assessment using smart home monitoring of complex tasks.” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, p. 1302–1313, 2013.

[12] B. D.Goldberg, D.Nichols and D.Terry, “Using collaborative filtering to weave an information tapestry.” *Communications of the ACM*, vol. 35, no. 12, p. 61–70, 1992.

[13] U.Shardanand and P.Maes, “Social information filtering: algorithms for automating “word of mouth.” *Proceedings of the SIGCHI conference on Human factors in computing systems*, p. 210–217, 1995.

[14] J. B.Sarwar, G.Karypis and J.Riedl., “Item-based collaborative filtering recommendation algorithms.” *Proceedings of the 10th International Conference on World Wide Web.*

[15] X.Su and T.M.Khoshgoftaar, “A survey of collaborative filtering techniques.” *Advances in Artificial Intelligence*, vol. 3, p. 1–19, 2009.

[16] M.J.Pazzani, “A framework for collaborative, content-based and demographic filtering.” *Artificial Intelligence Review*, vol. 13, no. 5, p. 393–408, 1999.

[17] R.Burke, “Hybrid recommender systems: survey and experiments.” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, p. 331–370, 2002.

[18] B. F.Ricci, L.Rokach and P.Kantor, “Recommender systems handbook.” *Springer,*, 2011.

[19] “Weka Trees,” https://www.analyticsvidhya.com/blog/2020/03/decision-tree-weka-no-coding/. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/03/decision-tree-weka-no-coding/.

[20] “Jwt,” https://jwt.io/. [Online]. Available: https://jwt.io/

[21] “Casas dataset,” http://casas.wsu.edu/datasets/. [Online]. Available: http://casas.wsu.edu/datasets/