Private High School of Engineering and Technologies

End of Studies Project

Software Engineering

Nutritional Mobile Application For Patients On Dialysis

Performed by

NESRINE NOUIRA

ESPRIT supervisor : Mrs. **Asma Mabrouk**

Military Hospital supervisors : Gen. **Anis Baffoun**

Dr. **Mouna Ben Azaiz**

Academic year : 2021/2022

The Welcoming company's supervisors
Dr BAFFOUN Anis , Dr Ben Azaiz Mouna

DEDICATIONS

To My Dear Parents

In my lowest moments, you were my unwavering pillar. Despite everything, you brightened my youth and taught me to appreciate the simple things in life.Thank you for providing me with such wonderful memories and lessons. You have led me throughout my life and assisted me in achieving my goals.You've given me the freedom to try new things all while teaching me to be strong.I feel incredibly honored and fortunate to be your daughter.I'm not sure I'll ever be able to repay you, but I want you to know that I'm always here for you.Thank you once again for everything.

To My Dear Sister

There are no words enough to express my gratitude for your constant presence, kindness, and generosity. In my life, you are a pure beam of light. I'm not sure what would i do if you weren't there. Thank you for more than words can express. May you forever be front and center in my life, sister.

To My Dear Brother

Without you, my life would be incomplete. Even though we have a lot of disagreements, I know you always have my best interests at heart. I know I can depend on you to cheer me up and make things better at any time. I am really pleased to be your sister and can't wait to watch you succeed in life.

NOUIRA Nesrine

ACKNOWLEDGEMENT

First and foremost, I thank **GOD** Almighty for his blessings and graces, which have given me the strength and patience to finish this project.

After that, I would like to convey my gratefulness to **Dr. BEN AZAIZ Mouna**, my military hospital supervisor, for the excellent supervision she offered me, her availability, establishing a conducive environment for the smooth execution of this internship, and for always being flexible and responsive to my ideas.

I would like also to thank **Mr Sami** for providing me with a basic overview of the patients' condition and environment right at the start of the project.

I also want to express my deepest gratitude to **Ms. MABROUK Asma** for the time she devoted to coach and follow-up this project; for the guidance she offered me; and for the meetings that highlighted the process.

I would like to thank **Dr BAFFOUN Anis** , **Dr Hichri** and **Mr ZERAI Mourad** for guiding and inspiring me at the start of the project.

And last and not least, the member of the jury, please accept my sincere gratitude for taking the time to evaluate my work, and I hope it meets your expectations.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

GENERAL INTRODUCTION

The convergence of science and technology in our dynamic digital era has led to the development of revolutionary digital health gadgets that allow easy and accurate characterization of health and disease.

Technological advancements and the miniaturization of diagnostic instruments to modern smartphone connected and mobile health (mHealth) devices have sparked a surge in interest in patient care, promising to lower healthcare costs and improve outcomes.

This mHealth "hype" has now collided with the "real world," yielding valuable insights into how patients and practitioners use digital health devices, especially chronic kidney disease (**CKD**) patients, who are in need of all the tools they can get their hands on to combat and manage this life-threatening condition.

Patients with chronic kidney disease are in desperate need of advanced technical assistance, that having a smartphone at their disposal to monitor their health would make a huge impact.

As diet plays a vital role when kidney function is reduced the correct one stabilizes the metabolism and helps slow down further damage. Dietary advice is of prime importance as soon as the first signs of malnutrition appear and should be an integral part of patient monitoring. Some foods like bananas or meloukhiya are critical to kidney disease patients.

The main issue here is that patients can only ask for advice from their doctors, they cannot monitor their everyday nutritional status, which led to a lack of solid assistance.

In fact, the internet is full of useful data concerning food constitution and mineral intakes but it's only foreign food and recipes. That's why we came up with the idea of a mobile application that offer the potential to help people living with chronic kidney disease (**CKD**) manage diet-related

challenges.

The application contains a large database of nutrition values and mineral intakes in Tunisian food and recipes that are useful to the patient. Moreover, it has an intelligent system that is in charge of calculating the correct amount of minerals consumed based on the patient consumption with an alerting system that alerts users when the amount of minerals is too high. Furthermore, it has a journalistic system that keeps a trace of the user's diet and food consumption, which is helpful to both users and doctors.

CHAPTER

ONE

GENERAL CONTEXT OF THE PROJECT

Introduction

In this chapter, we will examine our project's overall structure. Starting with a broad view of the host organization followed by an understanding of the project's context, the raised issues, and the global requirement. We will then finish by presenting the technical background and the development methodology.

1.1 Host organization

The following is an overview presentation of the host organization, a briefing of its history and a description of its services.

1.1.1 Military Hospital of Tunis

The Principal Military Hospital of Instruction of Tunis (**HMPIT**) is one of the largest and most prestigious university hospitals in Africa. It is a health facility under the jurisdiction of the **Tunisian Ministry of National Defence**. It has the best-equipped nephrology department in all of Tunisia.

It's well-known for its high-quality service and ability to handle even the most difficult circumstances.

Figure 1.1: HMPIT Logo

1.1.2 History

The Military Hospital was established in **1958**, he succeeded to the military hospital "Louis Vaillard" inaugurated by the French army in **1887**.

The Military Hospital of Tunis, its origin was the only structure existing military health for different armies under the leadership of a civilian medical and paramedical staff.

In **1972**, the military hospital El Omrane Supreme Armed Forces decided in **1989** to award the new hospital Habib Thameur, the services of the Army Medical Corps, it was baptized Military Hospital of Tunis instruction.

1.1.3 Military health organization

The General Directorate of Military Health includes:

1. The administrative and financial division.
2. The sanitary control and evaluation division.
3. The direction of technical and scientific action and training.
4. The direction of medical support and supplies.
5. International Cooperation Department.

6. Medical information management service.

7. The following health structures are subject to the supervision of the General Directorate of Military Health:

    - **7** Health establishments
    - **11** Specialized health centers

8. Military health schools:

    - School of Military Health.
    - Armed Forces Health Services Application School.

1.1.4 Military Hospital Missions

The Military Hospital of Tunis Instruction is a teaching hospital public administrative institution with legal personality and financial autonomy, its mission is:

- To provide highly specialized care to military and civilian personnel of the Ministry of National Defence, and their families.
- Participate and contribute to undergraduate and postgraduate education in the fields of medicine, pharmacy and dentistry and the training of paramedical personnel.
- If necessary, surgeons and medical teams can be created within the hospital to ensure the support of business units or working inside or outside the country and within the framework of humanitarian.
- Take all necessary measures to ensure the proper execution of the plan of medical intervention in case of natural disasters or during military operations.
- Participate in any action for preventive medicine and health education.
- Initiates and participates in all scientific research, especially in medicine, pharmacy and dentistry.

1.2 Project context

In **2020** almost **10%** of the **Tunisian population** was diagnosed with **chronic kidney disease**, about **12,000** dialysis patients with an estimated rise of **1.200** to **1.300** new cases **per year** growing up to becoming a major public health issue.

Although health coverage has extended to a large majority of the population, unfortunately, it still disproportionately burdens low-income to middle-income families making them unable to monitor their diet consult with **dieticians**, when dietary advice is of prime importance as soon as the first signs of malnutrition appear and should be an integral part of patient monitoring.

After several studies and observing the patients' conditions, we came up with the idea to create a mobile application to help maintain the best quality of health coverage and ease the financial burdens.

On one hand, That's why we came up with the idea of a **mobile application** that on one hand, offers the potential to help people living with chronic kidney disease (**CKD**) manage diet-related challenges and avoid critical circumstances.

On the other hand, it helps doctors keep track of their patients diet and more that will be exposed later on in this section. We came up with the idea to create a mobile application to help maintain the best quality of health coverage and ease the financial burdens. For more understanding of the diet-related difficulties in the next section, we will provide a brief explanation of how some nutrition values affect patient health.

For more understanding of the diet-related difficulties in the next section, we will provide a brief explanation of how some nutrition values affect patient health.

1.2.1 Chronic kidney disease

Chronic kidney disease, sometimes known as chronic kidney failure, is a condition in which the kidneys gradually lose function. Wastes and excess fluids in the blood are filtered by the kidneys and expelled as urine. When chronic kidney disease progresses, the body can accumulate excessive amounts of fluid, electrolytes, and wastes.

The patient may have minimal signs or symptoms in the early stages of chronic renal disease. Chronic renal disease may not be seen until the kidneys' function has deteriorated considerably.

Chronic kidney disease treatment focuses on reducing the progression of kidney damage, which is generally accomplished by addressing the underlying cause. Without mechanical filtration (dialysis) or a kidney transplant, chronic kidney disease can develop into end-stage kidney failure, which is deadly.

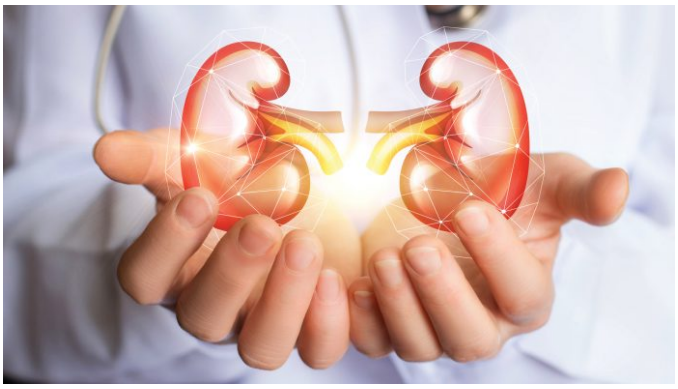


Figure 1.2: Kidneys

1.2.2 Renal Diet

People with compromised kidney function must adhere to a renal or kidney diet to cut down on the amount of waste in their blood.Food and drinks ingested are the sources of waste in the blood. The kidneys do not filter or eliminate waste adequately when their function is impaired. The electrolyte levels of a patient might be harmed if waste is left in the blood. A renal diet that is low in sodium, phosphorus, and protein can potentially assist to improve kidney function and delay the onset of total kidney failure.

A renal diet highlights the necessity of eating high-quality protein and, in most cases, reducing fluid intake. Some individuals may additionally require potassium and calcium restrictions. Because every person's body is different, it's critical that each patient engages with a renal dietitian to develop a diet that's suited to their specific needs. Below are the crucial substances to monitor to promote a renal diet.

1.2.2.1 Potassium

Potassium is a mineral that can be found in a variety of foods as well as in the human body. Potassium helps to maintain the heartbeat regular and the muscles in good operating order. Potassium is also required to keep the bloodstream's fluid and electrolyte balance.

The kidneys aid in maintaining a healthy potassium balance in the body by excreting excess potassium into the urine.

When the kidneys fail, the body's potassium levels rise because the kidneys are unable to eliminate extra potassium. Hyperkalemia is a condition in which there is too much potassium in the blood, which can lead to:

- Muscle weakness
- An irregular heart beat
- Slow pulse
- Heart attacks
- Death

1.2.2.2 Phosphorus

Phosphorus is an important mineral for bone health and development. Phosphorus is also important for the formation of connective tissue and organs, as well as the movement of muscles. When phosphorus-rich food is taken and digested, the phosphorus is absorbed by the small intestines and deposited in the bones.

Extra phosphorus in your blood can be removed by normal functioning kidneys. When kidney function is impaired, the kidneys are unable to eliminate excess phosphorus from the body. High phosphorus levels can deplete calcium in your bones, causing them to become brittle. Calcium deposits in the blood vessels, lungs, eyes, and heart are also a risk.

1.2.2.3 Sodium

One of the body's three major electrolytes that control the fluids going in and out of the body's tissues and cells. Sodium contributes to:

- Regulating blood pressure and blood volume
- Regulating nerve function and muscle contraction
- Regulating the acid-base balance of blood
- Balancing how much fluid the body keeps or eliminates

Too much sodium can be harmful for people with kidney disease because their kidneys cannot efficiently clear excess sodium and fluid from the body.

- Increased thirst
- Edema: swelling in the legs, hands, and face
- High blood pressure
- Heart failure: excess fluid in the bloodstream can overwork your heart, making it enlarged and weak
- Shortness of breath: fluid can build up in the lungs, making it difficult to breathe

1.2.2.4 Fluids

Fluid regulation is critical for patients with Chronic Kidney Disease in the latter stages since normal fluid consumption can lead to fluid buildup in the body, which can be harmful. Because dialysis patients' urine flow is generally reduced, extra fluid in the body can put undue strain on the heart and lungs.

1.3 Study of existing solutions

In this section we'll go over the existing solutions. We split the existing solutions into different categories: nutrition, medication and assistance.

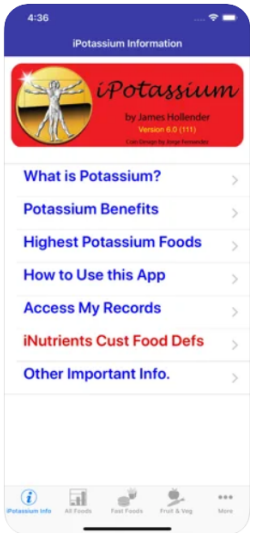
1.3.1 Nutrition

In this section, we will be analyzing solutions related to renal diets and nutrition for **CKD** patients. We'll start with an overview of the solution, then move on to its limitations.

iPotassium :

The iPotassium app gives you information about meals and how they're classified according to potassium content. This information is offered to help in the selection of foods to eat. Users may keep track of their food intake for any meal. The data is kept for a year and is given in reverse chronological order, sorted by day, with the most recent data at the top of the list.

- Pros
  - Food list with potassium content and code color
  - Possibility to search potassium by category (e.g. Fruits, Fish)
  - Tracking food intake
  - Customized serving size
- Cons
  - The database doesn't contain any recipes
  - Only potassium content tracking is provided
  - Unappealing user interface



(a) app features

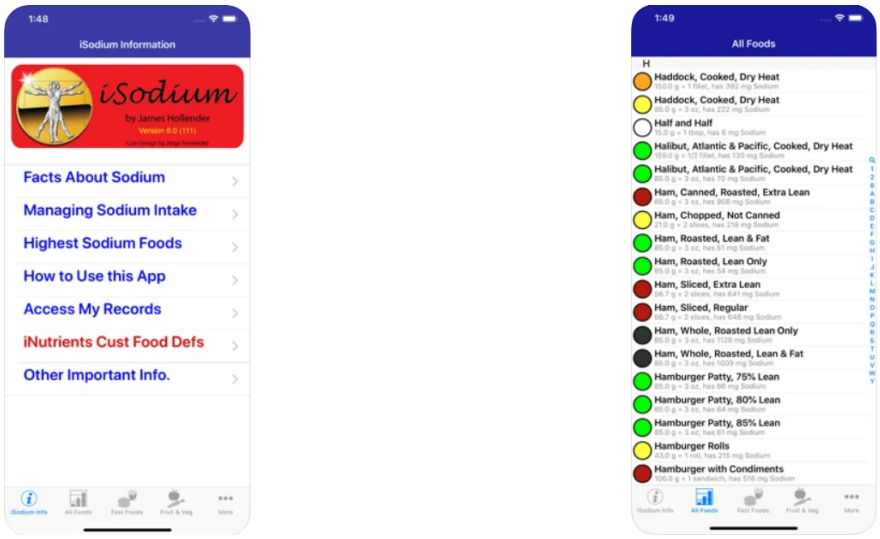
(b) Customized serving size

Figure 1.3: iPotassium interfaces

iSodium :

iSodium is an app that offers a large database of US foods ranked according to the content of sodium. The sodium levels in the food servings range from extremely low to extremely high presented using colors.

- Pros
  - Food list with sodium content and code color
  - Possibility to search food by category (e.g. Dairy, Meat)
- Cons
  - The database doesn't contain any recipes
  - Only Sodium content tracking is provided
  - Unappealing user interface
  - No food tracking



(a) app features (b) food list

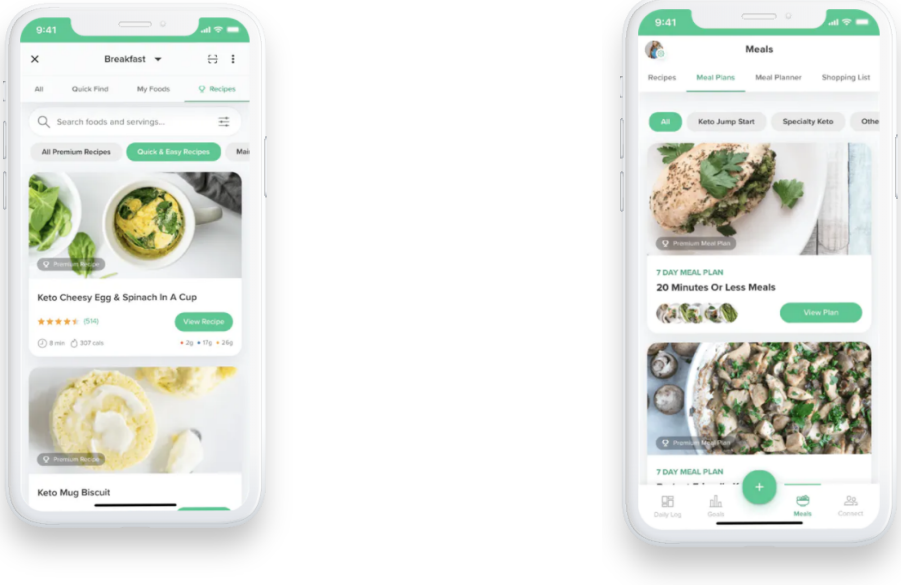
Figure 1.4: iSodium interfaces

Carb Manager

Carb Manager is a great option for people searching for a diet application that tracks calories, net carbs, ketones, and glycemic load, encourages healthy eating, and provides nutrition suggestions and information. In addition to the advanced options, the user can measure basic metrics like water intake, exercise, and body weight.

- Pros
  - More than one million foods in the app's database
  - Meals tracking
  - Water intake measurement
  - Displays progress towards reaching goals
  - Tracking can be done in different units of measure (ounces, cups, grams, kgs, etc.)
  - Fancy user interface
- Cons
  - The database doesn't contain any Tunisian recipes

- Only Water content tracking is provided
- Several tracking features are limited to the paid premium membership (including meal plans, recipes)
- Difficult to customize meal plans based on dietary restrictions or preferences



(a) Recipe list (b) Meal plan

Figure 1.5: Carb Manager interfaces

1.3.2 Medication

MyTherapy

MyTherapy is an application that helps remind users to stay in control of their medication, measurements, and exercise. Thanks to the app, users can focus on other important things in life and let the app manage their medication.

- Pros
  - Simple user interface
  - Free

- Cons
  - No cons to mention in this project context

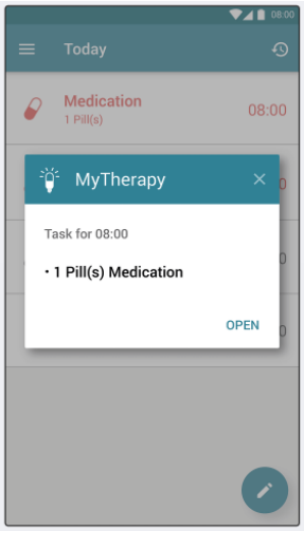


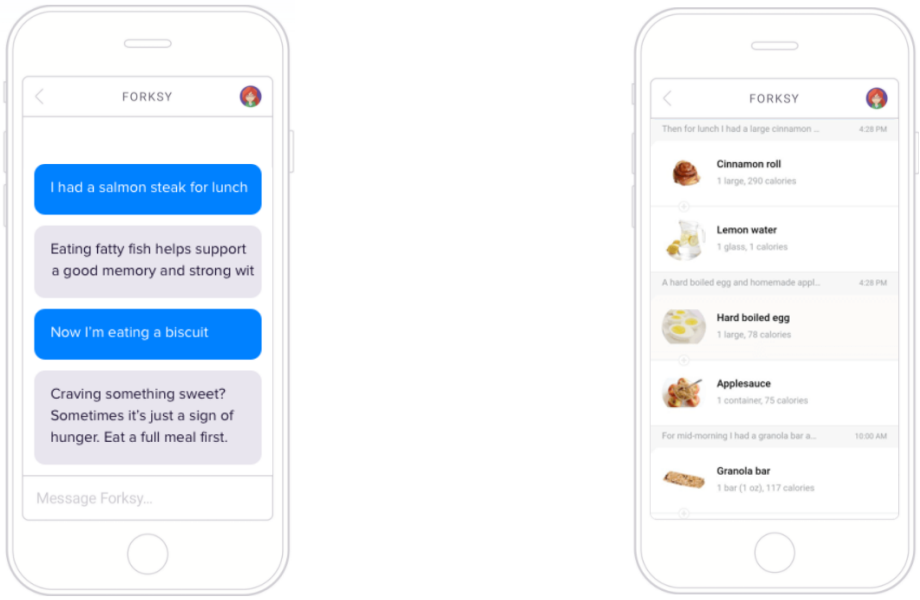
Figure 1.6: myTherapy interface

1.3.3 Assistance

Forsky AI

Forsky AI is a conversational food diary and AI for nutrition-related questions. The bot converses about diet an exercise. It can also ask the user to log their daily food.

- Pros
  - User can add meals to a food diary
  - User can ask questions about food content
- Cons
  - Communication in English
  - Only a free demo is available



(a) conversation with Frosky (b) food diary

Figure 1.7: Frosky AI interfaces

1.3.4 Issues

Having detailed the existing solutions, we can now move on to the issues. The major problem we're dealing with is the lack of an unified solution. An application that unifies the three categories of nutrition, medication, and assistance still does not exist. In addition to the shortcomings of existing solutions in the previously listed categories, the solutions' content is destined for people who want either to lose or gain weight. There is no specific solution for **CKD** patients. In most cases, the app provides tracking of one or two minerals mentioned in the first chapter. Here's a summary table 1.1 of the critical review:

App	Tracking	Meal Customization	Food Diary	Reminder	Assistance
iPotassium	potassium	only serving size	Diary	-	-
iSodium	sodium	-	-	-	-
Carb Manager	water	Customized	Diary	-	-
myTherapy	-	-	-	Med Reminder	-
Frosky	calories	Customized	Diary	-	Questions

Table 1.1: comparative table

1.3.5 Proposed solution

The proposed solution is a mobile app that has the potential to assist **CKD** patients in managing diet-related issues. A huge database of nutrition values and mineral intakes in Tunisian food and recipes must be included in the application. It also contains an intelligent system that calculates the correct amount of minerals consumed based on the patient's food intake, as well as an alerting system that warns users when the amount of minerals consumed is too high. It also features a journalistic system that records the user's diet and food intake, which is beneficial to both users and doctors. Finally, there will be a little chat-bot that will respond to some of the patient questions.

1.4 Methodology

Software development requires following a working methodology that ensures control throughout the development cycle and gives clear visibility into projects in order to reduce the complexity of problems and achieve business goals in less time and at a lower cost. As a result, we chose agile methodologies, notably the Scrum methodology, for this project.

1.4.1 Agile methodologies

Agile methodologies are a style of software development method. Because they have an iterative cycle and are dependent on changing customer requirements, there is a lot of feedback and interaction between the development team and the stakeholders. Due to their iterative approach, these

strategies minimize risks, provide organizations with the most return on investment, and result in user satisfaction.

1.4.2 Scrum

Scrum is an approach for building software that is iterative and adheres to agile principles and the Agile Manifesto. Scrum is formed of sprints, which are time blocks dedicated to delivering working software. A sprint usually lasts two to four weeks and is characterized by a goal or theme that helps to clarify the sprint's objectives.

Sprints are independent of change, allowing the developers to focus solely on creating functional product. Scrum focuses on assisting people who are committed to developing a project to deliver it in the best possible way.

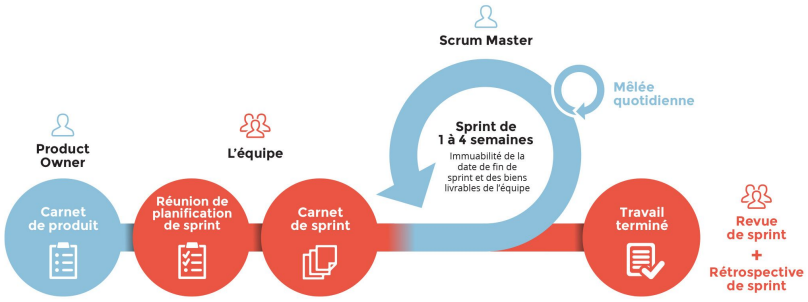


Figure 1.8: Scrum lifecycle

1.4.2.1 Roles

The major roles in Scrum are :

- **Scrum Master** : Their primary role is to guarantee that all Scrum team members understand and implement Scrum principles and practices. They set up daily meetings to handle disputes between members as well as any other concerns that may obstruct the sprint's progress.
- **Product Owner** : They are the only person that sets and prioritizes criteria for each sprint because they are the customer ambassador.
- **Development Team** : This group includes programmers, testers, front-end designers, and anyone else who is involved in the project from a different discipline.

1.4.2.2 Artifacts

The main artifacts of Scrum methodology are :

- **Product backlog** : It is a list of all work that the development team still has to do. The product owner is in charge of the product backlog, which is regularly prioritized.
- **Sprint backlog** : The sprint backlog is a subset of the product backlog, and it outlines all of the work that has to be done for a sprint.
- **Burn-down chart** : It's a visual representation of the remaining product development effort in a sprint.

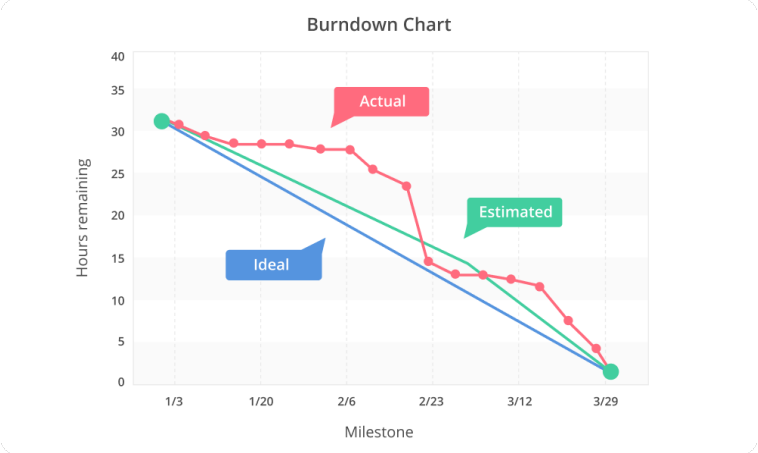


Figure 1.9: Burndown chart e.g.

Conclusion

We've taken a broad look at the project's background in this chapter, including some definitions and specific concerns. We went over the project's technical context, a detailed review of the existing solutions, outlined the issues, overview of the proposed solution, and finally the adopted methodology. The preliminary study will be presented in the following chapter, where we will analyze the technological context.

CHAPTER

TWO

SPRINT 0 : REQUIREMENT SPECIFICATION & ANALYSIS

2.1 Introduction

This chapter is an analytical phase in which we fully explain the behavior of our system. We start by describing the primary actors in the system, then cycle through the various functional and nonfunctional needs. Then, using use case diagrams and system sequence diagrams, we evaluate those requirements, describing every potential interaction between the actors and the system to be.

2.2 Actors

In our system we have two actors :

- **Patient** : The patient or his or her companion are the key actors in our application, and they use all of the features available.
- **Doctor** : All patients' journals, which contain information on their daily food, are accessible to doctors with a doctor account.

After defining the system actors, we move on the the requirement analysis.

2.3 Requirement analysis

The requirements are separated into two categories: functional and non-functional requirements.

2.3.1 Functional requirements

Functional requirements are the contract that binds system developers and clients along. As a result, our system must include the following features:

- **The application should allow the patient to** :
  - Manage a daily log
  - Customize meals
  - Manage food and recepies
  - Manage dialysis sessions reminder
  - Manage medication reminder
- **The application should allow the doctor to** :
  - Consult all patient journals

2.3.2 Non-functional requirements

Our major goal is to provide a high-performing solution. A solution that simply meets the functional criteria, on the other hand, is unlikely to satisfy the customer. As a result, when creating the solution, we must additionally consider non-functional requirements:

- **Ergonomics**:

  Functionality and aesthetics are in constant battle to deliver the very best experience. Aesthetically attractive design creates a pleasant impression in the user, allowing them to overlook minor usability flaws and provide better feedback. Users should be able to utilize the program without having to spend a long time looking for it thanks to the efficient navigation system. As a result, the question "how do I do this?" should never be asked.

- **Consistency**:

  The system's data should be taken from reliable sources and is calculated with precision.

- **Performance**:

  The system must have an acceptable response time while saving and manipulating the massive data.

2.4 Product Backlog

The Product Backlog, as previously stated, is the set of qualities and functions that form the intended product. User stories are the functional characteristics. The following are characteristics of user stories:

- **Id**: a unique identification for the particular story.
- **Description** : outlines an actor's requirement.
- **Acceptance criteria**: each user story has criteria linked with it, allowing the client to test it.
- **Estimation**: is an integer number that belongs to the Fibonacci sequence and is used to assess the story complexity.
- **Priority**: Priorities are used to establish the order in which things should be done, and they help to create a flow of stories that will keep the team going.

We sought to create user stories that fulfilled the requirements under the term **INVEST** while constructing our Backlog.

- **I - Independent**: Product Backlog Items should be self-contained
- **N - Negotiable** : Product Backlog Items should leave space for discussion.
- **V - Valuable**: Product Backlog Items must deliver value to the stakeholders.
- **E - Estimable**: We must always be able to estimate the size of a Product Backlog Item.

- **S - Small**: Product Backlog Item should not be so big as to become impossible to plan / task / prioritize within a level of accuracy.
- **T - Testable**: Product Backlog Item must provide the necessary information to be tested.

Product Backlog :

ID	User story	Acceptance criteria	Priority	Estimation
AN-01	As a patient, I am able to login to my account	Patient should login successfully	1	3
AN-02	As a patient, I am able to have access to food list	The food list must be displayed	2	7
AN-03	As a patient, I am able to have access to recipe list	The recipe list must be displayed	3	7
AN-04	As a patient, I am able to acces recipe details	The recipe details must be displayed	4	3
AN-05	As a patient, I am able to have a daily log	Patient should have access to his daily log	5	5
AN-06	As a patient, I am able to add food to his daily log from the food list	Food should be added to the patient daily log	6	2
AN-07	As a patient, I am able to add recipe to his daily log from the recipe list	Recipe should be added to the patient daily log	7	2
AN-08	As a patient, I am able to customize food portion	Nutrition values should dynamicallychange after changing the food portion	8	7
AN-09	As a patient, I am able to customize recipe portion	Nutrition values should dynamicallychange after changing the recipe portion	9	7
AN-10	As a patient, I am able to delete food from his daily log	Food should not be displayed in patient daily log	10	3
AN-11	As a patient, I am able to acces popular recipe list voted by other patients	The recipe list must be displayed	11	3
AN-12	As a patient, I am able to search for food by name dynamically	Search must display the correct result	12	3

AN-13	As a patient, I am able to search for recipe by name dynamically	Search must display the correct result	13	3
AN-14	As a patient, I am able to sort food list by multiple criteria	List must be sorted	14	1
AN-15	As a patient, I am able to sort recipe list by multiple criteria	List must be sorted	15	1
AN-16	As a patient, I am able to have an informative interface about his diet and daily minerale intakes	Intrface should load successfully	16	5
AN-17	As a patient, I am able to check his water intake	Water intake should be displayed as cups on the patient interface	17	3
AN-18	As a patient, I am able to update his water intake by clicking on water cups	Water cups should change state onclick empty/full state	18	3
AN-19	As a patient, I am able to save a custom meal	Meal should be saved and displayed in custom meals interface	19	5
AN-20	As a patient, I am able to have access to top 1500 rich food in potassium list	List should be displayed without much delay	20	3
AN-21	As a patient, I am able to add a custom recipe	Recipe should be added to database and displayed only for its owner	21	10
AN-22	As a patient, I am able to add a medicine	Medicine should be added successfully to database	22	7
AN-23	As a patient, I am able to add a dialysis session	Dialysis session should be added successfully to database	23	3
AN-24	As a patient, I am able to see his medicine list	Medicine list should be displayed	24	1
AN-25	As a patient, I am able to see his dialysis session list	Dialysis session list should be displayed	25	1
AN-26	As a patient, I am able to delete a medicine	Medicine should be deleted successfully	26	1
AN-27	As a patient, I am able to delete dialysis session	Dialysis session should be deleted successfully	27	1

AN-28	As a patient, I am able to have a medicine reminder that sends a custom notifications	Notification should be specific and on time	28	5
AN-29	As a patient, I am able to have a dialysis session reminder that sends a custom notifications	Notification should be specific and on time	29	5
AN-30	As a patient, I am able to have a water reminder that sends a custom notifications	Notification should be specific and on time	30	5
AN-31	As a patient, I am able to have access to his monthly log	Monthly log should be displayed successfully	31	2
AN-32	As a patient, I am able to acces every day meals through the monthly log	Meals should be displayed in the correct date	32	5
AN-33	As a Doctor, I am able to login as a doctor	Doctor should be able to login successfuly not as a patint but as a doctor	33	1
AN-34	As a Doctor, I am able to have the list of all registred patient	Patient List should be displayed	34	2
AN-35	As a Doctor, I am able to for a registred patient	Search result must be correct	35	1
AN-36	As a Doctor, I am able to acces patient monthly log	Monthly log should be displayed successfully	36	2
AN-37	As a Doctor, I am able to acces patient daily log through monthly log	Daily log should be displayed successfully	37	3
AP-38	As a patient, I am able to ask the chatbot Question	The chatbot must send reply to the patient	38	20
AN-39	As a patient, I am able to get a badge based on filling the daily log daily	if the patient meats the badge requirements badge should be displayed	39	5

Table 2.1: Product Backlog Sorted by priority

2.5 Global analysis

In this section we will be presenting the global use case diagram which we describe every possible interaction between the actors and the system, and the global class diagram.

2.5.1 Global use case diagram

Use case diagrams allow us to better describe the interactions between the different actors and the system.

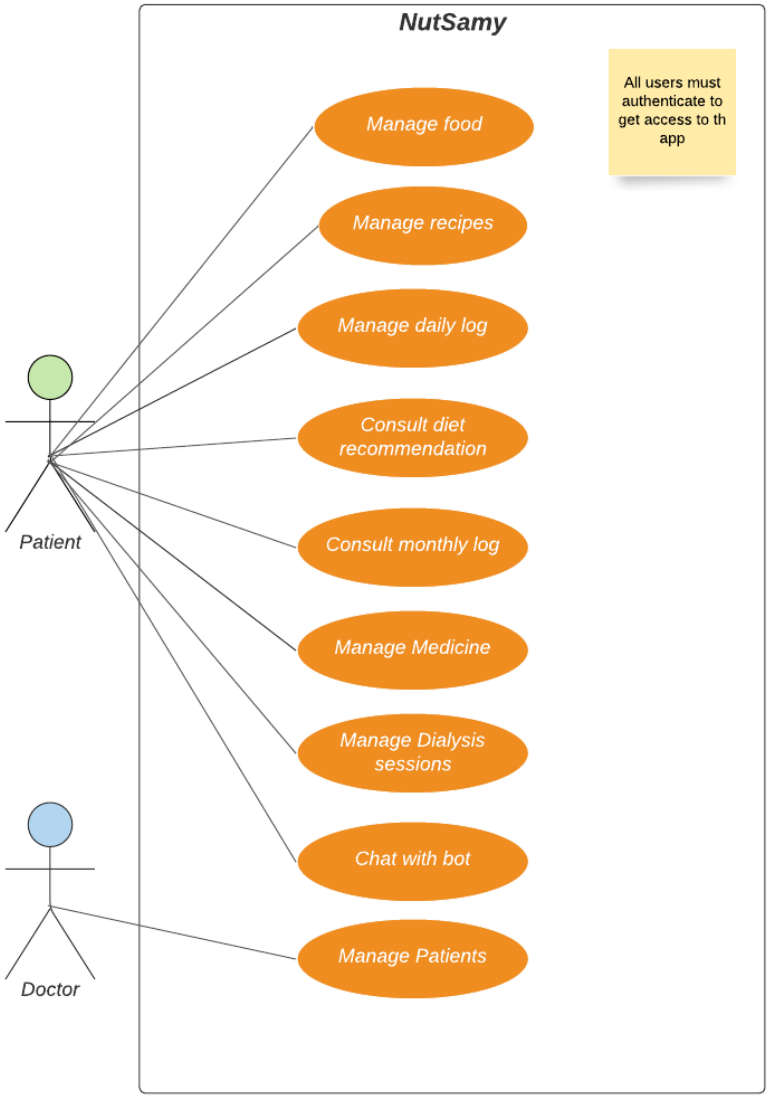


Figure 2.1: Global use case diagram

2.5.2 Global analysis class diagram

In this section we will be presenting the global class diagram used for data modeling.

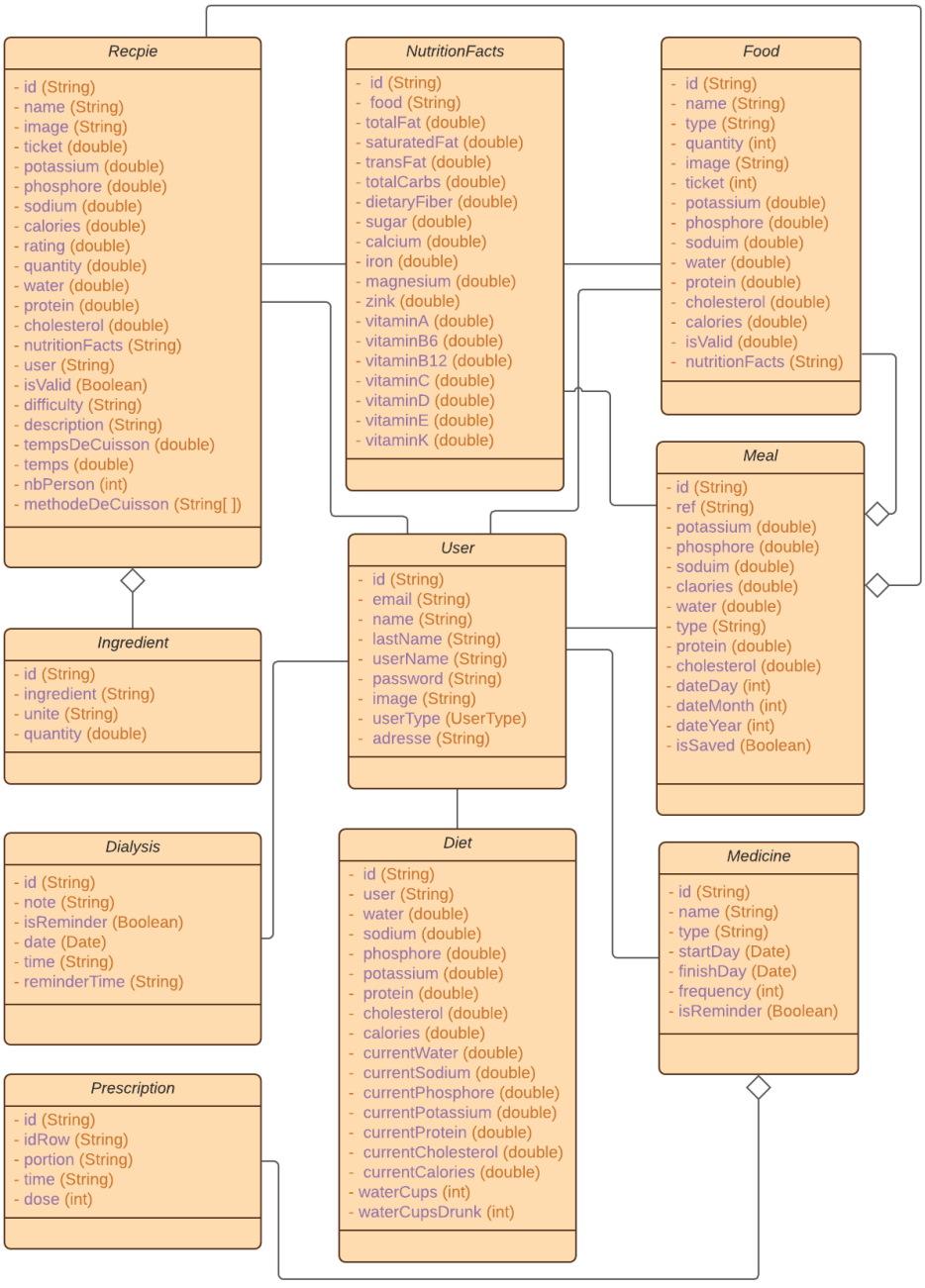


Figure 2.2: Global analysis class diagram

2.6 Technical choices

In this section we will be presenting the technical choices.

2.6.1 Mobile operating systems

This section is about mobile operating system studies (OS). Mobile OS, as specially designed software that runs on smartphones and other mobile devices, plays an important role in the proper operation of any mobile device. There are about a hundred mobile operating systems to choose from on the market today. However, in what follows, we'll look at the top two major ones that currently dominate the market and compare them in different ways.

1. **Android OS**

   The most popular mobile operating system nowadays has a large market share. Android was developed by a small startup then bought by **Google**. This expansion was really quick. Android's market share increased by **850%** from the second quarter of **2009** to the second quarter of **2010**, from **1.8%** to **17.2%**. Android had a **77.3%** share of the **global smartphone market** as of July **2018**. **Tunisia's Android market share** is has reached **90% in 2020**.

Figure 2.3: Android logo

2. **iOS**

   Many folks think that the iPhone, released in **2007**, revolutionized the concept of mobile devices. And their iOS operating system was groundbreaking in its own right. All Apple mobile devices, including the iPhone, iPad, iPod, and Apple Watch, run on iOS, which is derived from the Mac OS X operating system. Still, the iOS **market share** in **April 2020** was only **28.79%**, while the **Tunisian market share was 6%**.

Figure 2.4: iOS logo

2.6.2 Types of mobile applications

In fact there are three different ways to develop a mobile application.

1. **Native**

   Native apps are built and designed expressly for the operating system on which they operate like iOS and Android. One of the most significant advantages of developing a native app is that it fully utilizes the device's security, speed, and unique capabilities. For on-screen interactions or processing locally stored data, native code is the fastest. Because the app appears and functions like other applications on the user's smartphone, the user experience is typically considerably better. Furthermore, having access to a diverse set of APIs means that app usage is unrestricted. The expense of developing native apps is the most significant drawback. For each operating system, the developers must create a new app. As a result, you may require distinct developers for each platform on which you want your program to run.

2. **Hybrid**

   Hybrid applications are basically online apps wrapped in a native wrapper and are built utilizing a multi-platform technology driven by JavaScript, CSS, HTML5, or all three. Below are some examples of hybrid frameworks:

   - React Native (developed by Facebook)
   - Angular Mobile (developed by Google)
   - Flutter (developed by Google)
   - Xamarin (developed by Microsoft)

   All of the benefits of hybrid applications arise from the fact that, rather than creating numerous native apps for each operating system, you just have to create one core program and adapt it to operate

across all platforms. There are fewer developers, faster and easier upgrades, and cheaper maintenance costs with an unified code base. Performance is one of the most significant drawbacks of hybrid apps. Getting such kinds of programs to function across platforms can be time-consuming. The entire cost of building and maintaining the app might often exceed the cost of developing distinct native applications, making the cost savings negligible.

3. **Progressive Web Applications**

   Progressive web applications (**PWAs**) are online applications that load like ordinary web pages or websites but may provide users with features like offline working, push notifications, and device hardware access that was previously only accessible in native mobile apps. **PWAs** are a new type of website that combines the open web standards provided by contemporary browsers with the advantages of a rich mobile experience. Simply defined, Progressive Web Apps (**PWAs**) are mobile web pages that are meant to function similarly to native apps.

2.6.2.1 Final choice

We decided to build a native mobile app. Because native applications are developed for a specific platform and compiled using the platform's core programming language and APIs, they have a very high level of performance; they are incredibly quick and responsive. As a result, the app is far more effective. And we picked Android as the operating system since it has 90% of the mobile market share in Tunisia in 2020.

2.6.3 Backend development

In the following part, we'll go over the most popular backend frameworks and languages for Android apps in 2021.

1. **Node.js**

   The following are some of the most popular websites and apps that use Node.js as their backend:

   - Netflix
   - Amazon
   - eBay

Node.js has grown in popularity in recent years as a result of its low weight and adaptability. To make programming easier, Node.js comes with a large library of JavaScript plugins. Node.js has risen in popularity as a result of its open-source design for online and mobile app development.

Figure 2.5: Node.js logo

- **Pros**
  - High Performance

    The JavaScript code is interpreted by Node.js using Google's V8 JavaScript engine. Because it supports non-blocking I/O operations, the runtime environment improves the speed of code execution.
  - Handles the Requests Simultaneously

    Because Node.js has the option of non-blocking I/O systems, it allows you to process multiple requests at the same time. The system can handle concurrent requests more efficiently than other systems, such as Ruby or Python. The incoming requests are organized and carried out in a timely and systematic manner.
  - Highly Extensible

    Node.js is well-known for its extensibility, which means that you may tweak and extend it to meet your specific needs. JSON can also be used to provide a means for data to be sent between a web server and a client.
- **Cons**
  - Unstable API

    The frequent API updates, which are often backward-incompatible, are one of the most major problems that drive them insane. Unfortunately, this forces them to alter the access code on a regular basis to keep up with the latest version of the Node.js API.

2. **Laravel**

Although Laravel is by far the most popular PHP framework, it is included for good measure when considering the enormous spectrum of choices. This adaptable functionality is integrated into the apps. There are additional modular packaging solutions in use.

Figure 2.6: Laravel logo

- **Pros**
  - Built-in access control system

    User authentication and role-based access control are within your control with Laravel. All of the essential options are already included. As a result, user identification logic and access control have become more sophisticated and, as a result, safer.
  - Reverse Routing

    This is a really helpful tool. It allows you to link to named routes from within the structure. When generating links, all you have to do is provide the name of the individual router, and the system will automatically insert the desired URL.
  - IoC Container

    Inversion of Control is an approach for creating a new object without the requirement for external libraries to be bootstrapped. You can retrieve the items from anywhere you're writing, and you'll never have to deal with hard solid constructions again.
- **Cons**
  - Difficulty with some updates

    Platforms with long-term support frequently have issues following upgrades. Laravel is no exception, which is why it is occasionally chastised. These minor issues, on the other hand, may be swiftly resolved, providing that programmers are quick enough to respond to updates and have adequate expertise.

3. **Python**

Python is steadily gaining in popularity. Because it is a highly secure programming language with several built-in libraries, certain functions do not require programming, and you can just insert code from the repository and be done. Python is also device-independent.

Figure 2.7: Python logo

- **Pros**
  - Rich standard library and ecosystem

    Python has a lot of pre-written code in its libraries. As a result, developers will save time by not having to create simple components. These libraries also enable programmers to handle and alter the data necessary for Machine Learning's continuous data processing.
  - Flask framework

    Flask is a microframework that is most often used to build tiny solutions with a focus on lean functionality. Prototypes are also created using the framework.
  - Server-Side Scripting

    One of the advantages of using Python for server-side scripting is its easy syntax, which, as previously said, substantially speeds up the process. The code is made up of functional modules and their connections, allowing you to run the program algorithm depending on user input.
- **Cons**
  - Speed Limitations

    Python is frequently chastised for its speed. It's an interpreted script language, which means it's slower than compiled rivals like C/C++ or Java because of the multiple ways it employs to translate code. Nonetheless, some Python benchmarks outperform C and C++ benchmarks.

2.6.3.1 Final choice

Node is clearly not as reliable as PHP, which is approximately 25 years old and has a multitude of excellent frameworks. Nonetheless, Node.js is rapidly evolving, has extremely high-performance indicators, and in certain cases beats PHP, which is why we choose it as a backend. Because of its asynchronous features, Node.js is also a superior alternative for creating data-intensive apps and managing API calls simultaneously.We are also going to use python to develop the chatbot, using its **nltk** library and **ML** algorithms.

2.6.4 Database

A database is a logically organized collection of structured data kept electronically in a computer system. A database management system is generally in charge of a database (DBMS). The data, the DBMS, and the applications that go with them are referred to as a database system, which is commonly abbreviated to a just database. We'll go through two distinct databases in this section MongoDB and MySQL:

- **MySQL**

  MySQL is a relational database management system (RDBMS) that is free and open-source. It was initially developed by MySQL AB, but it is now owned by Oracle.

  It works on the principle of storing data in rows and tables, which are then organized into a database. It accesses and transfers data using SQL, and commands like "SELECT," "UPDATE," "INSERT," and "DELETE" are used to manage it.

A/c number	First name	Last name	A/c Type	Branch
12345756453	Michael	Calder	Saving	Manhattan
12345978675	Nick	Brown	Current	Brooklyn

Figure 2.8: MySQL DB e.g.

- **MongoDB**

  MongoDB is a prominent document-oriented database that falls within the NoSQL database umbrella. It uses the document store format, which is a key-value pair format. BSON files, which are essentially a slightly modified form of JSON files, are used to store documents in MongoDB, and therefore all JS is supported.

Furthermore, MongoDB's schema-free design eliminates the need to provide a fixed structure. These models allow for the depiction of hierarchical relationships as well as the flexibility to modify the record's structure.

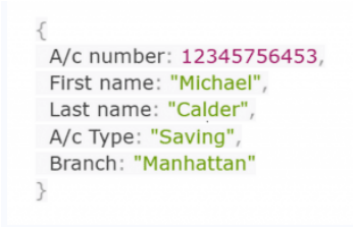


Figure 2.9: MongoDB e.g.

2.6.4.1 Final choice

We choose MongoDB as our database over MySQL because it is faster than MySQL due to its ability to handle large amounts of unstructured data and it is easier when handling unstructured data. When compared to MySQL, MongoDB's capacity to scale effectively is one of its strongest points. MongoDB provides flexible data models, allowing the database structure to adapt to changing business needs.

Figure 2.10: MongoDB Logo

2.6.5 Web services

A web service is any piece of software that uses a standardized XML communications protocol to make itself available over the internet. Web services are self-contained, modular, distributed, dynamic programs that may be specified, published, located, and used to build goods, processes, and supply chains through the internet. Local, distributed, or web-based applications are all possibilities. TCP/IP, HTTP, Java, HTML, and XML are all open standards that web services are based on. There are two types of web services:

- **SOAP Web Services**

SOAP is an XML-based protocol. The most significant benefit of utilizing the SOAP Web Service is its inherent security. The acronym SOAP refers to the Simple Object Access Protocol. A WSDL file is present in every application that serves SOAP queries. Web Service Description Language (WSDL) is an XML document. The WSDL specifies all of the web service's methods, as well as the request and response types. It defines the agreement between the service provider and the customer. SOAP was created as a means to deliver XML via HTTP to make remote procedure calls to distant objects.

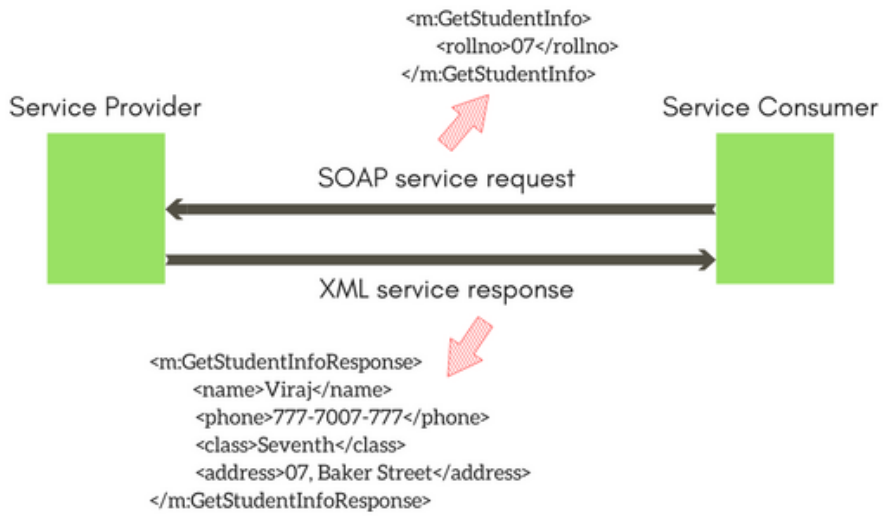


Figure 2.11: SOAP request e.g.

- **REST Web Services**

  The REST stands for Representational State Transfer.REST is a style of software design, not a collection of standards or regulations. RESTful applications are those that adhere to this architecture. Unlike SOAP, which focuses on activities, REST is primarily concerned with resources. REST locates resources via URLs, and the activities that can be performed on them are determined by the type of transport protocol. The REST service uses the URL to find the resource and the transport action verb to conduct the action. It is primarily based on architectural style and customs. HTTP Methods for RESTful Services:

  - GET: read an object.
  - POST: insert or create an object.
  - PUT: update an object

– DELETE: delete an object.

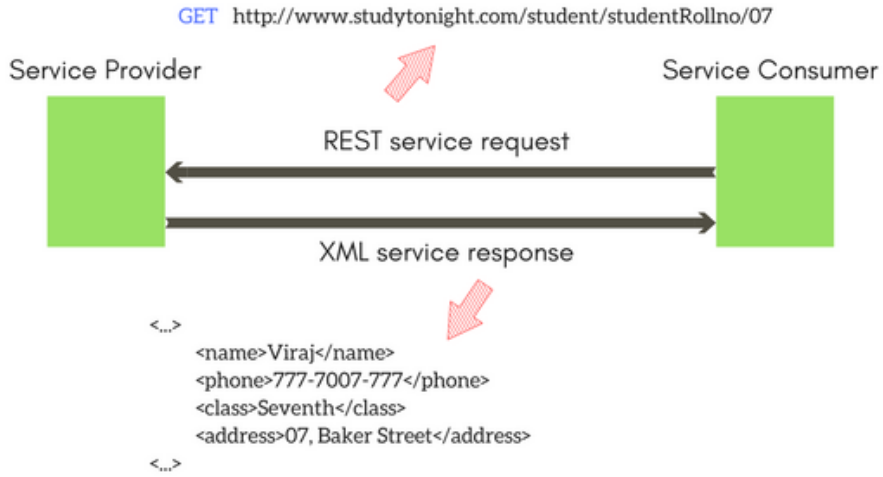


Figure 2.12: REST request e.g.

2.6.6 Difference between REST and SOAP

The following are some key distinctions between the two types of web services:

SOAP	REST
Only the XML format is supported by SOAP	REST supports a variety of data types, including plain text, HTML, JSON, and XML
SOAP is a set of standards or a protocol	REST is a software architectural style
SOAP has its own security layer that it definesr	The security features of REST are inherited from the underlying transport protocols
SOAP exposes business logic through the service interface	To expose business logic, REST employs URIs
Because SOAP is a protocol, it cannot be used with REST	Because REST is a concept, it can use any protocol, including HTTP and SOAP

Table 2.2: comparative table SOAP vs REST

2.6.6.1 Final choice

Because our solution does not necessitate a thorough mapping of object sets to the client, REST is a better choice than SOAP in this case. The transfer of object data requires a lot of bandwidth, which might be constrained in circumstances where connectivity is expensive. One of the use-cases where SOAP should be avoided at all costs is an API used mainly by mobile apps.

2.7 Global solution architecture

The figure 2.13 presents the solution's global architecture.

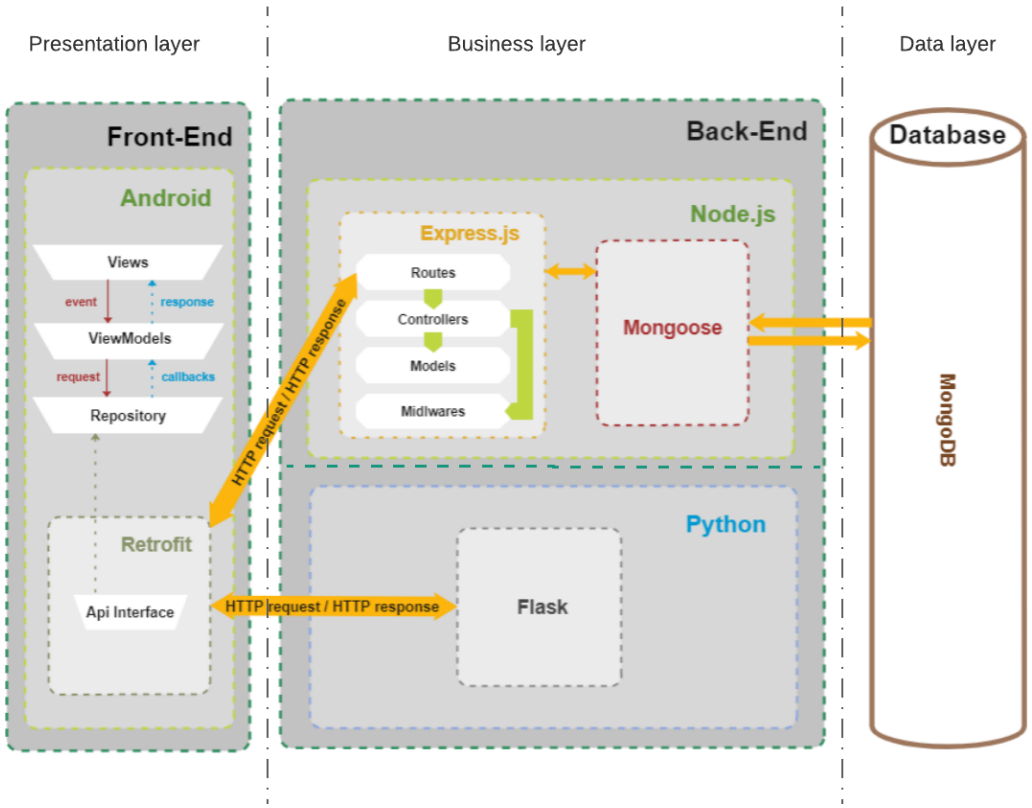


Figure 2.13: App Architecture

In this architecture We distinguish three logical layers : Data layer, Business layer, and Presentation layer.

- **Presentation layer** : This layer is the main interaction point with the users, this layer embraces UI components and UI process components.

- **Business layer** : It is the heart of the mobile app, exposing all of its features. To decrease burden, the business logic layer is placed on the back-end servers and accessed remotely by the mobile application.
- **Data layer** : The data layer is responsible of storing and managing the data of our project.

For the physical architecture we kept both presentation and data layer intact while we spliced the business layer into two. As we can see in the figure above we have our android client in the presentation layer, mongodb server in the data layer, and finally a python server which is responsible for the chatbot and a node.js server which is the responsible for all the app features. With one client in the front-end, one database server, and two servers in the back-end, our architecture is a N-tier architecture and specifically a 4-tier architecture.

We used the MVVM design pattern in the frontend. The Model-View-ViewModel (MVVM) design eliminates the tight connection between each component. Most significantly, the children in this design do not have a direct connection to the parent; instead, they rely on observables.

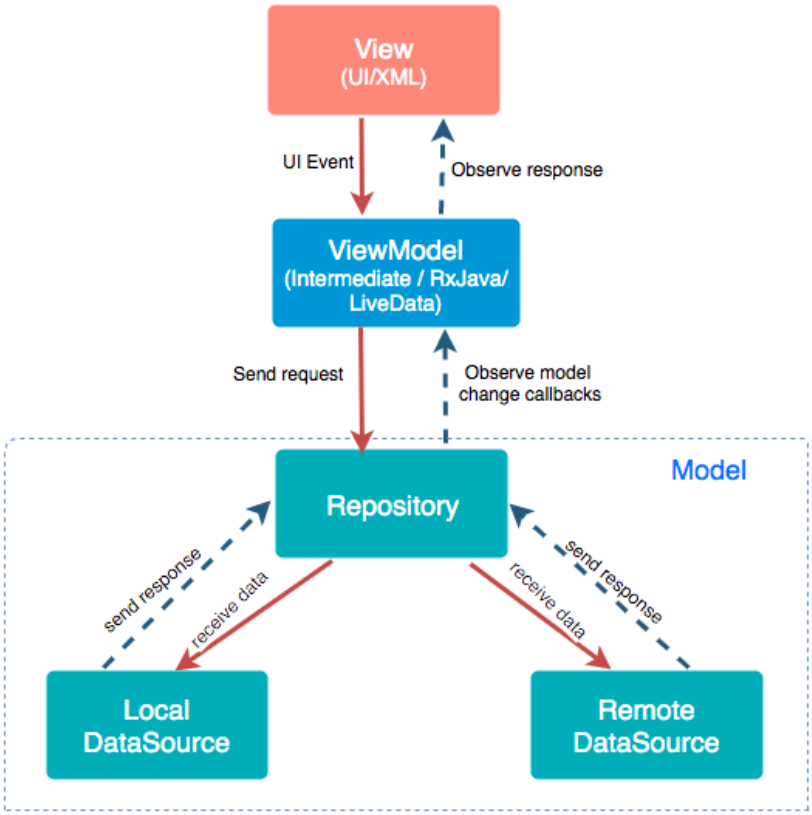


Figure 2.14: MVVM design pattern

- **Model** : It is where the Android Application's data and business logic are stored. The business logic

is comprised of local and remote data sources, model classes, and a repository.

- **View** : It is made up of UI code (Activity, Fragment) and XML. It communicates the user action to the ViewModel, but it does not receive a direct response.
- **Model** :It serves as a link between the View and the Model (business logic). The ViewModel should be unaware of the view with which it is interacting. It communicates with the Model and exposes the observable that the View may see.

We also used the MVC design pattern in the backend node.js server .An application must have a data model, presentation information, and control information, according to the model-view-controller (MVC) design pattern. Each of these must be divided into separate items according to the pattern.

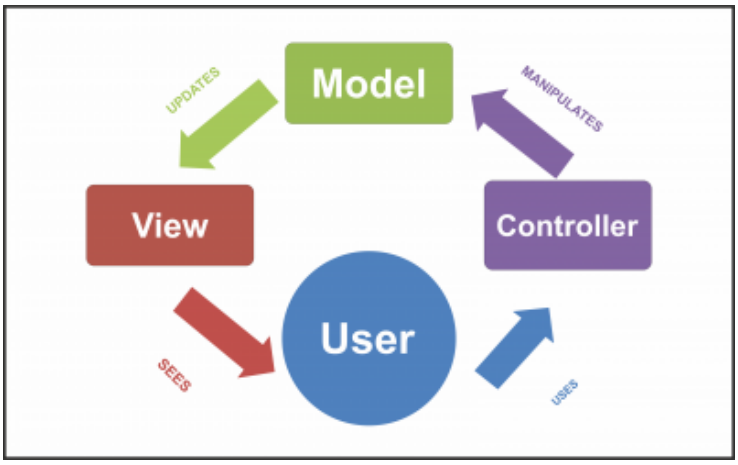


Figure 2.15: MVC design pattern

- **Model** : There is no logic indicating how to display the data to a user in the model; it merely holds the pure application data.
- **View** : The user can see the data from the model thanks to View. The view understands how to get to the data in the model, but it has no idea what that data represents or how the user can alter it.
- **Controller** : Between the view and the model is the Controller. It listens for events that are triggered by the view (or another external source) and responds appropriately.

2.8 Tools

The key tools used for the design, development, versioning, and writing of this report are presented in this part.

2.8.1 User interface design tools

- **Adobe Xd**

  Adobe XD is a user experience (UX) and user interface (UI) design tool developed by Adobe Inc. It's a versatile and simple-to-use vector-based experience design platform that gives teams the tools they need to collaborate on creating the world's top experiences.

Figure 2.16: Adobe Xd logo

- **Adobe Illustratorr**

  Adobe Illustrator is a vector-based design and drawing program for professionals. Illustrator may be used to create everything from single design elements to whole compositions when used as part of a wider design workflow. Illustrator is used by designers to make posters, symbols, logos, patterns, and icons, among other things.

Figure 2.17: Adobe Illustrator logo

2.8.2 Development Environment

- **Android Studio**

  Android Studio, which is based on IntelliJ IDEA, is the official Integrated Development Environment (IDE) for developing Android apps. Android Studio, in addition to IntelliJ's sophisticated code editor and development tools, has a number of capabilities that help you build Android apps more quickly. We'll be using Android Studio for developing the frontend of our app.

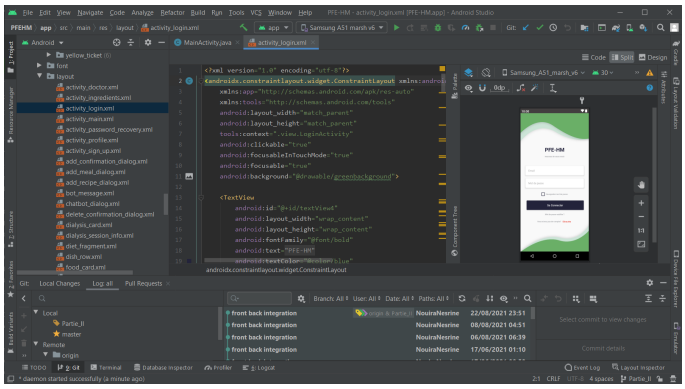


Figure 2.18: Android Studio

- **Visual Studio Code**

  Visual Studio Code is a desktop source code editor that is both lightweight and powerful. It features built-in support for JavaScript, TypeScript, and Node.js, as well as a diverse ecosystem of extensions for additional languages(including C++, C#, Java, Python, PHP, and Go) and runtimes (such as .NET and Unity). We'll be developing our backend (Node.js and python) with Visual Studio Code.

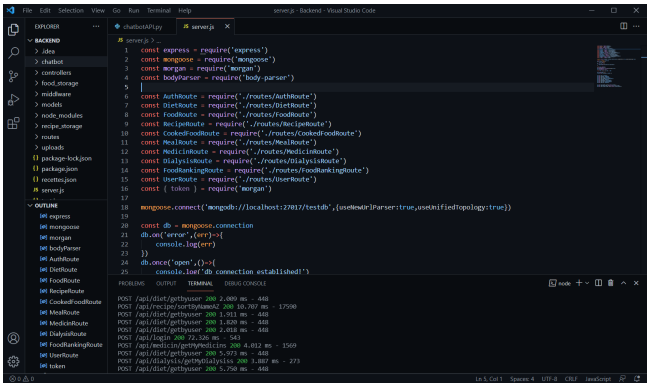


Figure 2.19: Visual Studio Code

- **MongoDB Compass**

  MongoDB Compass is a sophisticated visual interface for searching, collecting, and analyzing your MongoDB data. In just a few clicks, any user may see and explore your data using ad-hoc searches. We'll be using MongoDB Compass to visualize data.

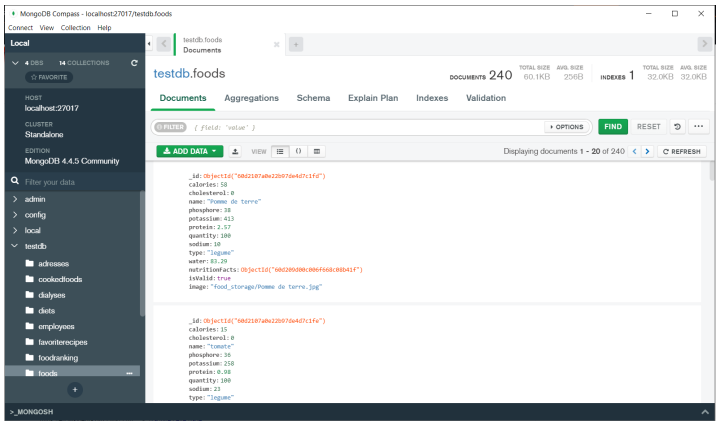


Figure 2.20: MongoDB Compass

- **Postman**

  Postman is an API client for developers that makes it simple to create, distribute, test, and document APIs. Users may generate and save basic and complicated HTTP/s queries, as well as view their answers, to do this. As a consequence, work is more efficient and less tiresome. We'll be using Postman to test our APIs.

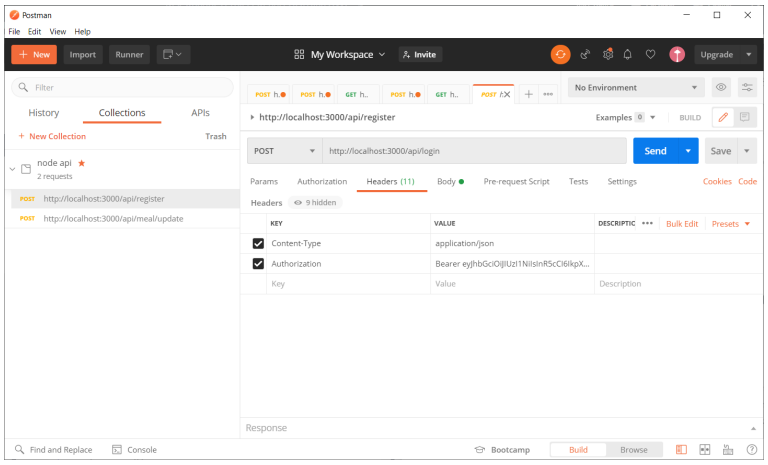


Figure 2.21: Postman

2.8.3 Version Control Tools

- **Git**

  Git is a decentralized version control system. It is intended to be useful for both minor and large tasks. Unlike svn or cvs, Git is a decentralized version control system, which means that instead of working on a centralized server, each user may work on their own repository.Also Android Studio has an amazing integration with Git for source control management which make Git the best choice.

Figure 2.22: Git

2.9 Sprint planning

We split our Release into four sprints based on our product backlog and the decomposition of our project into modules:

- **Sprint 1**: Food & Recipe Management
- **Sprint 2**: Log Management
- **Sprint 3**: Medication Management
- **Sprint 4**: Chatbot

2.10 Conclusion

The requirements analysis gave us a clearer picture of the subject and a better grasp of the challenges at hand. It also resulted in the creation of the Product Backlog, which will serve as the foundation for the implementation of our app.

CHAPTER

THREE

SPRINT 1 : FOOD & RECIPE MANAGEMENT

Introduction

Following a review of the project's division based on what we provided in the product's Backlog, we've settled on the following goal: "**Food & Recipe Management**" which means we'll strive to build all of the aspects involved in food management during this sprint. During this sprint, the patient will be able to browse different food and recipe lists and customize food portions.

3.1 Sprint Backlog

Sprint Backlog :

ID	User story	Acceptance criteria	Priority	Estimation
AN-01	As a patient, I am able to login to my account	Patient should login successfully	1	3
AN-02	As a patient, I am able to have access to food list	The food list must be displayed	2	7
AN-03	As a patient, I am able to have access to recipe list	The recipe list must be displayed	3	7

AN-04	As a patient, I am able to acces recipe details	The recipe details must be displayed	4	3
AN-08	As a patient, I am able to customize food portion	Nutrition values should dynamicallychange after changing the food portion	8	7
AN-09	As a patient, I am able to customize recipe portion	Nutrition values should dynamicallychange after changing the recipe portion	9	7
AN-11	As a patient, I am able to acces popular recipe list voted by other patients	The recipe list must be displayed	11	3
AN-12	As a patient, I am able to search for food by name dynamically	Search must display the correct result	12	3
AN-13	As a patient, I am able to search for recipe by name dynamically	Search must display the correct result	13	3
AN-14	As a patient, I am able to sort food list by multiple criteria	List must be sorted	14	1
AN-15	As a patient, I am able to sort recipe list by multiple criteria	List must be sorted	15	1
AN-20	As a patient, I am able to have access to top 1500 rich food in potassium list	List should be displayed without much delay	20	3
AN-21	As a patient, I am able to add a custom recipe	Recipe should be added to database and displayed only for its owner	21	10

Table 3.1: Food & Recipe Management Sprint Backlog Sorted by priority

3.2 Use case diagram

Use case diagrams are UML diagrams illustrate a software system's static and global functional behavior. The use case diagram for the first sprint is shown in the figure 5.4.

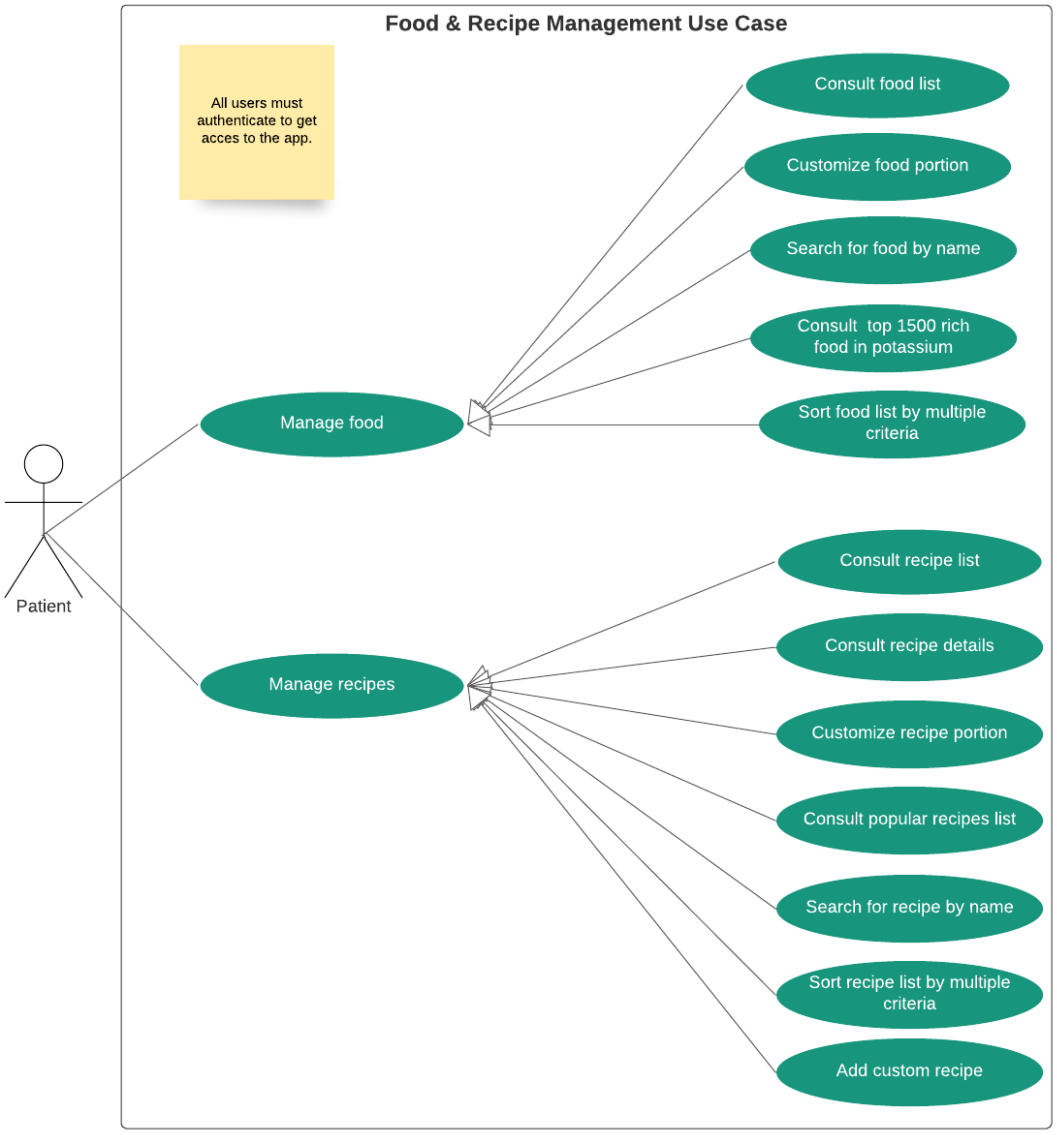


Figure 3.1: Food & Recipe Management Use Case

The diagram above depicts the functionalities that the user will be able to utilize following this sprint. The system sequence diagram will be used to provide a better and more dynamic vision of each use case.

3.3 Sequence diagrams

In the UML formulation, sequence diagrams are graphic representations of interactions between actors and the system in chronological order. The system sequence diagrams of the most significant defined use cases are then presented.

3.3.1 Sequence diagram : Authentication

To access specific areas of the application, users must first authenticate. The user must supply valid Login credentials that are associated with an existent account.

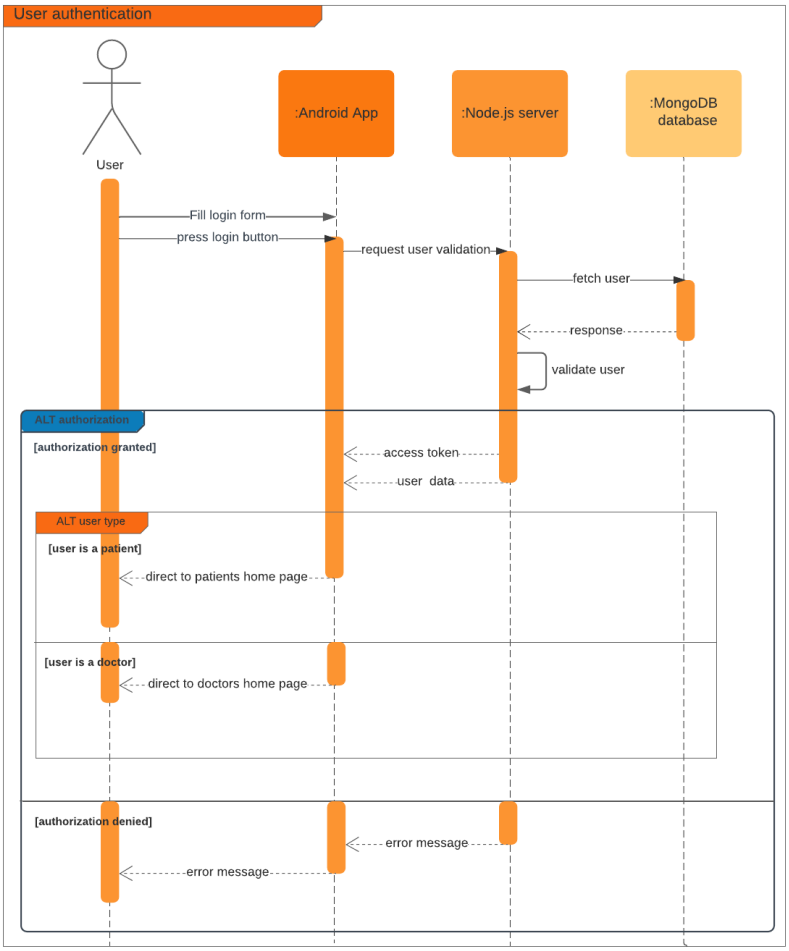


Figure 3.2: Sequence diagram : Authentication

3.3.1.1 Sequence diagram : Customize food portion

The patient can choose his or her own meal servings.

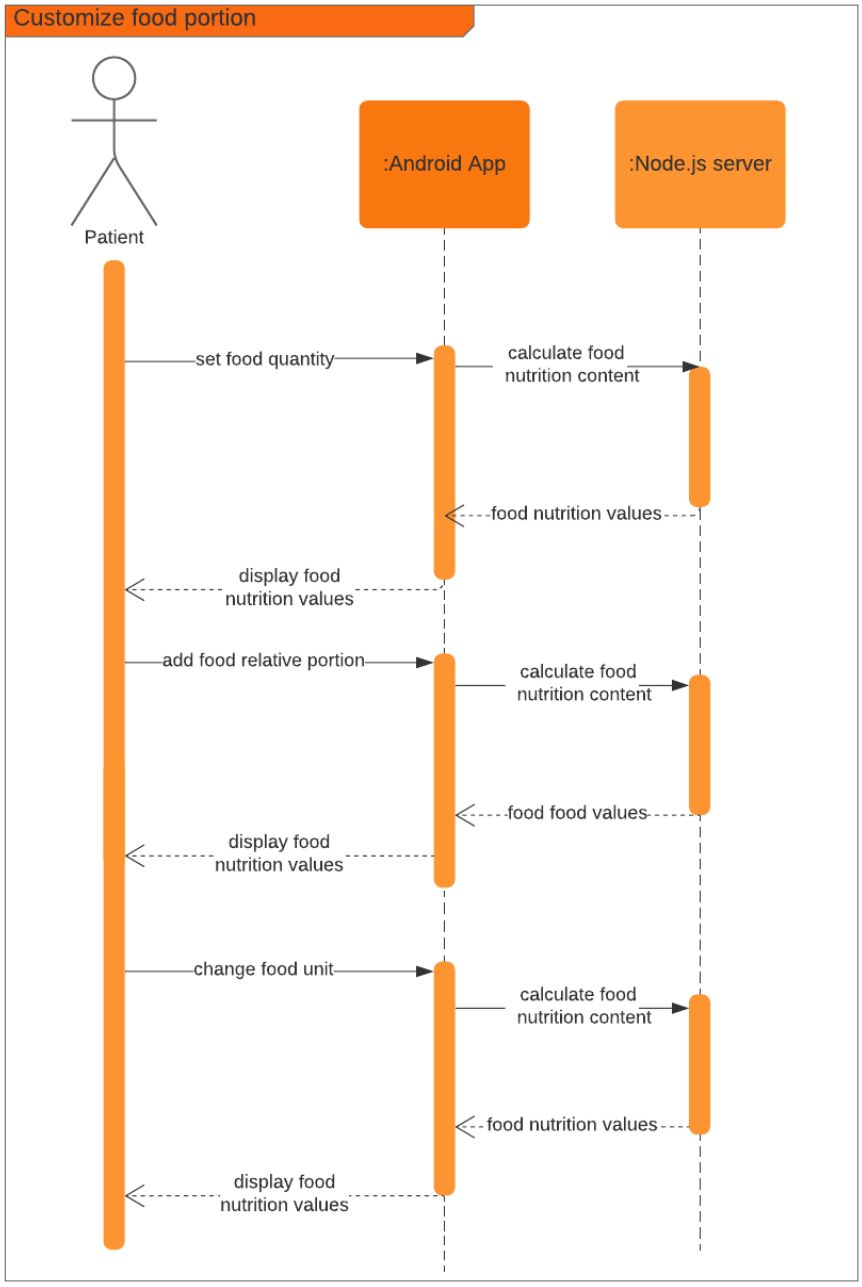


Figure 3.3: Sequence diagram : Customize food portion

3.3.2 Sequence diagram : Consult Popular Recipes list

The patient can browse a popular recipes.

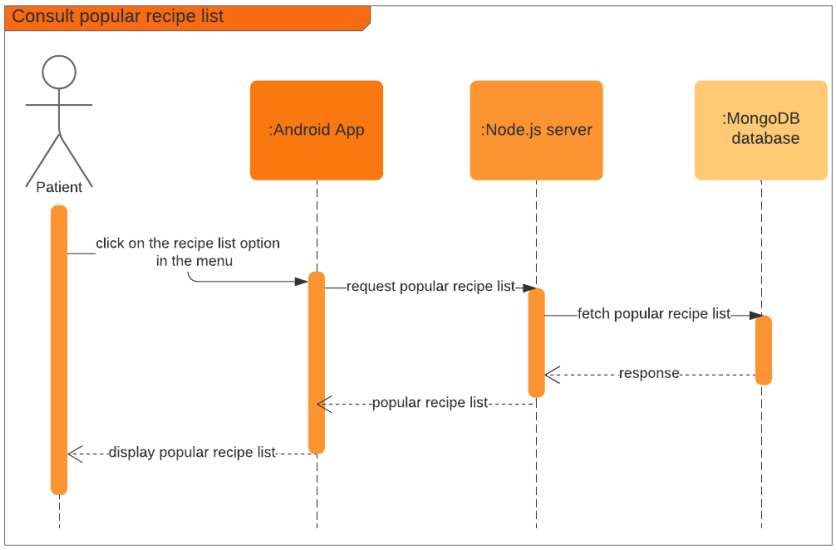


Figure 3.4: Consult popular recipes list sequence diagram

3.3.3 Sequence diagram : Search for recipe

The patient can search for recipes.

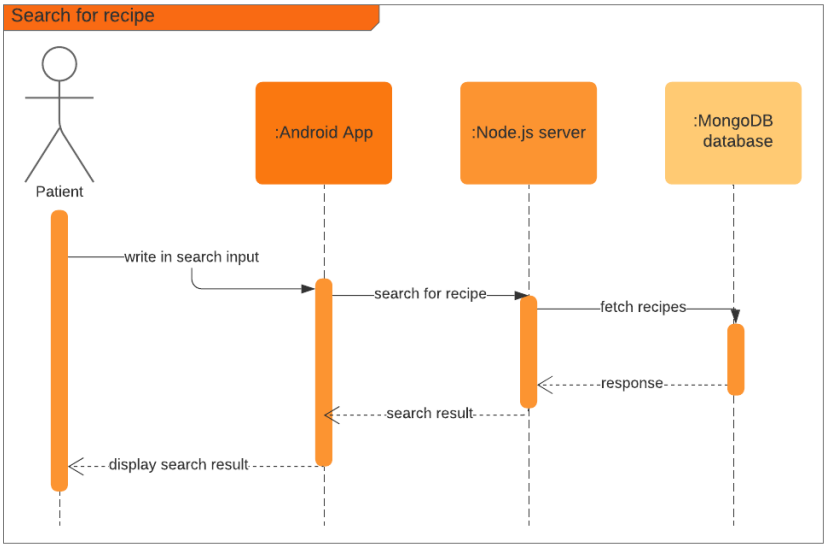


Figure 3.5: Sequence diagram : Search for recipe

3.3.3.1 Sequence diagram : Sort food list

The patient can sort food list by date , potassium content and alphabet.

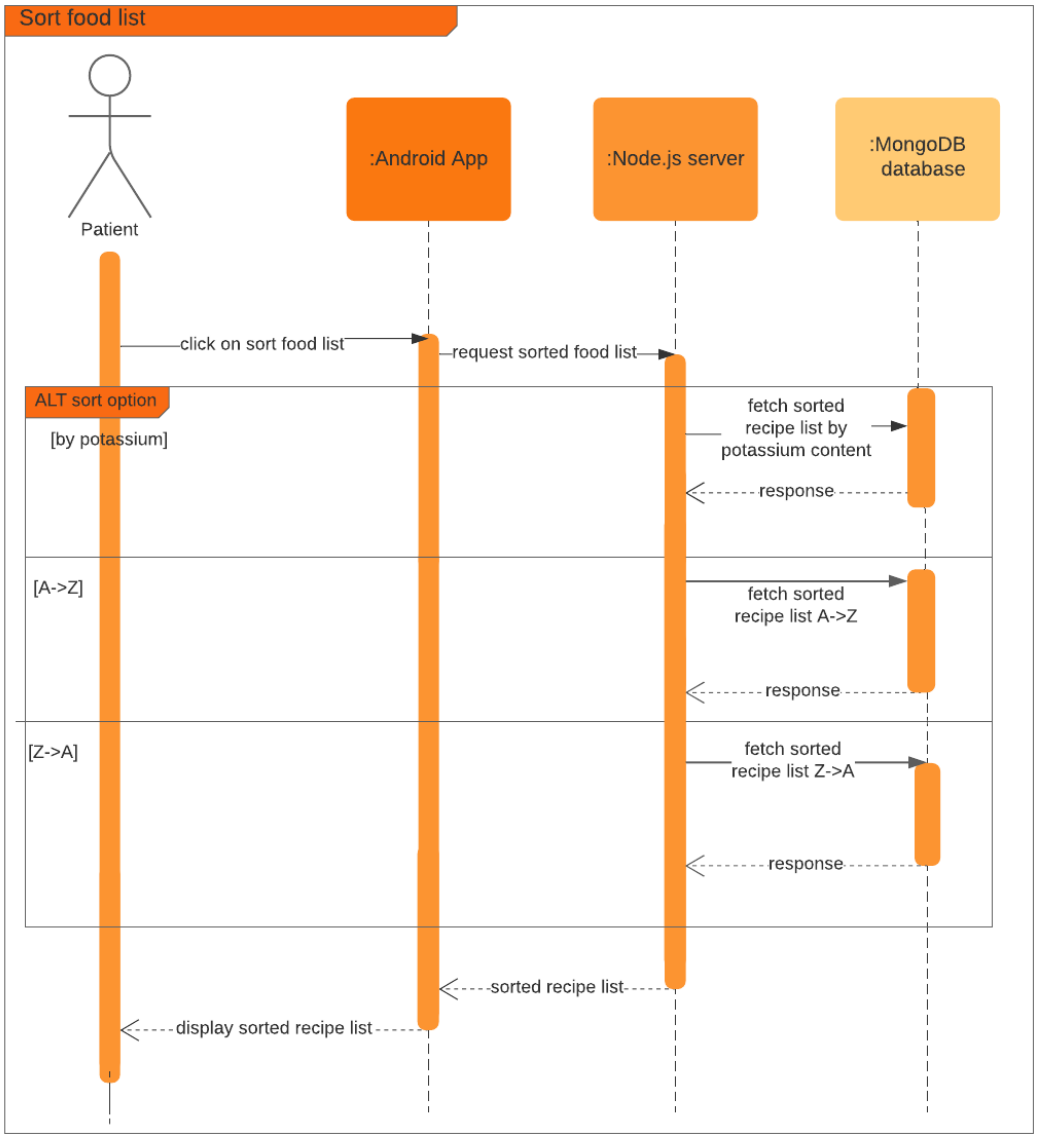


Figure 3.6: Sequence diagram : Sort food list

3.3.4 Sequence diagram : Add custom recipe

The patient can add custom recipe.

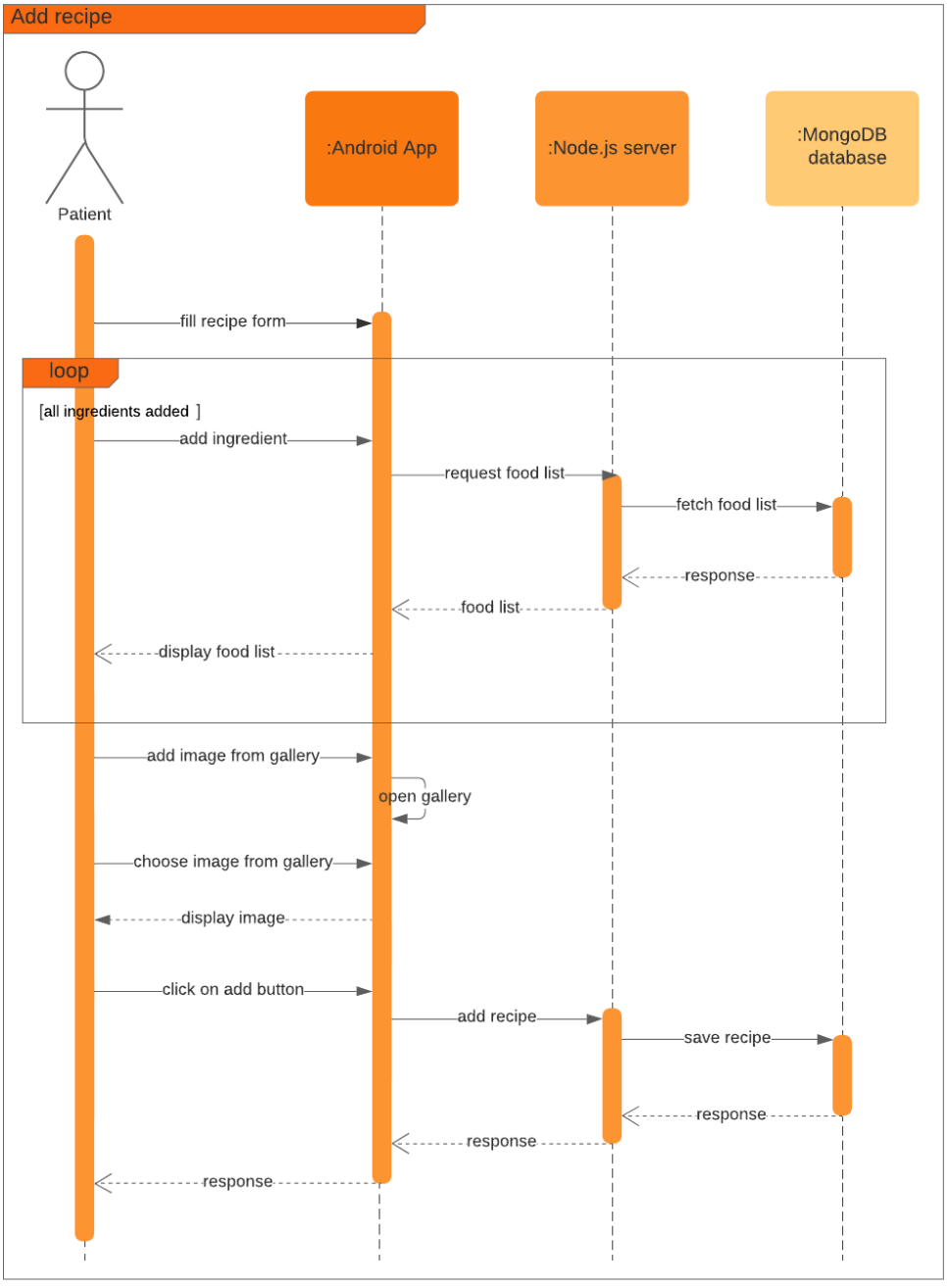


Figure 3.7: Sequence diagram : Add custom recipe

After we've outlined the key use cases, we need to look at the logical connections between them.

3.3.5 Activity diagram

The activity diagram is a UML behavioral diagram that is used to portray a workflow inside a use case or across several use cases, as well as to define the logic of an operation.

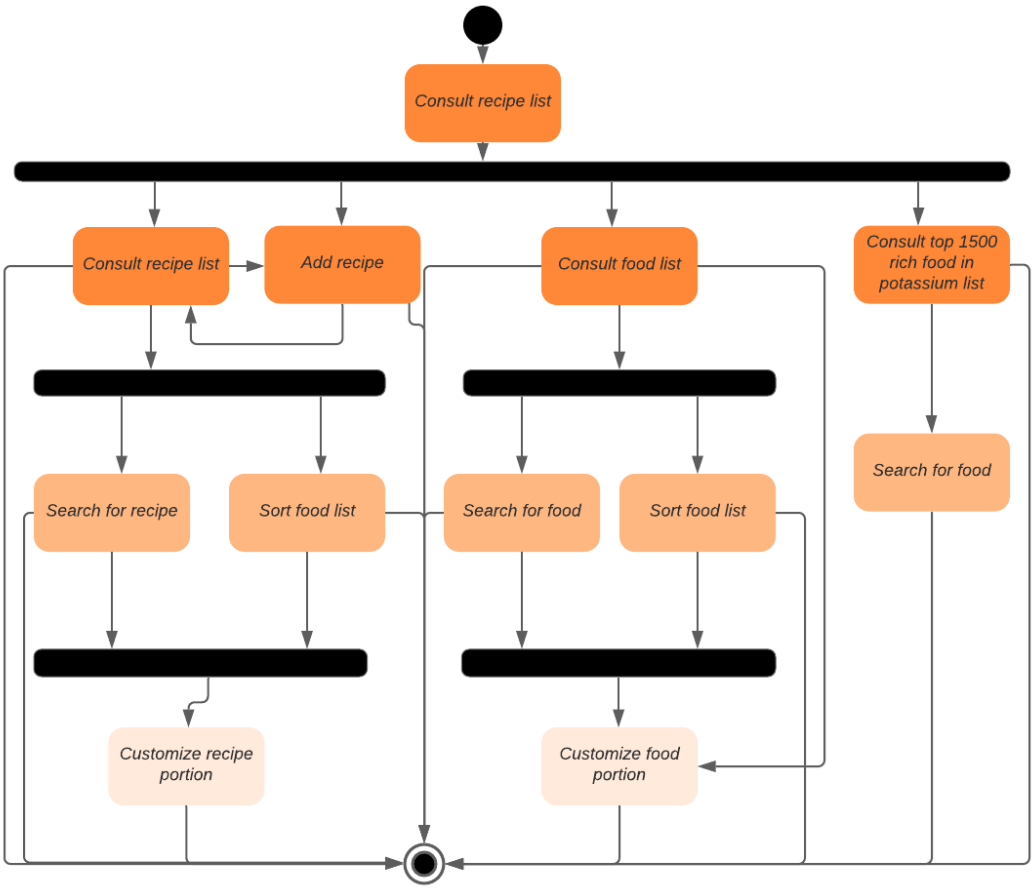


Figure 3.8: Sprint 1 Activity diagram

The user begins by checking his home page's popular recipe list, after which he can do anything, even completing the sequence without engaging in any other action.

3.4 Activities & Fragments

In this section We'll go over the activities and fragments included in this sprint.

3.4.1 Activities

The Activity class is an important aspect of an Android app, and how activities are launched and assembled is an important part of the platform's application paradigm. Unlike traditional programming paradigms, where programs are started by calling the main() function, the Android system starts code in an Activity instance by calling particular callback methods that correspond to different stages of its lifecycle.

Activity lifecycle :

The activity lifecycle is the collection of states that an activity can be in during the course of its existence, from the moment it is formed to the moment it is terminated and the system reclaims its resources. Activities change states when the user interacts with your app and other apps on the device.

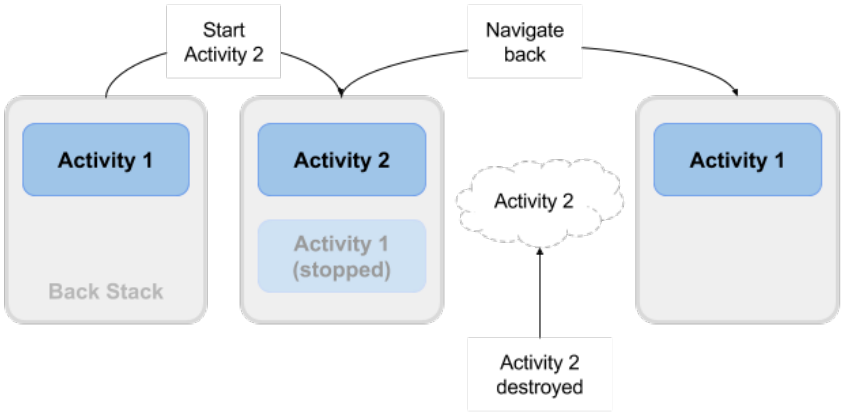


Figure 3.9: Activity lifecycle

We utilized the following activities in Sprint 1:

- **LoginActivity**: This is the system's initial activity, and it's used to authenticate the user before allowing him to access the MainActivity (for patients) , or redirect him to DoctorActivity (for doctors).
- **MainActivity**: This is the project's core activity, and it covers nearly all of the app's services.
- **DoctorActivity**: This activity encompasses almost all of the services given to the doctor actor.

3.4.2 Fragments

A fragment is a component of an activity. In an activity, there might be several fragments. Multiple screens are represented as fragments within a single activity. Because fragments are included in activity, the lifespan of an Android fragment is influenced by the lifecycle of an activity.

Fragment lifecycle :

The use of a Fragment lifetime is quite similar to the use of an Activity lifecycle. You may specify how your Fragment behaves when it is in a specific state, such as active, paused, or stopped, using the Fragment lifecycle callback methods.

The Fragment is added by an Activity.The Fragment passes through three phases after being added: Active (or restarted), Paused, and ultimately Stopped.

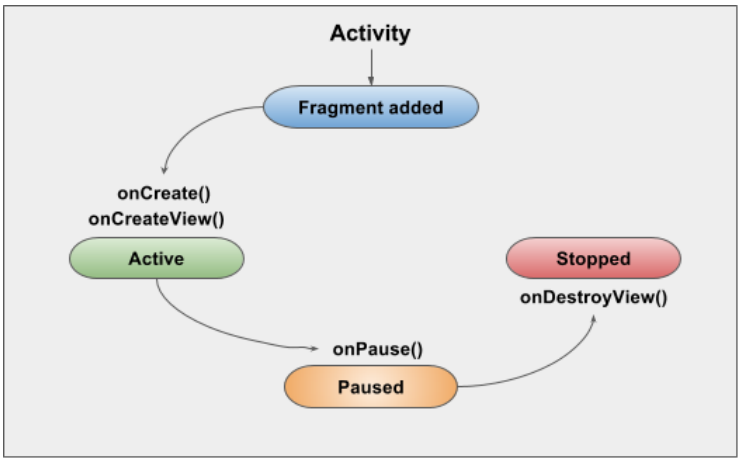


Figure 3.10: Fragment lifecycle states

We utilized the following fragments in Sprint 1:

Fragment	Service	Attached to
ListFoodFragment	Access food list / search for food / sort food list /customize food portion	Mainactivity
ListRecipeFragment	Access recipe list / search for recipe/ add custom recipe / sort food list / customize recipe portion	Mainactivity
RecipeFragment	Access recupe details / like recipe	Mainactivity

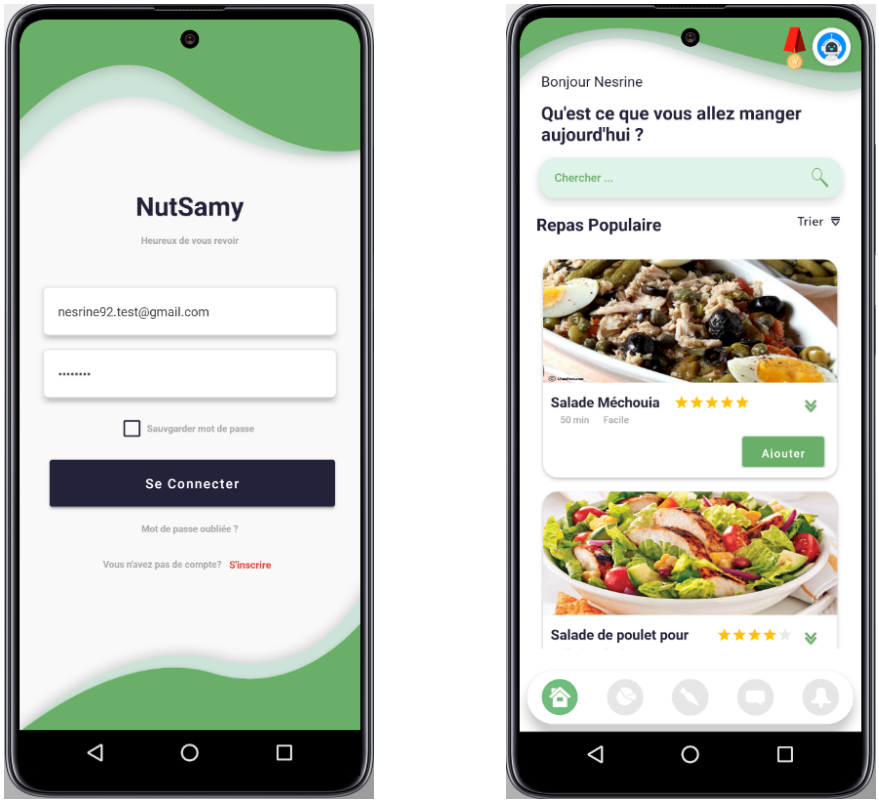
PopularRecipeFragment	Access popular recipe list / search for recipe / sort food list / customize recipe portion	Mainactivity
FoodRankFragment	Access to top 1500 rich food in potassium / search for food	Mainactivity

Table 3.2: Fragments utilized in Sprint 1

3.5 Review

The goal of the review is to illustrate what was accomplished during the sprint so that the remainder of the project may be drawn from it.We selected the most important interfaces to present in this section.

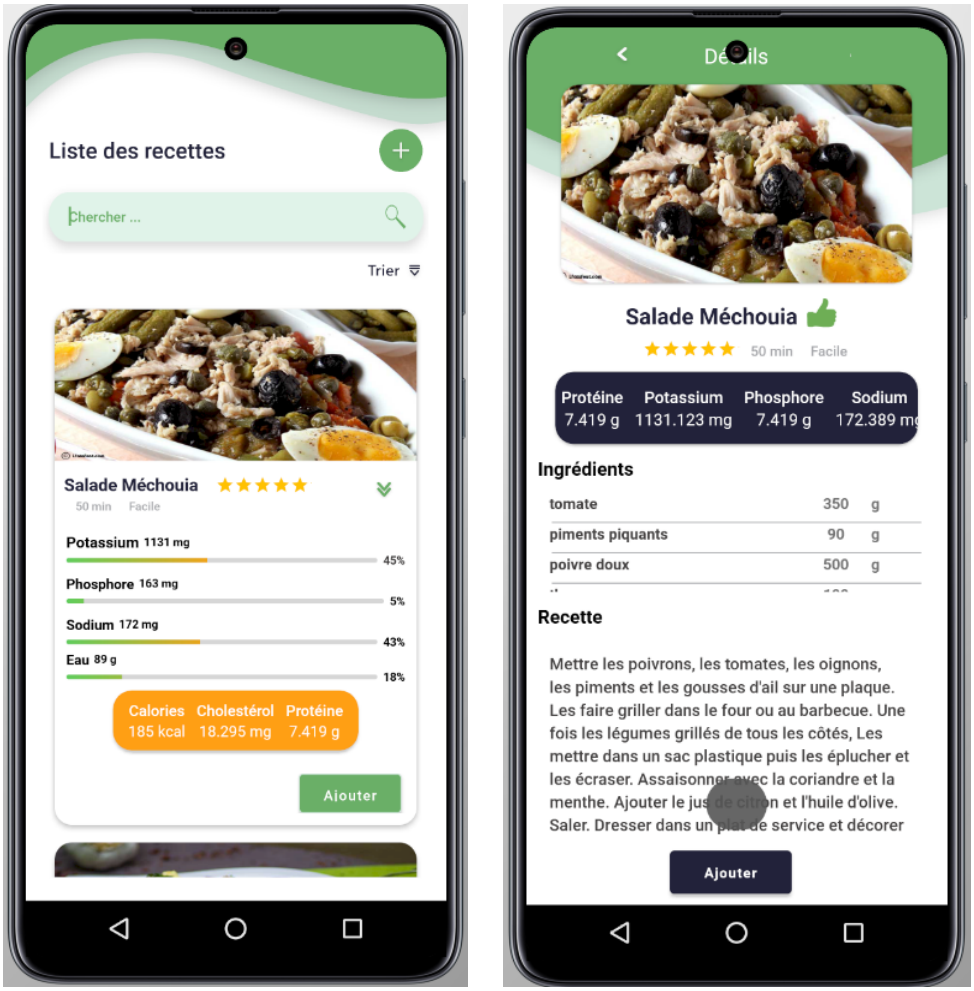
First is the login interface, which requires the user to fill out a form with his credentials in order to log in. If the user's credentials are valid he gets redirected to the home page in the figure below.



(a) Login (b) Home page

Figure 3.11: Login & home page interfaces

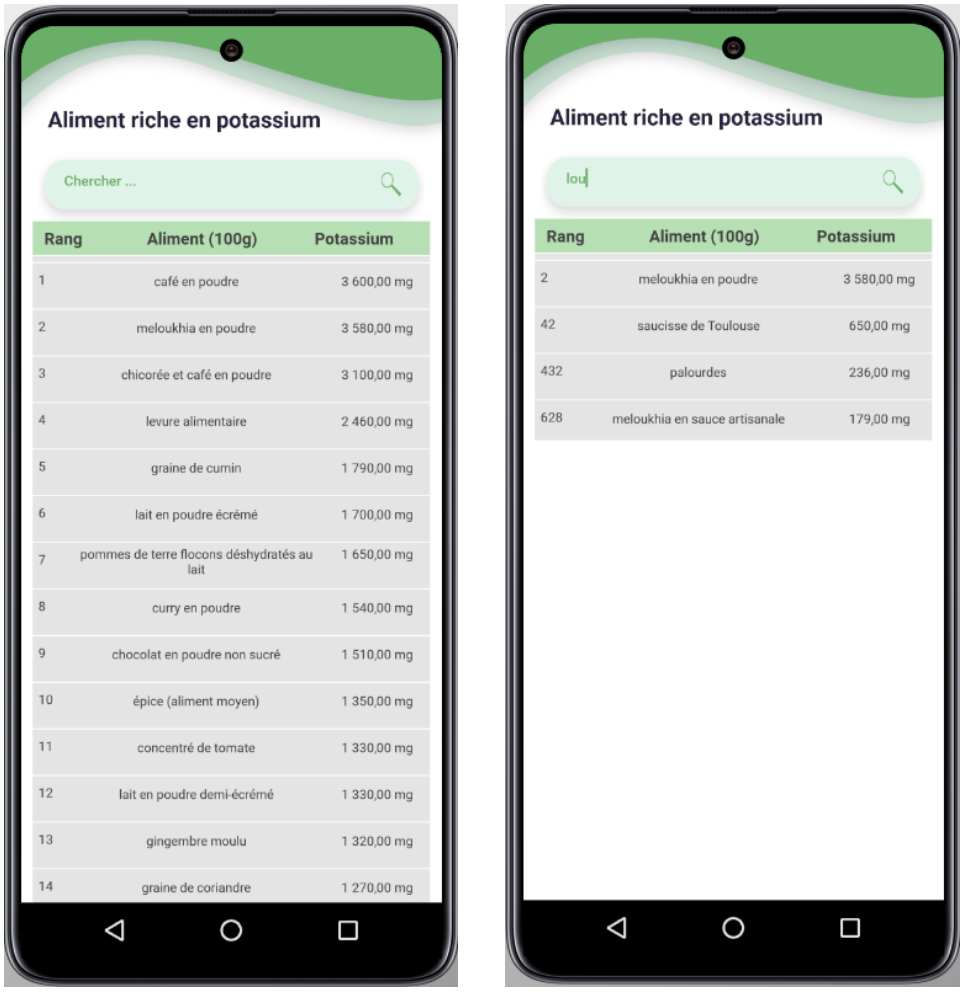
As shown in the Recipe details figure 3.12, both interfaces have all the valuable information a patient needs to know before consuming the food. The recipe card provides detailed information on the nutrient values with a simple click.



(a) Recipe card view (b) Recipe information's interfaces

Figure 3.12: Recipe details

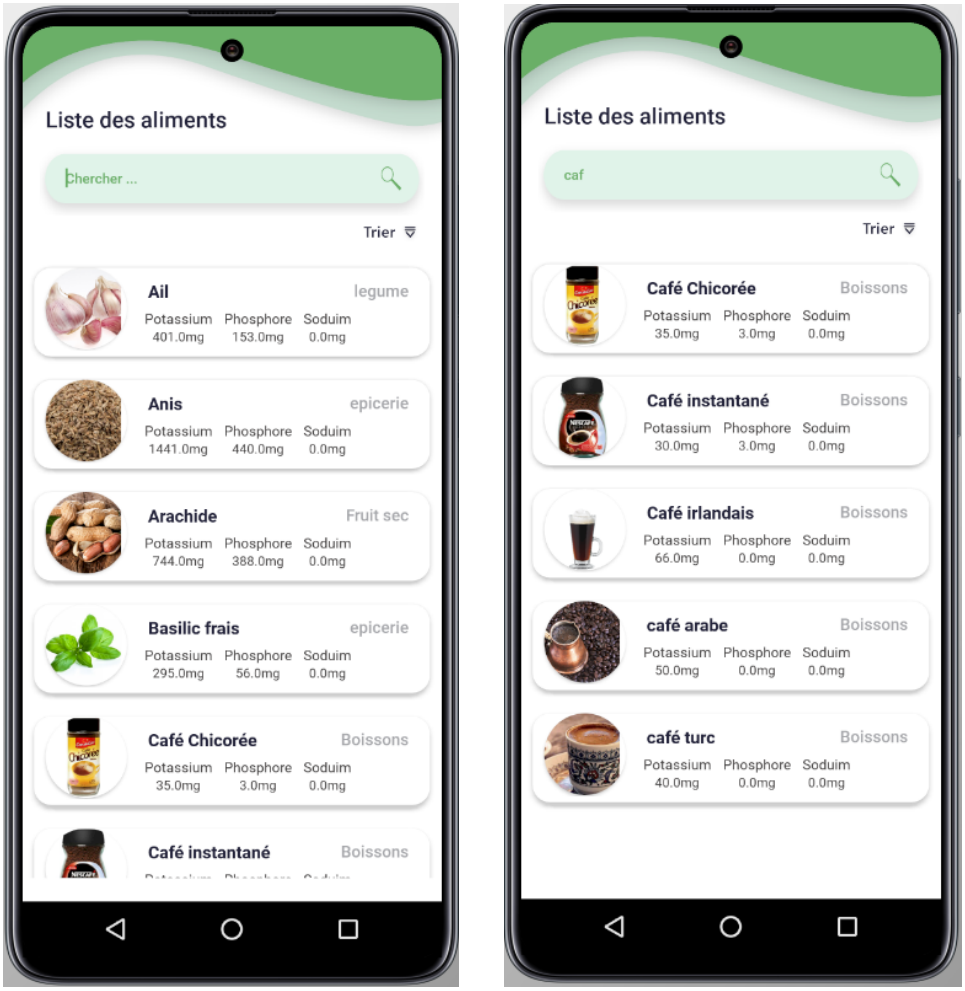
As shown in the Top 1500 rich food in potassium figure 3.13, the user can browse and search foods.



(a) Rich food in potassium list (b) Search rich food in potassium

Figure 3.13: Rich food in potassium interfaces

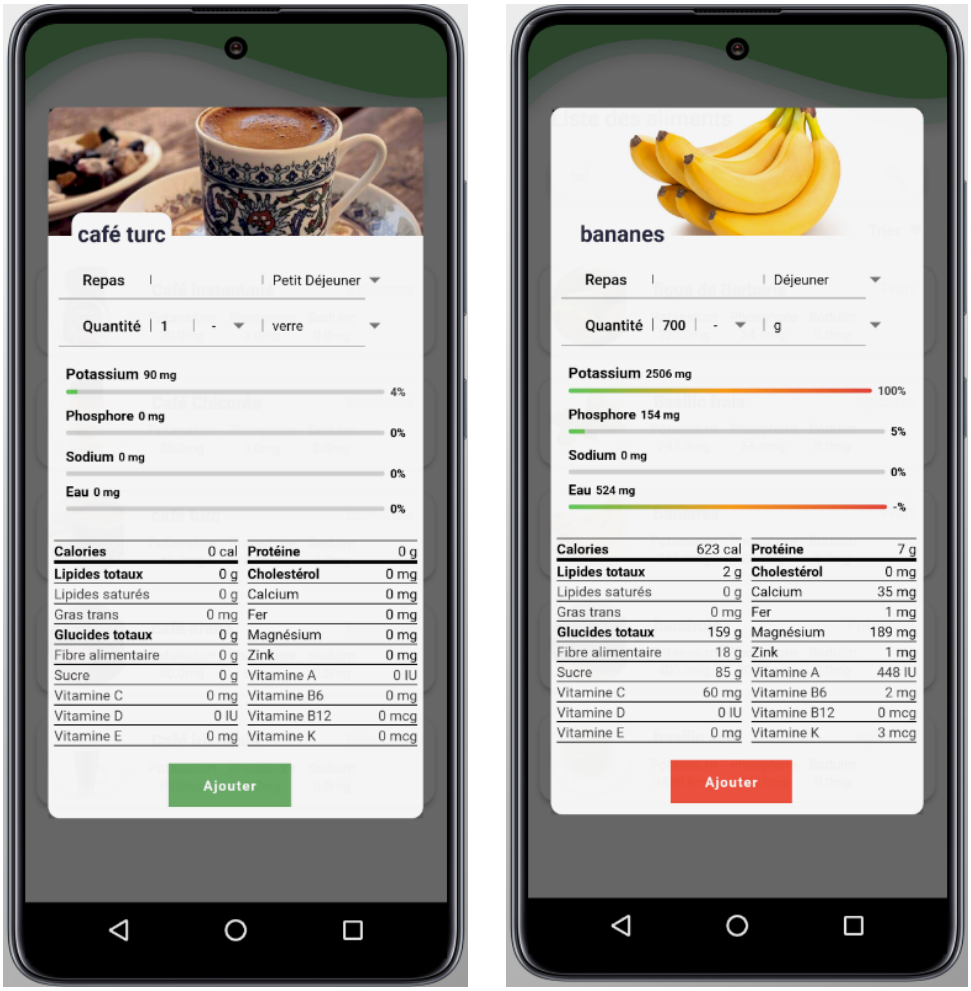
With a few swipes, the patient may browse the food list stored in our database, as well as search for a specific food by name. The outcome is dynamically displayed. He can also sort the provided results; all operations are independent of one another.



(a) Food list (b) Search for food

Figure 3.14: Food list interfaces

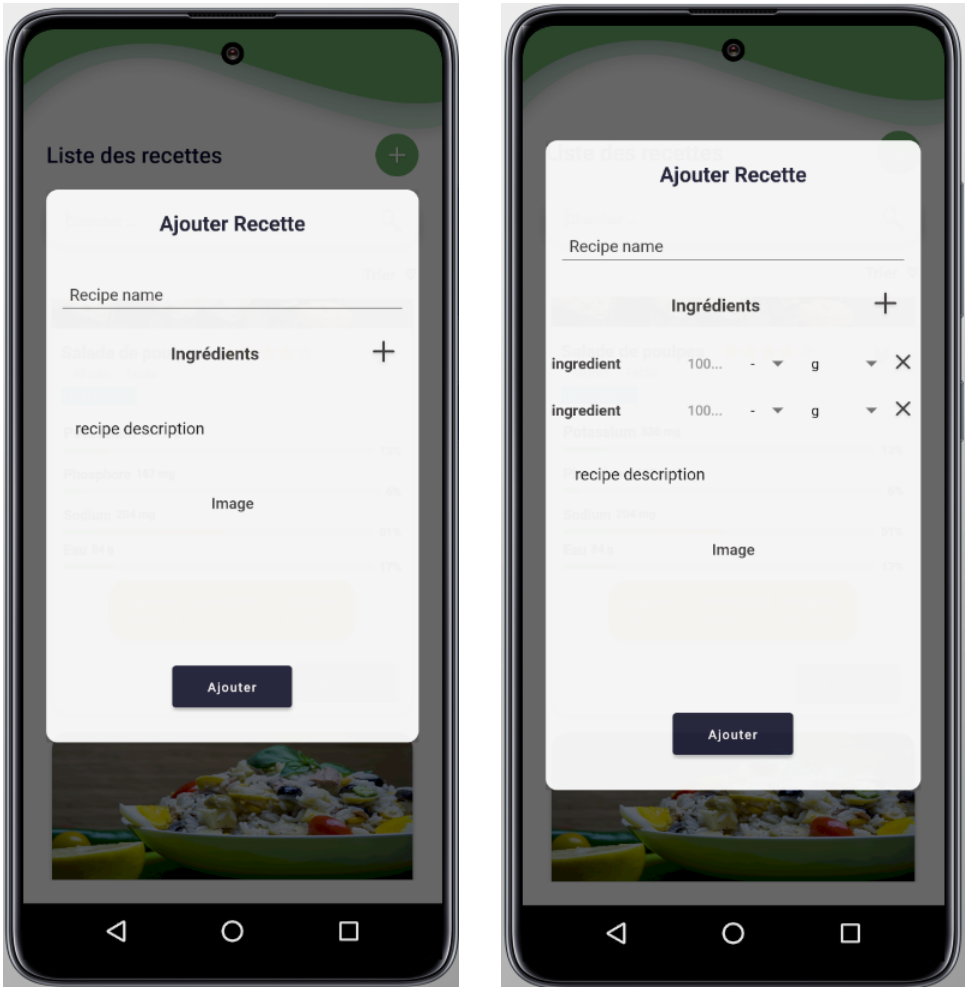
By clicking on the food or recipe card a window will pop up.In this window the user can customize the portion as he wants.First he starts by choosing the day meal (breakfast, lunch, or diner) then he can change the quantity (all the nutrient and mineral values are calculated setting the default quantity to 100 g) he has the freedom to choose the unit (g, kg, cup, bowl ....) and he s able to add Halves, Thirds and Fourths. While the user is setting the food portion the nutrient and mineral values are dynamically updating. The user is free to set the food quantity he wants but when minerals content value suppress the recommended diet values, the add button "**Ajouter**" color turns red and becomes disabled.



(a) Food cardView (b) Example of exceeded quantity

Figure 3.15: Add Food interface

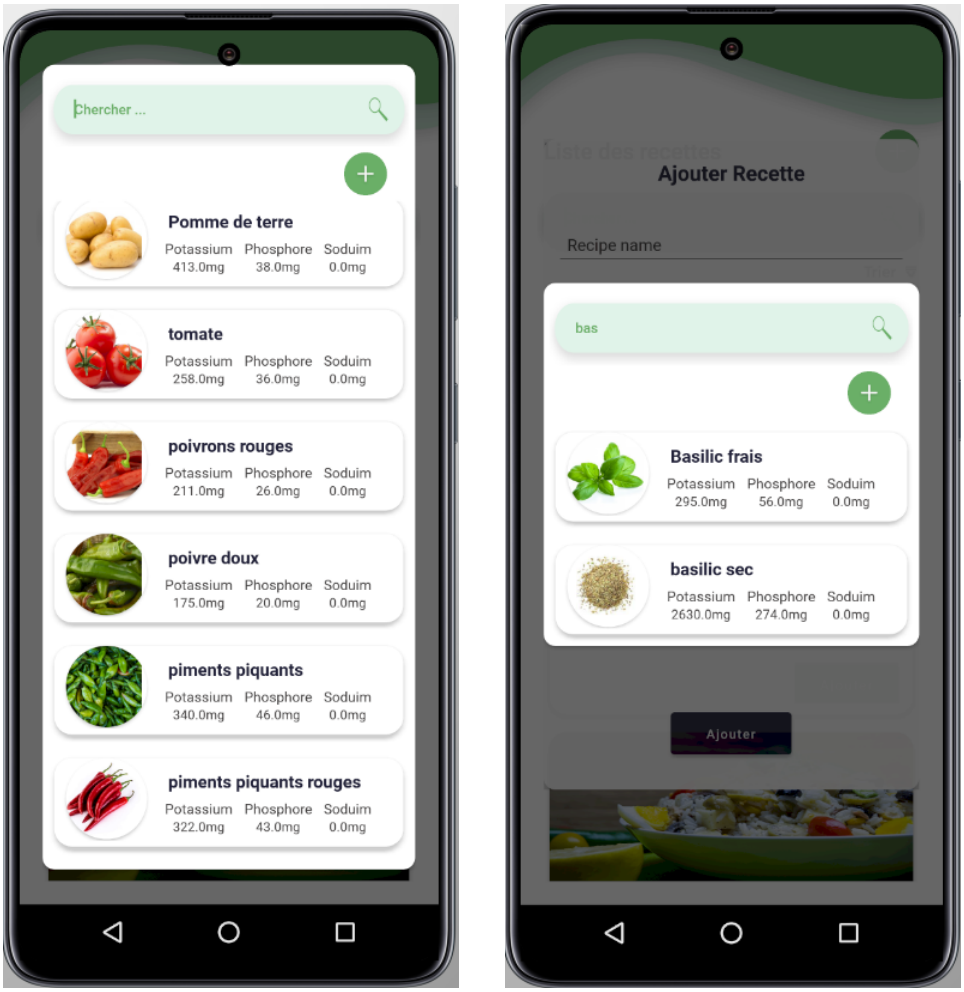
To add a recipe the user should first set the recipe name, then he can add ingredients by clicking on the plus icon on the right of "ingredients" text view a dynamic row will be added to the window that contains : ingredient , quantity & unit , and a delete button "X".



(a) Add recipe name & description (b) Add recipe ingredients

Figure 3.16: Adding custom recipe interface (1)

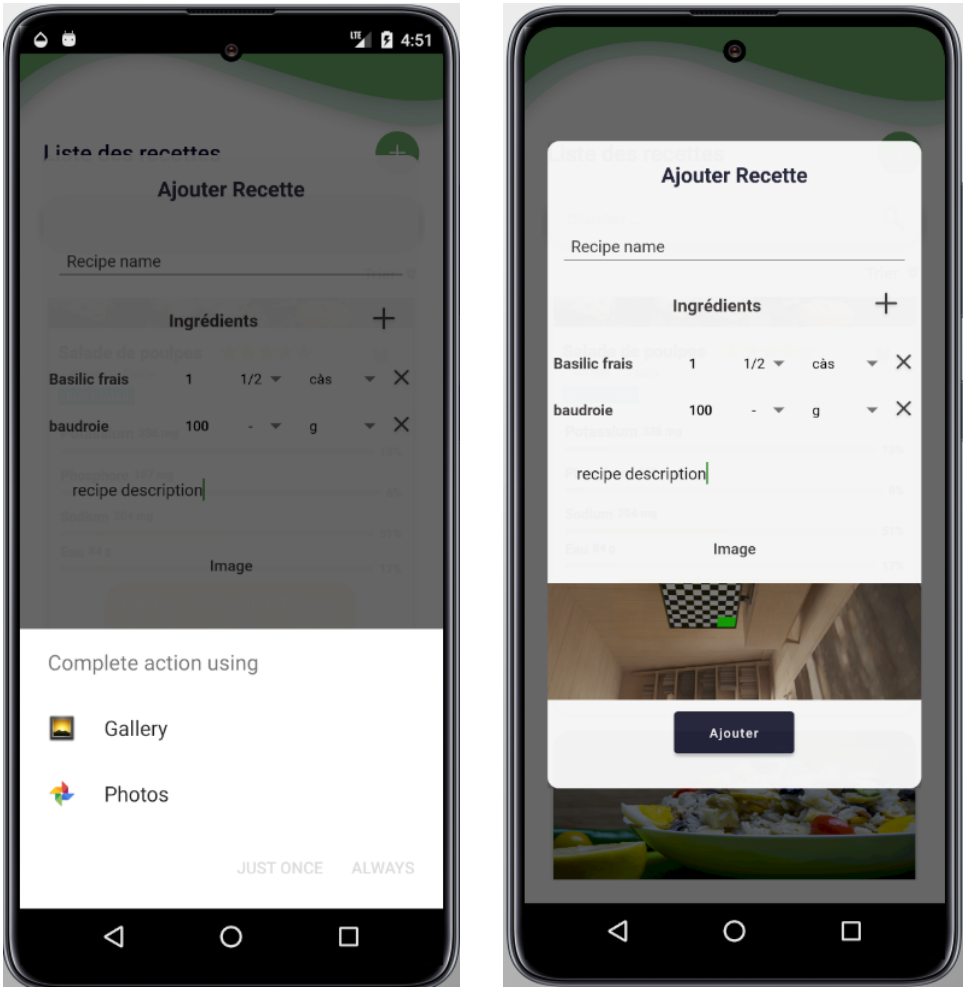
To add an ingredient the user has simply to click on the "ingredient" text view inside the rows. The list of ingredients will pop up with possibility to use the search function. To choose the ingredient the user clicks on the food card and it will be added, the name of the ingredient will be displayed instead of "ingredient" text.



(a) Choosing ingredients (b) Searching ingredients

Figure 3.17: Adding custom recipe interface (2)

Finaly the user adds a brief description and a the recipe image. The image can be added from the mobile **Gallery**.Clicking on the add button "**Ajouter**" will store the recipe in the database. Note that any saved custom recipes is available only for its creator.



(a) Add recipe image from gallery (b) Save recipe

Figure 3.18: Adding custom recipe interface (3)

To make the navigation easy for the user we added a side menu in addition to the bottom navigation bar. The side menu can be accessed by swiping the left side of any app interface to the right.

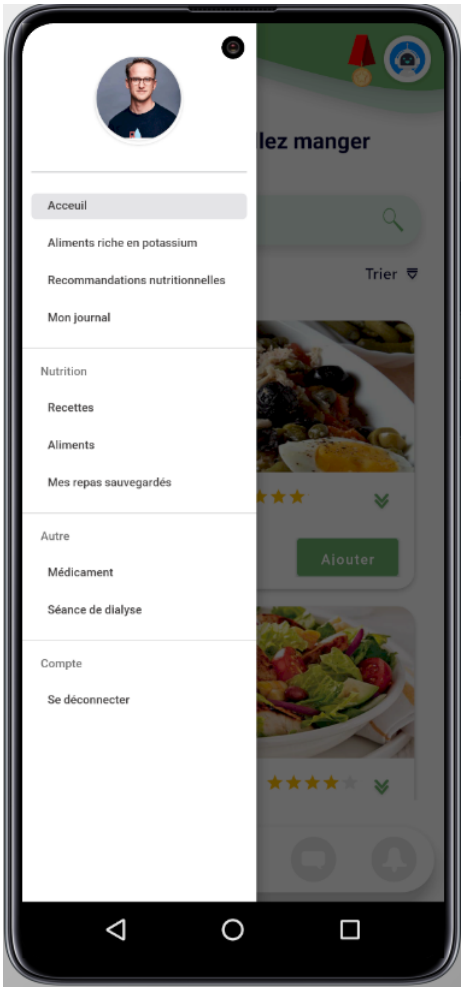


Figure 3.19: Side Menu interface

3.6 Sprint retrospective

The work progressed as expected throughout this sprint, and everything went smoothly. The figure 3.20 represents the burn down chart for sprint 1.

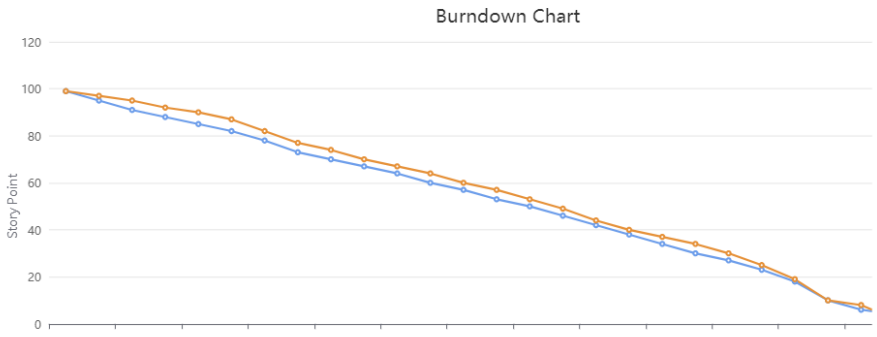


Figure 3.20: Burn down chart sprint 1

The blue curve represents the estimated burndown , while the orange one represent the actual burndown.

Conclusion

In this sprint, we started with a functional definition that included use case diagrams and system sequence diagrams. A front end and back end model, as well as a class diagram, were used to better define the concept. Finally, we exhibited off a variety of app interfaces.

CHAPTER

FOUR

SPRINT 2 : LOG MANAGEMENT

Introduction

After establishing the aim of our second sprint, which is "**Log management**", we'll proceed in the same manner as the preceding sprint. During this sprint patients will be able to update a daily log that covers all information relevant to their daily food management following this sprint, and clinicians will have access to all patient logs, with the app saving up to 30 days' worth of data.

4.1 Sprint Backlog

Sprint Backlog :

ID	User story	Acceptance criteria	Priority	Estimation
AN-05	As a patient, I am able to have a daily log	Patient should have access to his daily log	5	5
AN-06	As a patient, I am able to add food to his daily log from the food list	Food should be added to the patient daily log	6	2
AN-07	As a patient, I am able to add recipe to his daily log from the recipe list	Recipe should be added to the patient daily log	7	2

AN-10	As a patient, I am able to delete food from his daily log	Food should not be displayed in patient daily log	10	3
AN-16	As a patient, I am able to have an informative interface about his diet and daily minerale intakes	Intrface should load successfully	16	5
AN-17	As a patient, I am able to check his water intake	Water intake should be displayed as cups on the patient interface	17	3
AN-18	As a patient, I am able to update his water intake by clicking on water cups	Water cups should change state onclick empty/full state	18	3
AN-19	As a patient, I am able to save a custom meal	Meal should be saved and displayed in custom meals interface	19	5
AN-31	As a patient, I am able to have access to his monthly log	Monthly log should be displayed successfully	31	2
AN-32	As a patient, I am able to acces every day meals through the monthly log	Meals should be displayed in the correct date	32	5
AN-33	As a Doctor, I am able to login as a doctor	Doctor should be able to login successfuly not as a patint but as a doctor	33	1
AN-34	As a Doctor, I am able to have the list of all registred patient	Patient List should be displayed	34	2
AN-35	As a Doctor, I am able to for a registred patient	Search result must be correct	35	1
AN-36	As a Doctor, I am able to acces patient monthly log	Monthly log should be displayed successfully	36	2
AN-37	As a Doctor, I am able to acces patient daily log through monthly log	Daily log should be displayed successfully	37	3

Table 4.1: Log Management Sprint Backlog Sorted by priority

4.2 Use case diagram

The use case diagram for the first sprint is shown in the graphic below.The functions that the user will be able to use after this sprint are depicted in the log management use case.

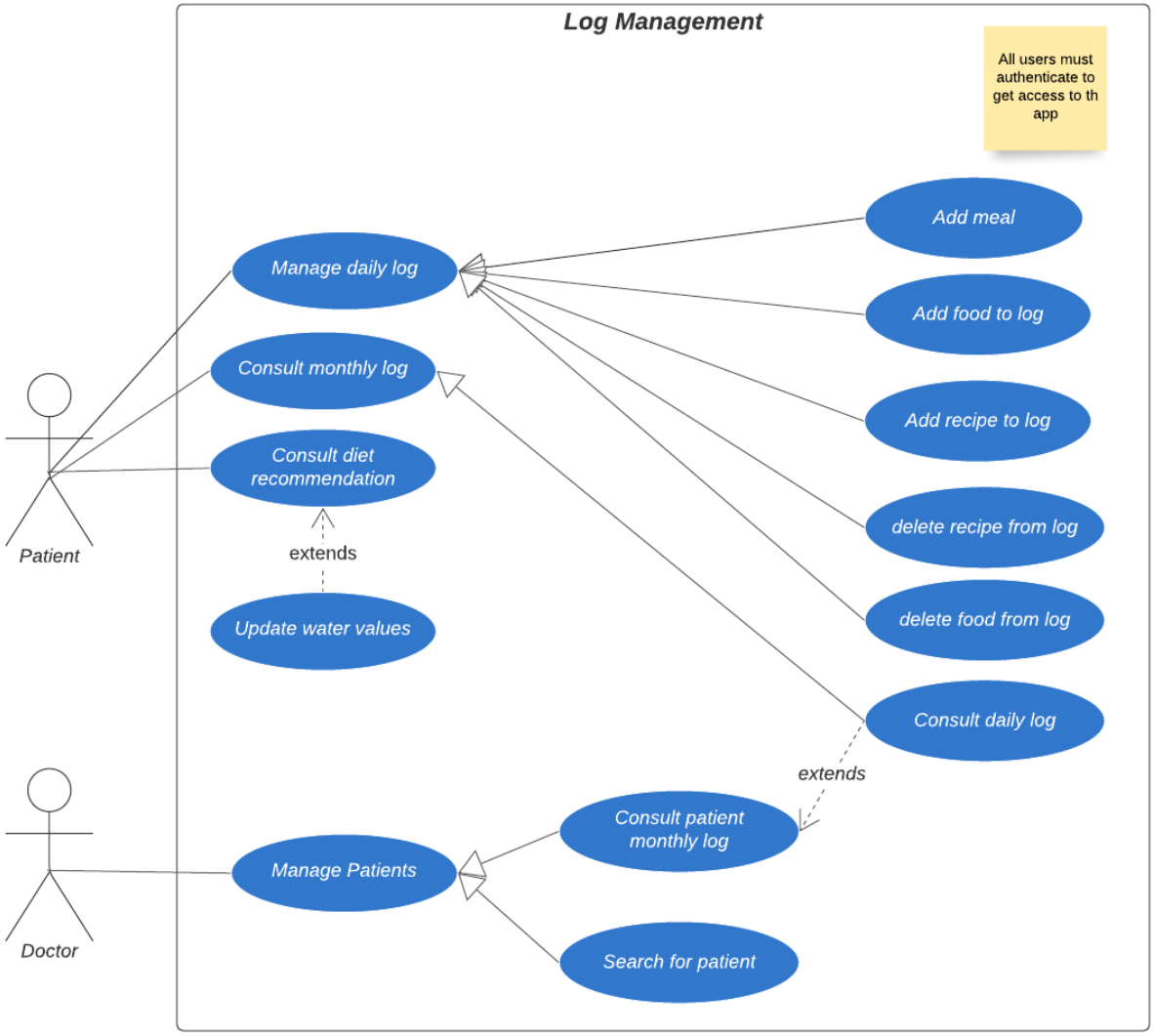


Figure 4.1: Log Management Use Case

The system sequence diagram will be utilized to give each use case a clearer and more dynamic representation.

4.2.1 Sequence diagrams

The system sequence diagrams of the most significant defined use cases are presented.

4.2.2 Sequence diagram : Add food to daily log

The patient can add food or recipe to his daily log.

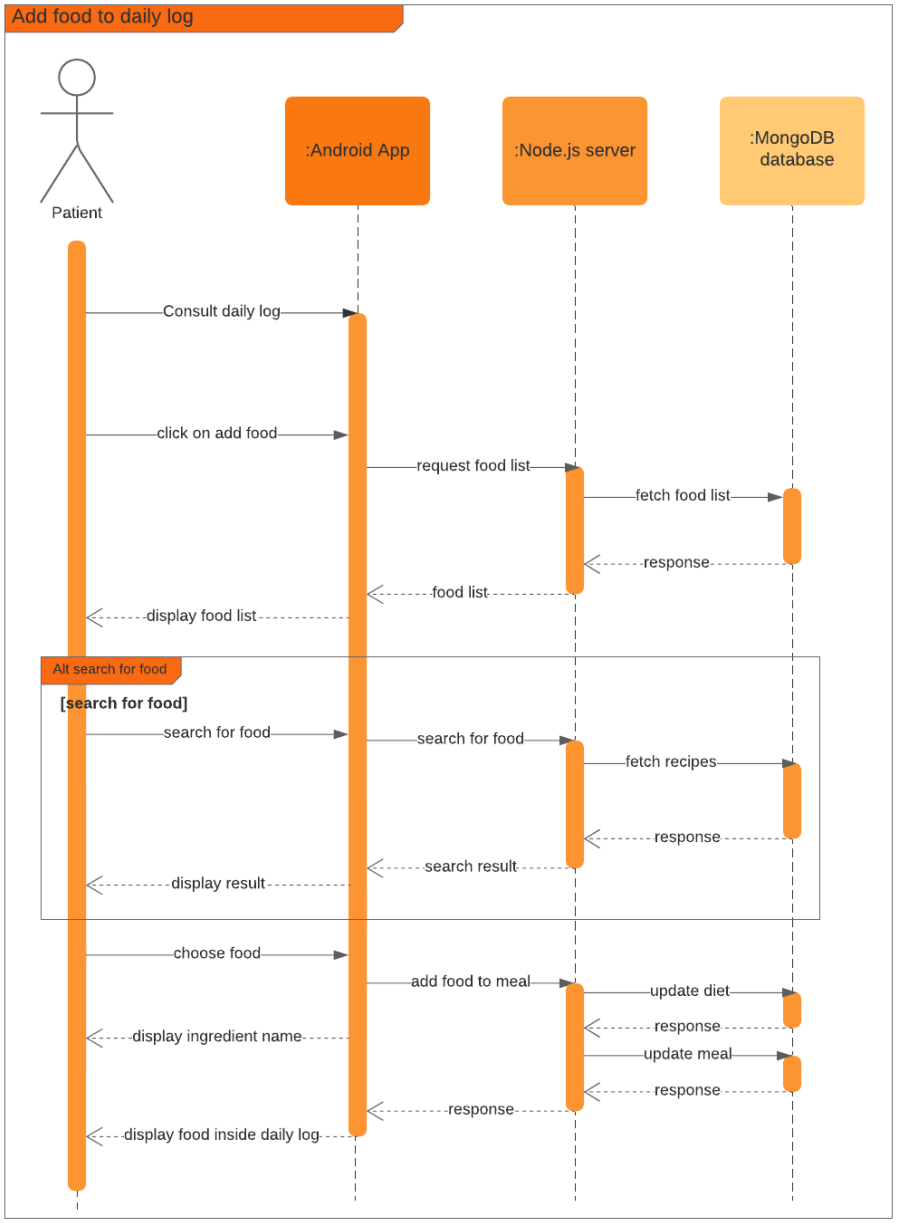


Figure 4.2: Add food to daily log sequence diagram

4.2.3 Sequence diagram : Save a meal

The patient can save a meal from his daily log.

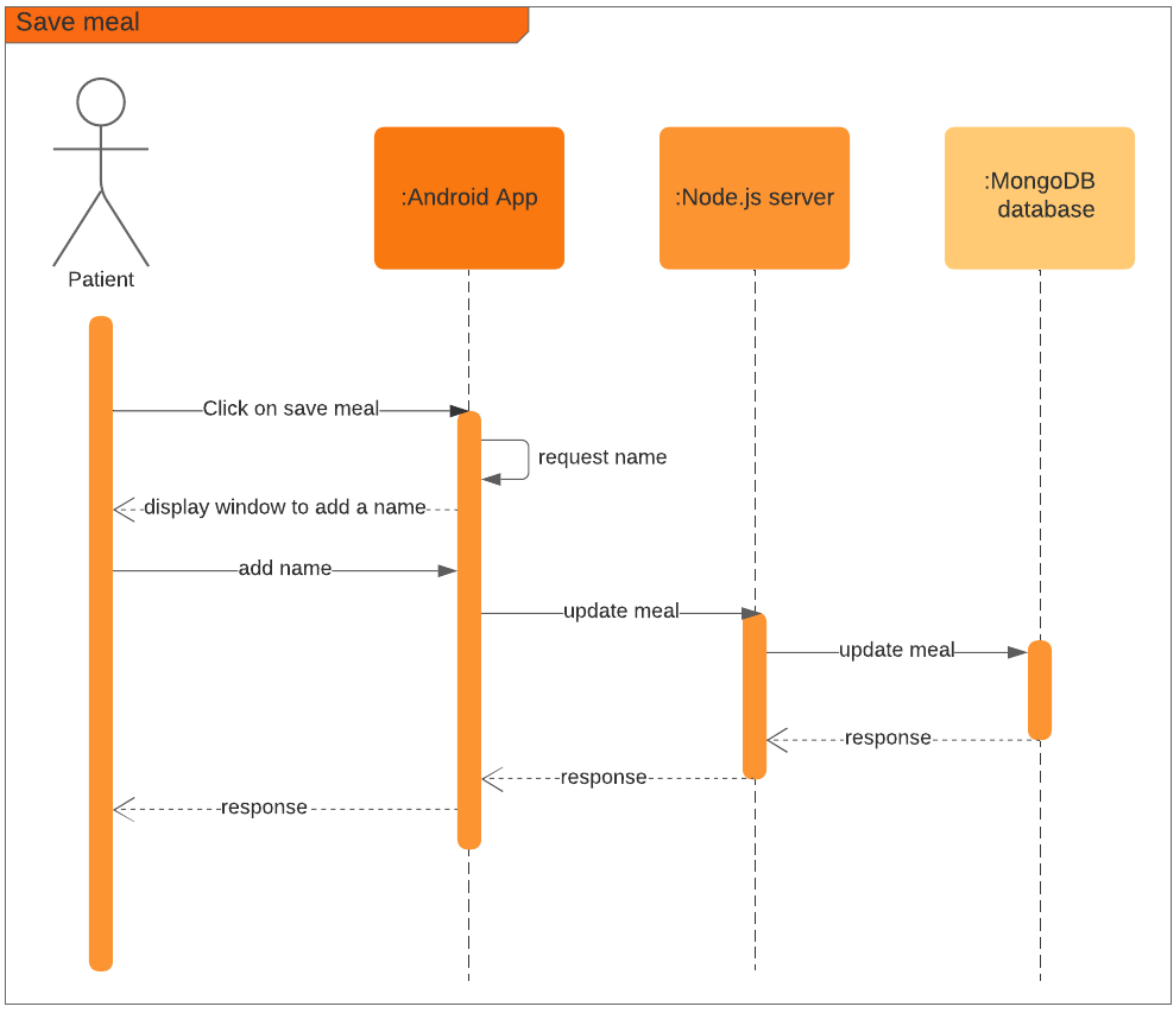


Figure 4.3: Save a meal from daily log sequence diagram

4.2.4 Sequence diagram : Consult patient log

The doctor can consult any patient logs.

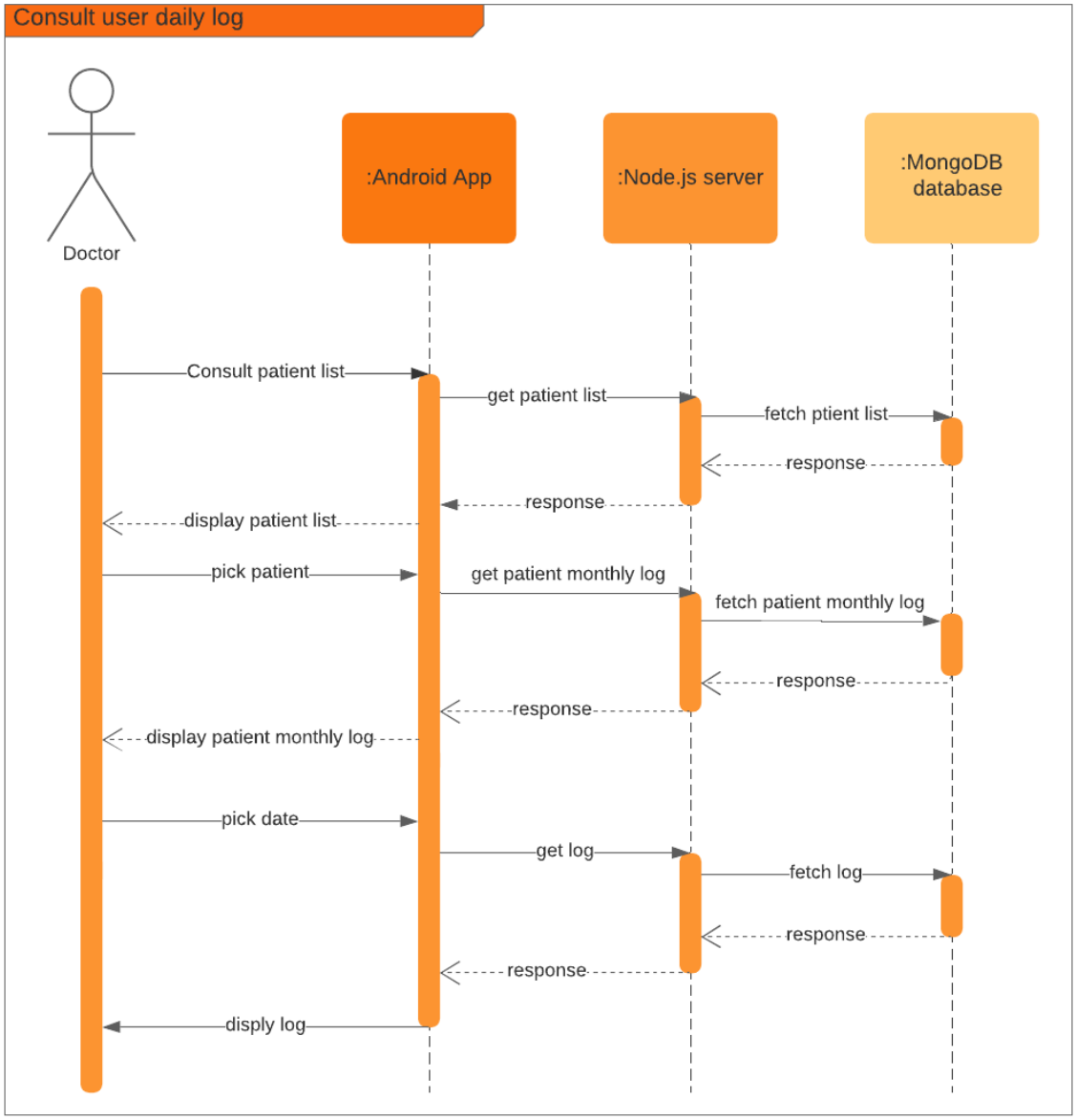


Figure 4.4: Consult patient log sequence diagram

4.3 Activity diagram

The diagram below describes the possible action the app will provide to the patient actor in this sprint.

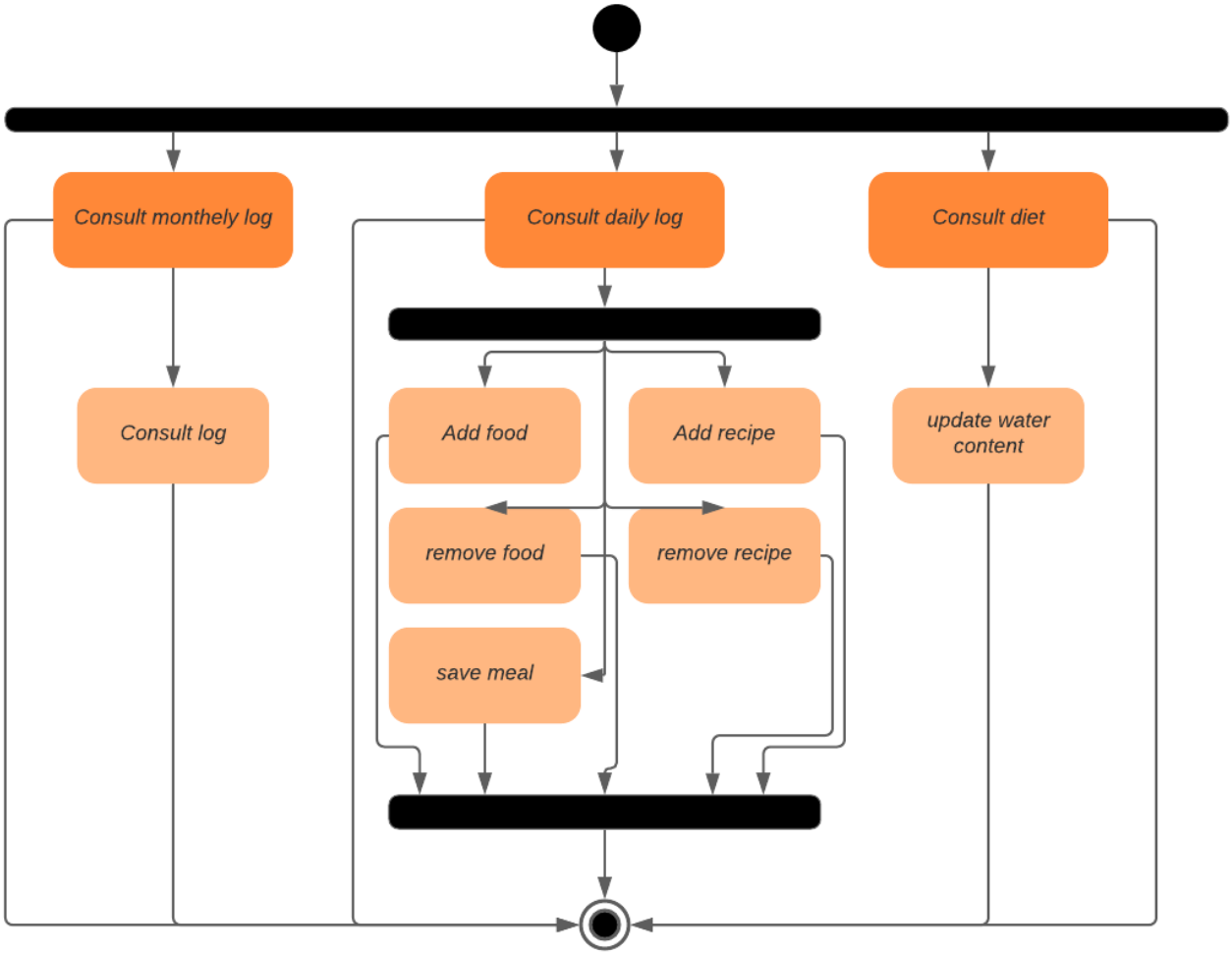


Figure 4.5: Sprint 2 diagram activity for patient actor

The diagram below describes the possible action the app will provide to the doctor actor in this sprint.

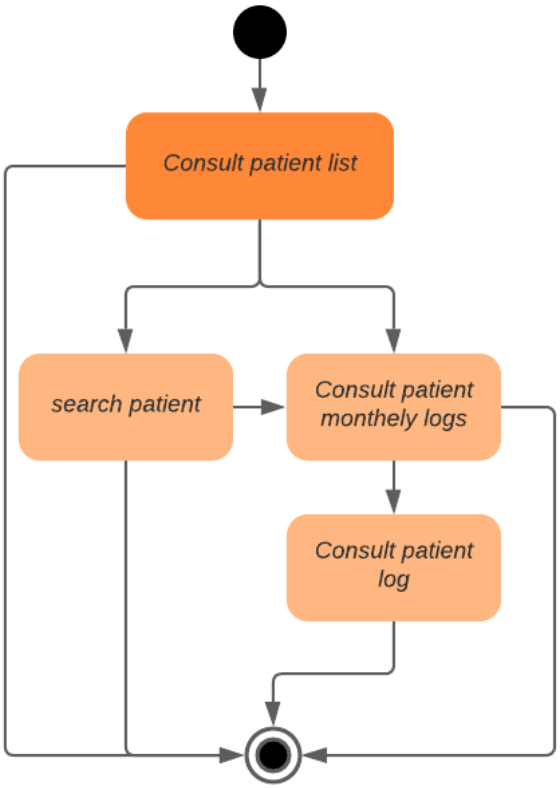


Figure 4.6: Sprint 2 diagram activity for doctor actor

4.4 Activities & Fragments

In this section we'll go over the activities and fragments included in this sprint.

4.4.1 Activities

We utilized the following activities in Sprint 2:

- MainActivity: This is the project's core activity, and it covers nearly all of the app's services.
- DoctorActivity: This activity encompasses almost all of the services given to the doctor actor

4.4.2 Fragments

We utilized the following fragments in Sprint 2:

Fragment	Service	Attached to
LogFragment	Manage patient log	MainActivity
MonthlyLogFragment	Consult monthly log	MainActivity
ListSavedMealsFragment	Consult custom meals	MainActivity
RecNutritionFragment	Consult diet / update water intake	MainActivity
PatientListFragment	Consult patient list / search patient	DoctorActivity
PatientMonthlyLogFragment	Consult patient monthly and daily log	DoctorActivity

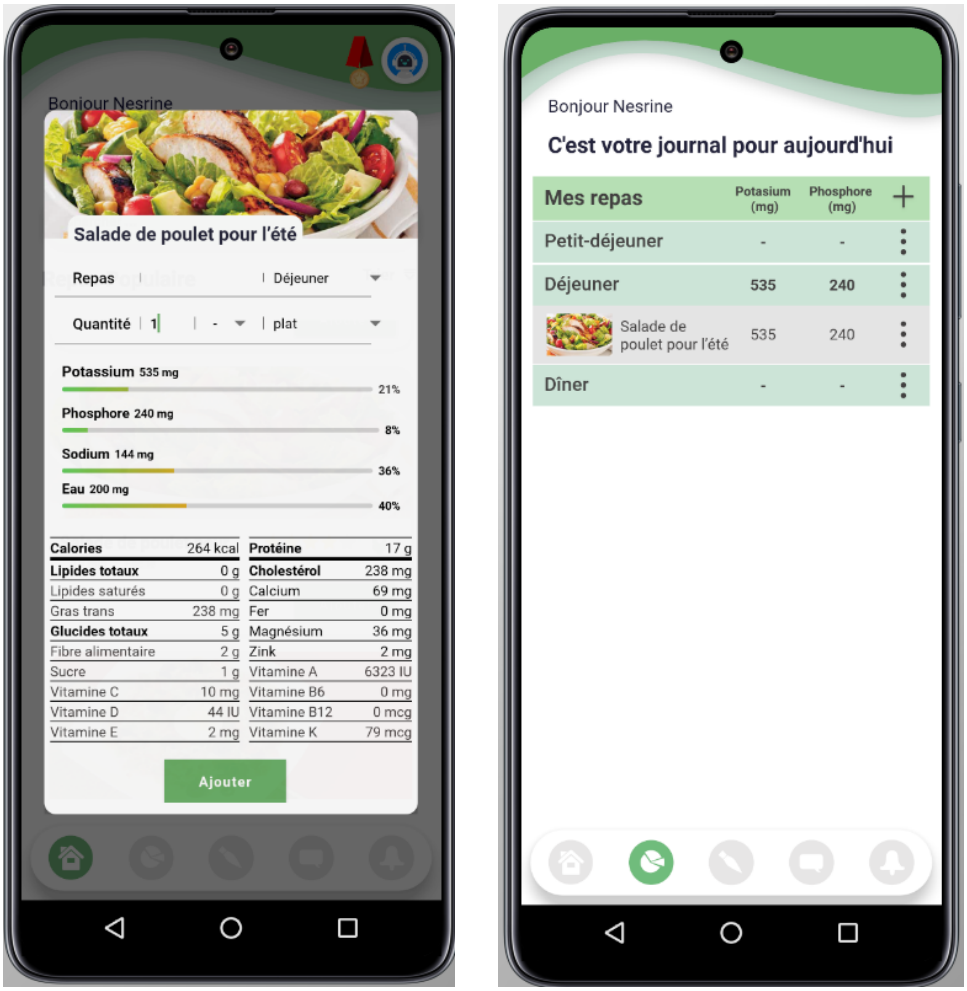
Table 4.2: Fragments utilized in Sprint 2

4.5 Review

The review's purpose is to show what was accomplished in the second sprint. In this part, we've highlighted the most essential interfaces. In the start of everyday the patient log get initialized automatically to a new one as shown below.

Figure 4.7: Empty log

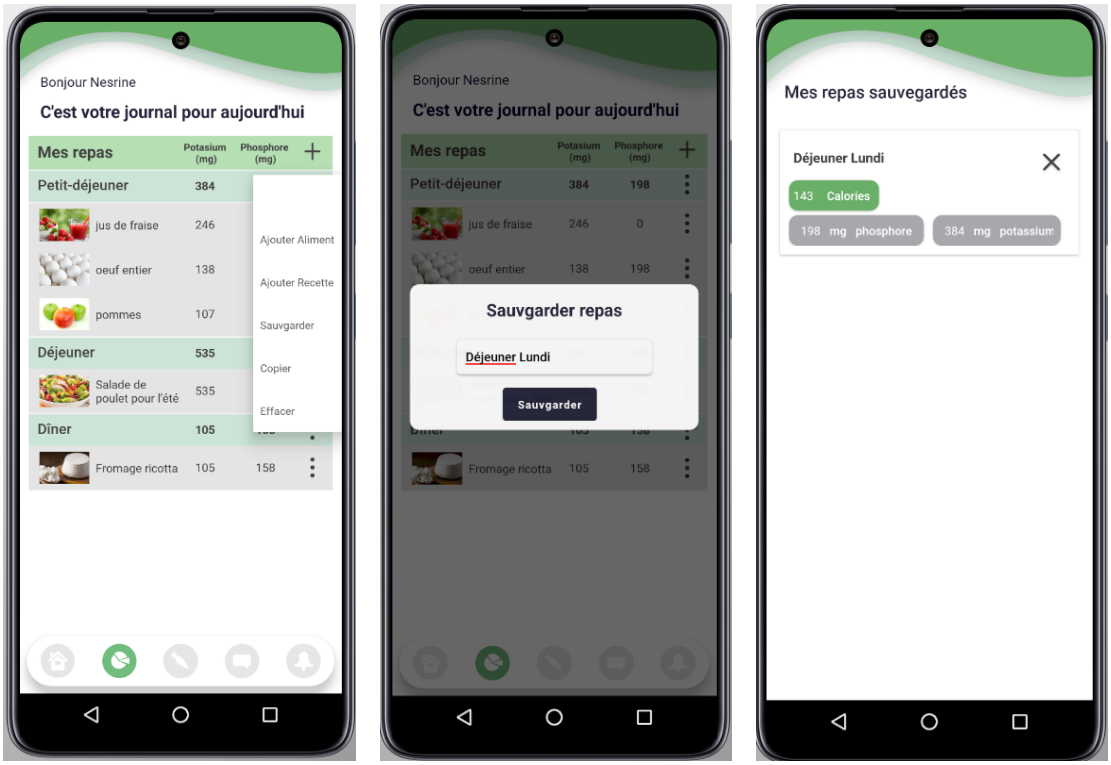
The patient can add foods or recipes to his daily log. He has to specify in which meal he is going to update and the portion. The system will estimate the quantity of minerals and nutritional values acquired and subtract them from his regular diet. When he tries to add another item, the patient's mineral intake limit will be reduced until he can no longer add anything until he deletes foods or recipes from his daily diary. The figure 4.8 is an example that illustrate adding recipe to daily log.



(a) Add recipe to log (1) (b) Add recipe to log (2)

Figure 4.8: Adding recipe to daily log

The patient has the possibility to save a custom meal. Custom meals are saved in a separate list, its purpose is to save the patient time instead of adding foods and recipes to the log he can directly add a saved meal instead.



(a) Save custom meal (b) Set custom meal name (c) custom meal list

Figure 4.9: Save custom meal

By selecting "**recommendation nutritionelles**" from the side menu, the patient may access his diet. By clicking on the water cups, he may update his water consumption. The user will be reminded to drink water until all of the cups are empty.

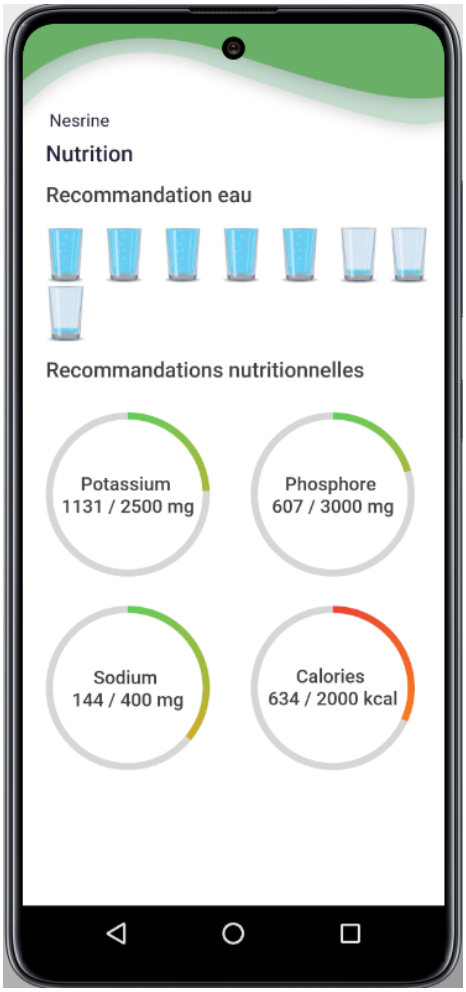
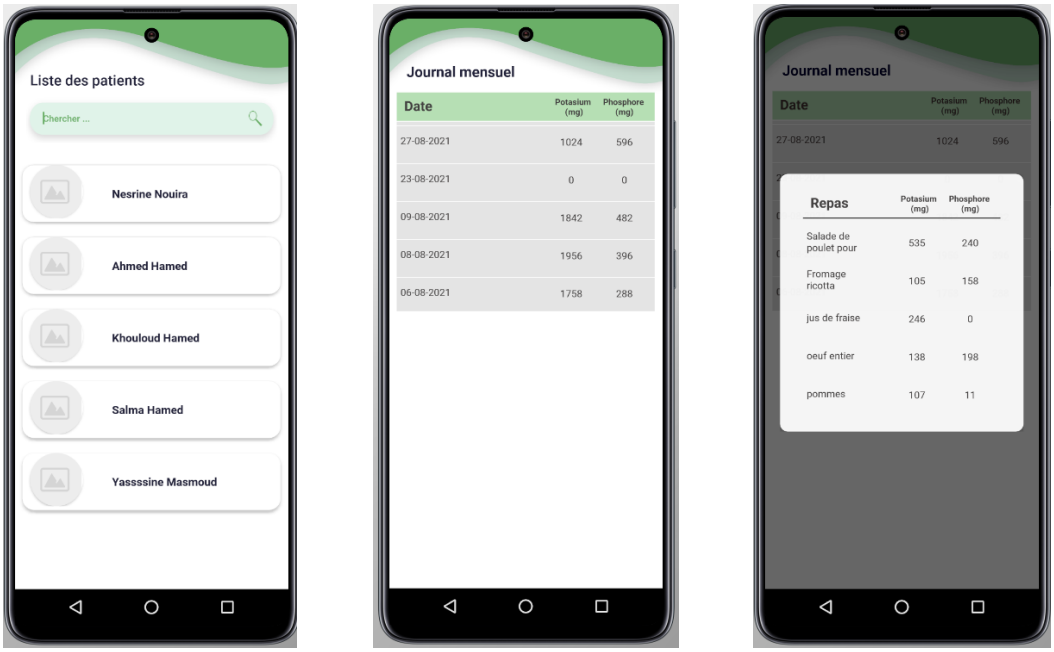


Figure 4.10: Diet interface

Doctors can browse the list of patients and consult their logs in this sprint.



(a) Patients list (b) Patient monthly log (c) Patient log

Figure 4.11: Doctors interface

4.6 Sprint retrospective

Throughout this sprint, the work moved as planned, and everything went well.

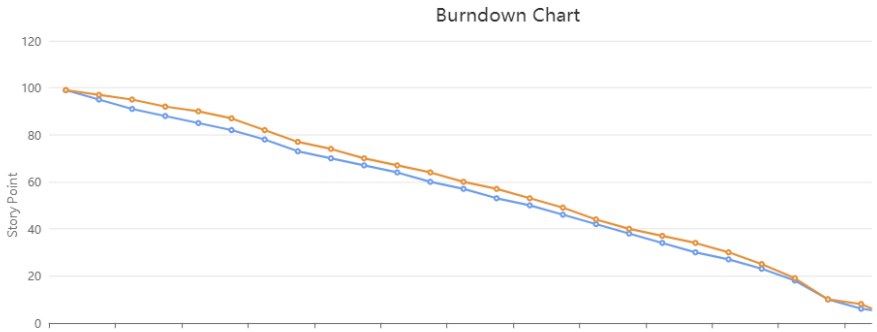


Figure 4.12: Burn down chart sprint 2

The blue curve represents the estimated burndown , while the orange one represent the actual burndown.

Conclusion

We proceeded with a broad definition, which comprised use case diagrams, system sequence diagrams, and activity diagrams in this sprint.Finally, we demonstrated several app interfaces.

CHAPTER

FIVE

SPRINT 3 : MEDICATION MANAGEMENT

Introduction

The goal of our third sprint, "**Medication management**," is to aid patients by giving them the ability to set up medication alerts, which will remind them when they have medicine to take, as well as a dialysis session reminder, which will help them remember their session times.

5.1 Sprint Backlog

Sprint Backlog :

ID	User story	Acceptance criteria	Priority	Estimation
AN-22	As a patient, I am able to add a medicine	Medicine should be added successfully to database	22	7
AN-23	As a patient, I am able to add a dialysis session	Dialysis session should be added successfully to database	23	3
AN-24	As a patient, I am able to see his medicine list	Medicine list should be displayed	24	1
AN-25	As a patient, I am able to see his dialysis session list	Dialysis session list should be displayed	25	1
AN-26	As a patient, I am able to delete a medicine	Medicine should be deleted successfully	26	1

AN-27	As a patient, I am able to delete dialysis session	Dialysis session should be deleted successfully	27	1
AN-28	As a patient, I am able to have a medicine reminder that sends a custom notifications	Notification should be specific and on time	28	5
AN-29	As a patient, I am able to have a dialysis session reminder that sends a custom notifications	Notification should be specific and on time	29	5
AN-30	As a patient, I am able to have a water reminder that sends a custom notifications	Notification should be specific and on time	30	5

Table 5.1: Medication Management Sprint Backlog Sorted by priority

5.2 Use case diagram

The figure 5.1 presents the use case diagram for the first sprint. The log management use case demonstrates the functions that the user will be able to utilize after this sprint.

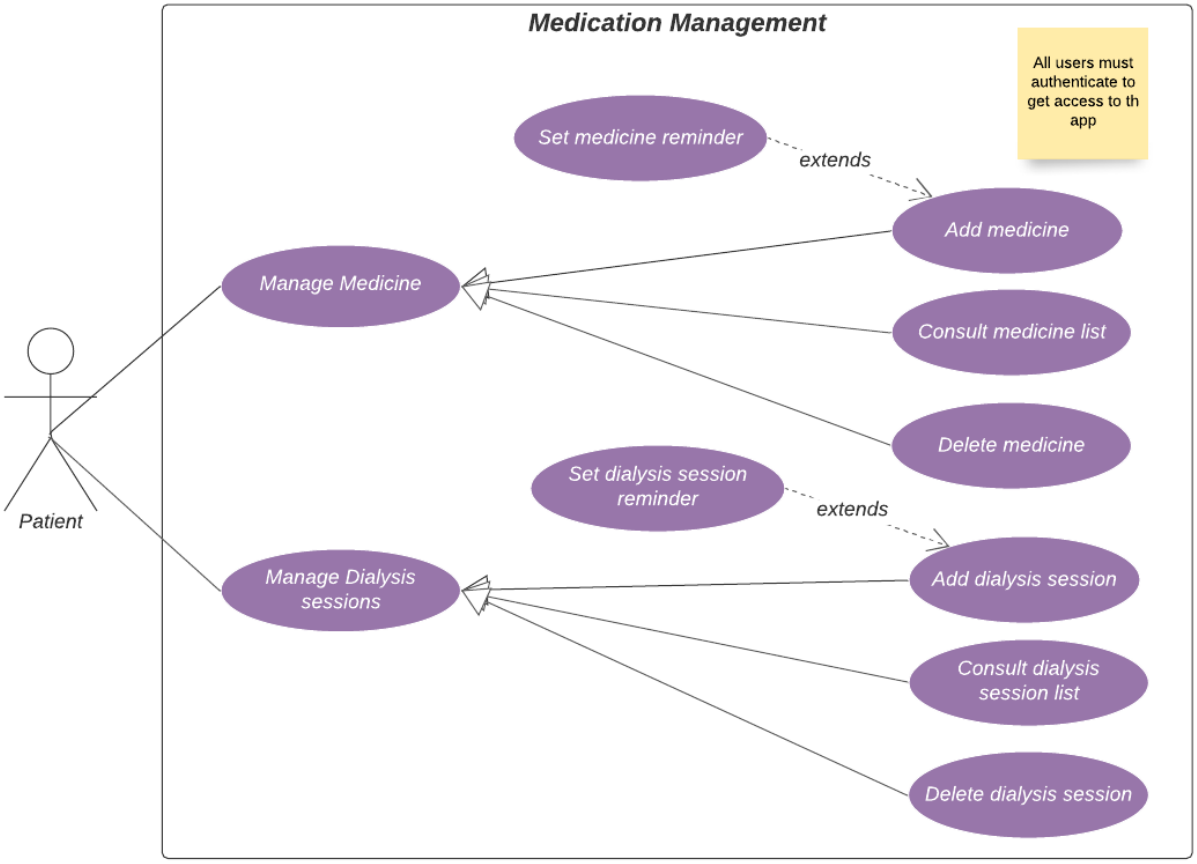


Figure 5.1: Medication Management Use Case

The system sequence diagram will be used to depict each use case in a more dynamic and transparent manner.

5.3 Sequence diagrams

The system sequence diagrams of the most significant defined use cases are presented.

5.3.1 Sequence diagrams : Add Medicine

The patient has the option of adding medicines. The case's relative sequence diagram is shown below.

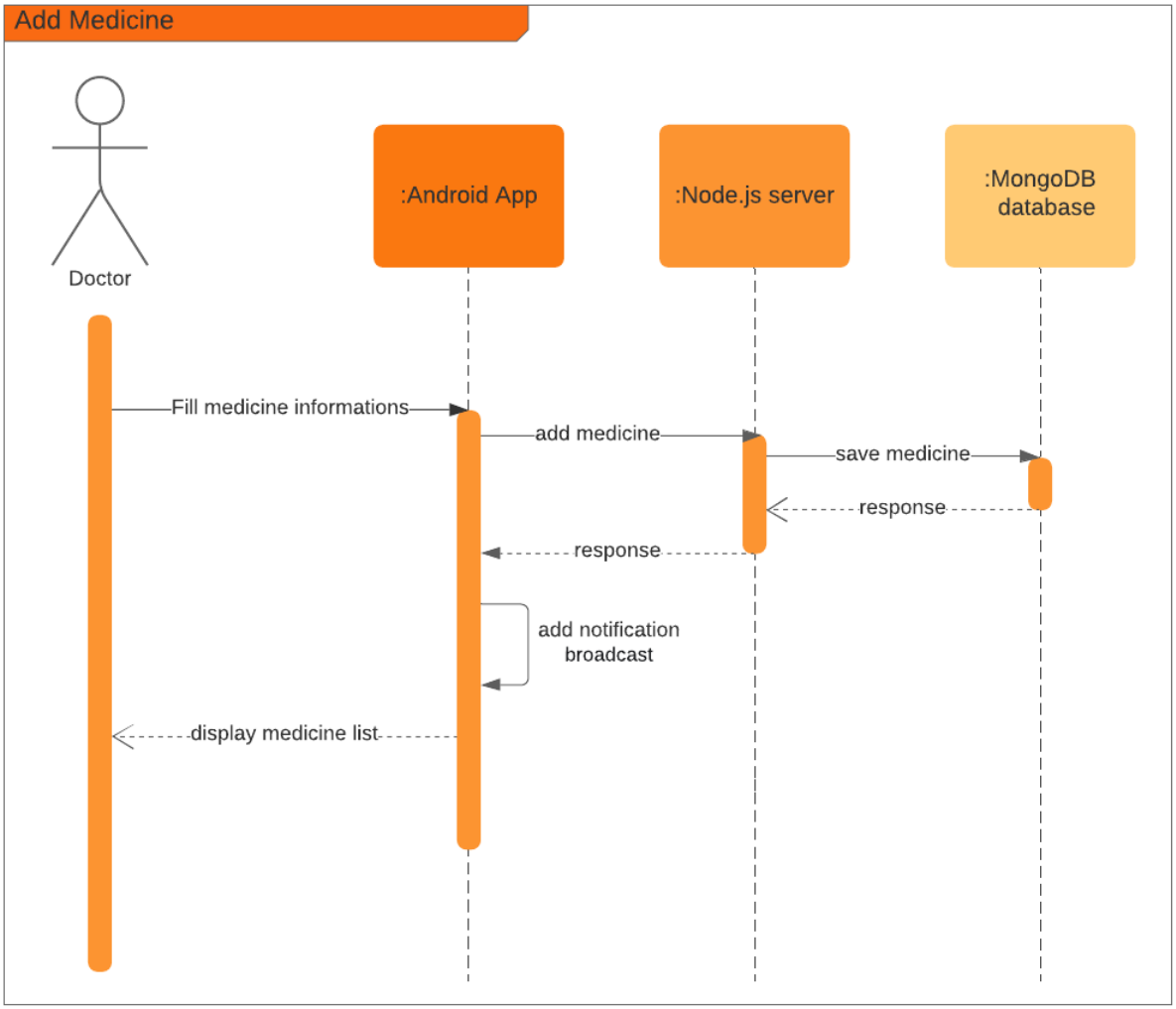


Figure 5.2: Add medicine sequence diagram

5.3.2 Sequence diagrams : Add Dialysis Session

The patient has the option of adding dialysis sessions. The case's relative sequence diagram is shown below.

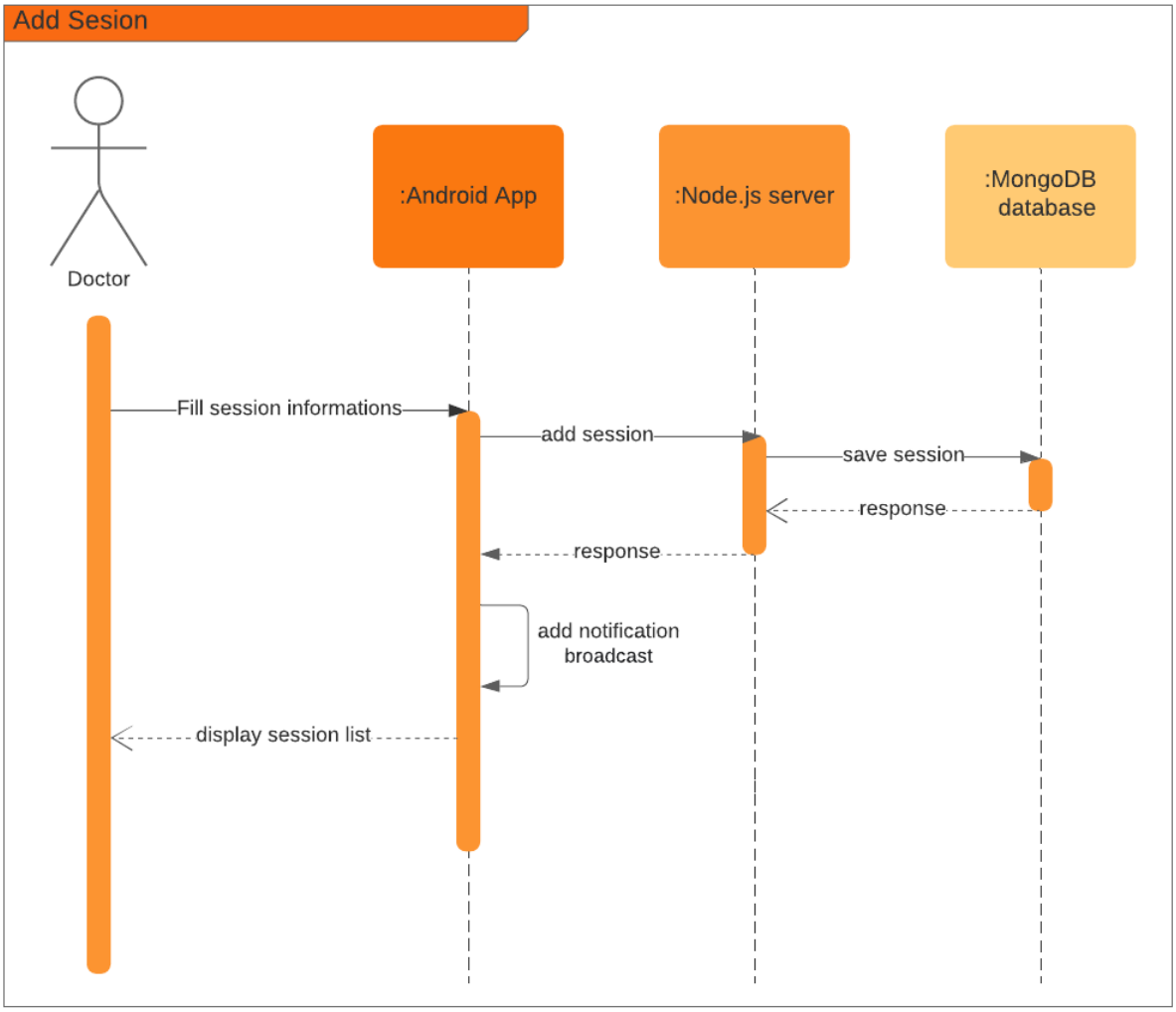


Figure 5.3: Add dialysis session sequence diagram

5.4 Activity diagram

In this sprint, the app's potential actions are depicted in the diagram below.

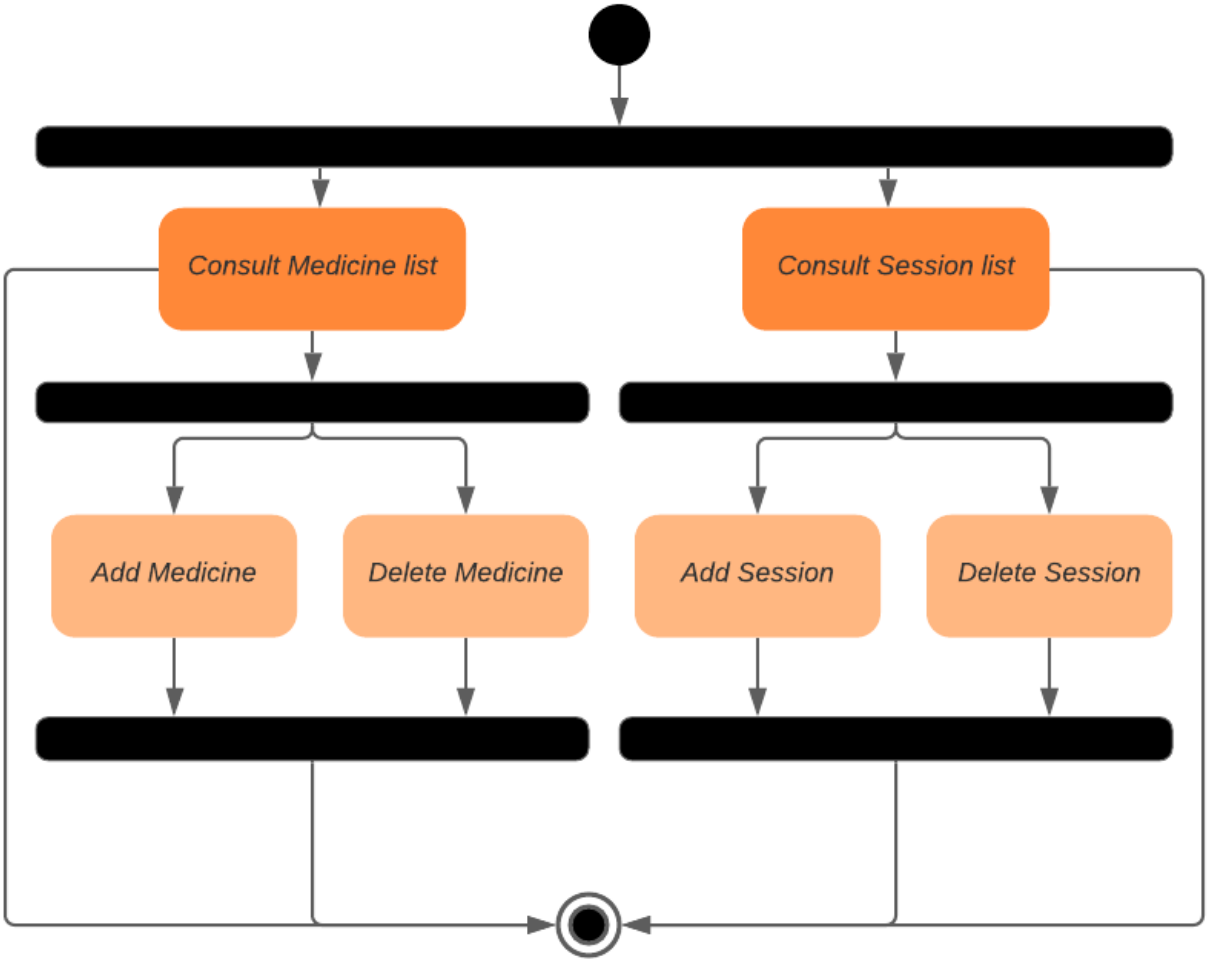


Figure 5.4: Sprint 3 activity diagram

5.5 Activities & Fragments

The activities and fragments covered in this sprint will be covered in this section.

5.5.0.1 Activities

We utilized the following activity in Sprint 3:

- MainActivity: This is the project's core activity, and it covers nearly all of the app's services.

5.5.1 Fragments

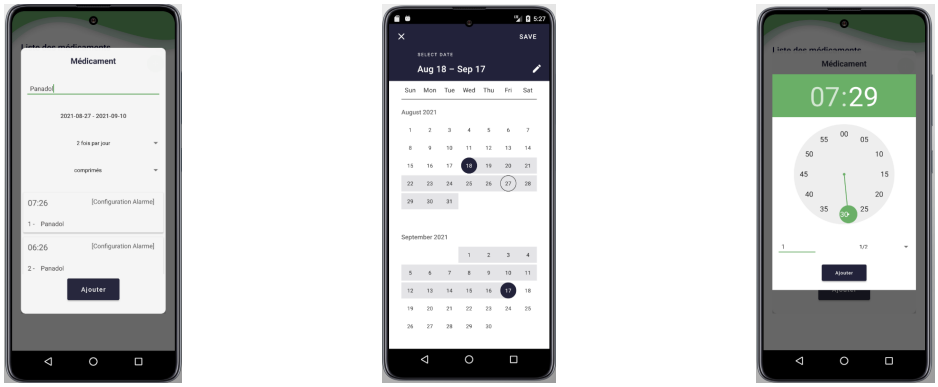
We utilized the following fragments in Sprint 3:

Fragment	Service	Attached to
ListPillFragment	Manage patient medicines	MainActivity
ListDialysisSessionsFragment	Manage patient dialysis sessions	MainActivity

Table 5.2: Fragments utilized in Sprint 3

5.6 Review

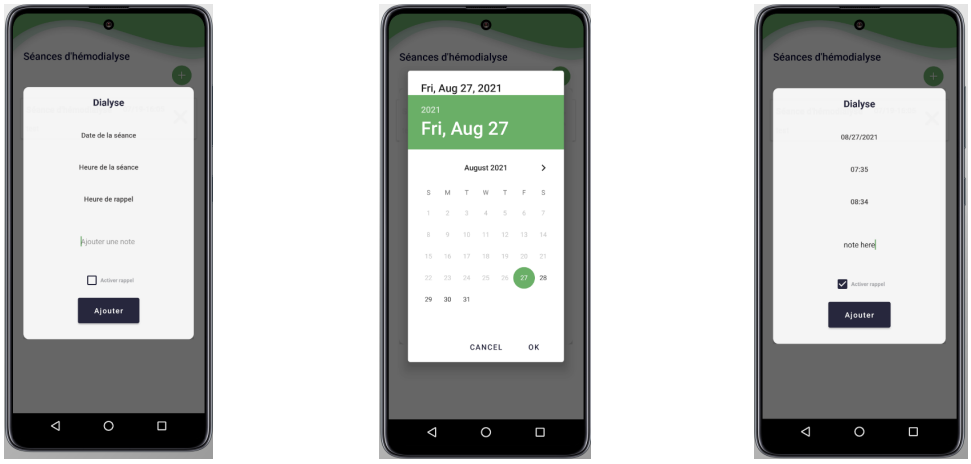
The goal of the review is to demonstrate what was done during the third sprint. We've highlighted the most important interfaces in this section.The processes for adding medicine are illustrated in the figure below.



(a) Add medicine interface (b) Add medicine period (c) Set medicine dose & time

Figure 5.5: Add medicine

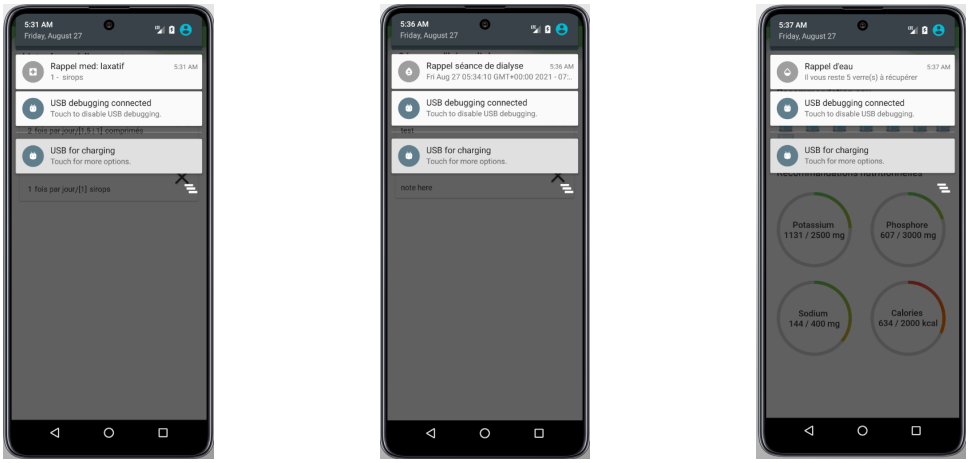
The processes for adding dialysis session are illustrated in the figure 5.6.



(a) Add dialysis session interface (b) Add dialysis session date (c) Set dialysis session reminder

Figure 5.6: Add dialysis session

The user getting notifications to remind him of medicine, dialysis session, and water is depicted in the figure 5.7.



(a) Medicine reminder (b) Session reminder (c) Water reminder

Figure 5.7: Reminder notifications

5.7 Sprint retrospective

Throughout this sprint, the work moved as planned, and everything went well.

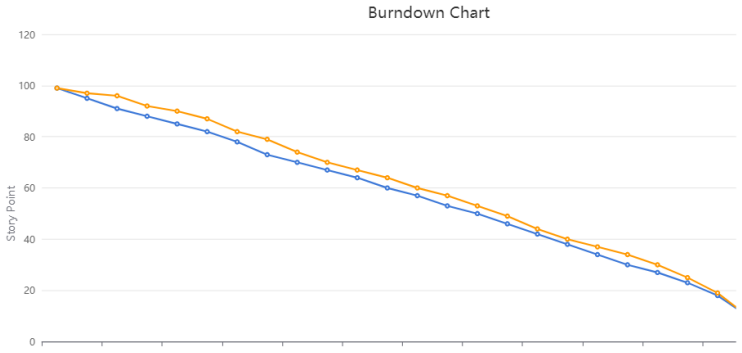


Figure 5.8: Burn down chart sprint 3

The blue curve represents the estimated burndown , while the orange one represent the actual burndown.

Conclusion

In this sprint, we used a wide definition that included use case diagrams, system sequence diagrams, activity diagrams, and class diagram. Finally, we showed off a variety of app interfaces.

CHAPTER

SIX

SPRINT 4 : CHATBOT

Introduction

The purpose of this fourth sprint, "Chatbot," is to create an artificial intelligence-powered chatbot to answer some of the most common patient questions about kidney disease and diet issues. We developed an algorithm later in the sprint that assigns a badge to the user based on how frequently he updates his daily log.The goal of this badge is to encourage users to utilize the app on a daily basis.

6.1 Sprint Backlog

Sprint Backlog :

ID	User story	Acceptance criteria	Priority	Estimation
AP-38	As a patient, I am able to ask the chatbot Question	The chatbot must send reply to the patient	38	20
AN-39	As a patient, I am able to get a badge based on filling the daily log daily	if the patient meats the badge requirements badge should be displayed	39	5

Table 6.1: Chatbot Sprint Backlog Sorted by priority

6.2 Natural Language Processing

Computational linguistics—rule-based human language modeling—is combined with statistical, machine learning, and deep learning models in NLP. These technologies, when used together, allow computers to interpret human language in the form of text or speech data and 'understand' its full meaning. NLP tasks help the machine understand what it's consuming by breaking down human text and speech input in patterns that the computer can understand. Here is some of the NLP tasks we utilized in building the chatbot :

- **Tokenization** : Tokenization is the process of breaking down a part of the text into small tokens. Tokens can be words, characters, or subwords.

```python
import nltk
tokenizer = nltk.data.load('tokenizers/punkt/french.pickle')

def tokenize(sentence):
    tokens = tokenizer._tokenize_words(sentence)
    str_words = [str(w) for w in tokens]
    return str_words
```

Figure 6.1: Tokenize function using nltk library

example:

» **input** : "Bonjour, je voudrais manger une banane."

» **output** : ['bonjour', ',', 'je', 'voudrais', 'manger', 'une', 'banane', '.']

- **Stemming** : The process of reducing a word to its word stem, which affixes to suffixes and prefixes or to the roots of words known as a lemma, is known as stemming.

```python
import spacy
stemmer= spacy.load('fr_core_news_md')
def stem(sentence):
    x=[]
    doc = stemmer(sentence)
    for token in doc:
        x.append(token.lemma_.lower())
    return x
```

Figure 6.2: Stem function using spacy library

example:

» **input** : [’bonjour’, ’,’, ’je’, ’voudrais’, ’manger’, ’une’, ’banane’, ’.’]

» **output** : [’bonjour’, ’,’, ’je’, ’vouloir’, ’manger’, ’une’, ’banane’, ’.’]

Tokenization, stemming, and lemmatization are amongst the most primitive natural language processing techniques. In this section, we learned how to tokenize and lemmatize using the spaCy and NLTK libraries.

6.3 Data

The chatbot’s capabilities are influenced by the amount of data utilized to train it. It’s true that the data volume we utilized in this project was small, but it may be increased in the future. The following is the data structure:

Tags: are several types of user intent when they ask a question.

Patterns: The typical methods in which consumers ask inquiries about a certain tag

Responses: The model can pick from a collection of predefined responses for each tag in the dataset to react to a specific query.

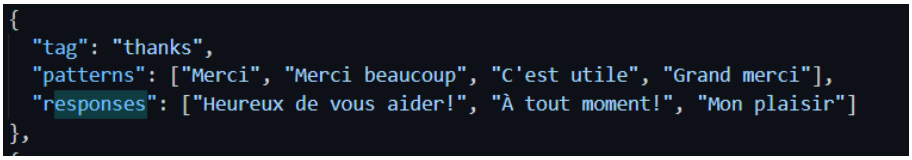


Figure 6.3: Data Structure

The group to which each message belongs is indicated by the tag on each dictionary in the file. We’ll use this data to train a neural network to categorize a phrase of words as one of the tags in our dataset. Then all we have to do is get a response from those groups and show it to the user. The chatbot will be smarter and more complicated the more tags, replies, and patterns we supply.

6.4 Model architecture

A Feed Forward Neural Network is a type of artificial neural network in which nodes are connected in a circular pattern. A recurrent neural network is the polar opposite of a feed forward neural network, in which specific routes are cycled. Because input is only processed in one direction, the feed forward model is the simplest kind of neural network. Regardless of how many hidden nodes the data passes through, it always travels in one direction and never reverses. We used a simple Feed Forward Neural Network with two hidden layers in this project.

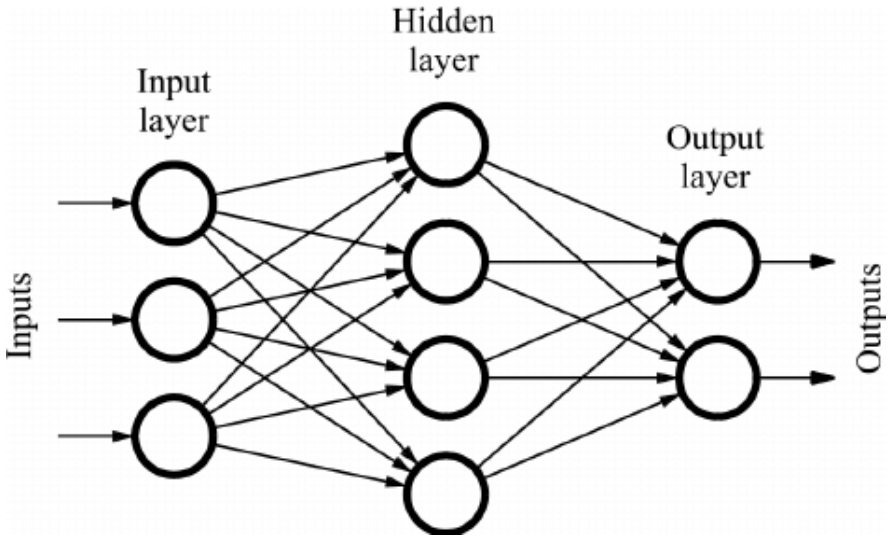


Figure 6.4: Feed Forward Neural Network with one hidden layer architecture

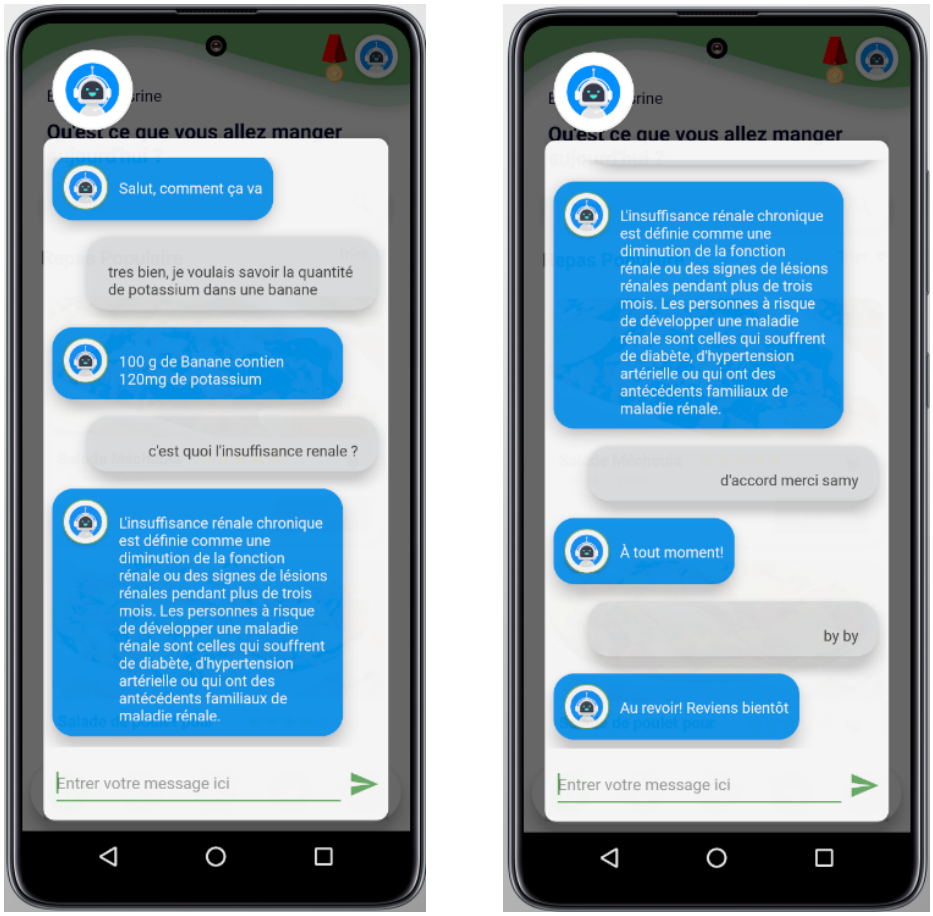
An epoch is a machine learning term that refers to how many rounds the machine learning algorithm has made across the whole training dataset. Batches are often used to organise data collections. With epochs set to 1000 for training, the best model was found at 1000th epoch:

```
Epoch [1000/1000], Loss: 0.0025
final loss: 0.0025
training complete. file saved to data.pth
```

Figure 6.5: Final Loss

6.5 Review

The purpose of the review is to show what was accomplished during the fourth sprint. Here's an example of a possible conversation between the user and the chatbot.



(a) (b)

Figure 6.6: Chatbot Conversation

Finally, to encourage users to check in to the app on a regular basis, we designed an algorithm that awards the user a badge based on how frequently he updates his daily record.

Figure 6.7: Badges

6.6 Sprint retrospective

We completed the final sprint as planned. The burndown chart for the last sprint is shown below.

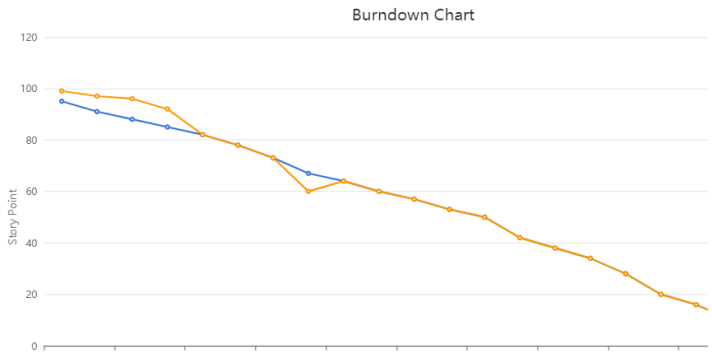


Figure 6.8: Burn down chart sprint 4

The blue curve represents the estimated burndown , while the orange one represent the actual burndown.

Conclusion

Throughout this sprint we explored some data science concepts and demonstrated several app interfaces.

CONCLUSION

In this report we have introduced the newly developed health mobile application. We went over the project's background, from the host company to the overall setting, to the issues stated and the suggested solution.This mobile application can assist many patients in keeping track of their mineral and nutrient intake. We've also studied at all of the Sprints in our backlog to come up with a design that emphasizes the system's static and dynamic elements while correctly portraying the aspect solution.After all of this effort, we can only hope that this program can help patients improve their habits and live better lives.

Several perspectives might be addressed at the end of this project. Starting with the chatbot, expanding the dataset to include all conceivable topics that patients could inquire about would undoubtedly improve the user experience. Depending on the users' behavior, this work might mark the start of a new generation of eHealth apps in Tunisia. If users trust our application, we may expect this project to grow in the future, especially by incorporate additional artificial intelligence and deep learning algorithms to help anticipate all possible complications.

REFERENCES

[1] *Republic of tunisia ministry of national defense,the military hospital of tunis*, http://www.hopmil.defense.tn/index.php/en.

[2] *Chronic kidney disease*, https://www.researchgate.net/publication/338563596_Chronic_Kidney_Disease.

[3] *Chronic kidney disease diet*, https://www.researchgate.net/publication/312641656_Diet_in_chronic_kidney_disease_in_a_Mediterranean_African_country.

[4] *Nutritional mobile applications for ckd patients*, https://www.sciencedirect.com/science/article/pii/S2468024918303115.

[5] *Mobile phone applications for kidney patients*, https://www.jrnjournal.org/article/S1051-2276(13)00097-6/pdf.

[6] *Data source: U.s. department of agriculture*, https://www.usda.gov/topics/food-and-nutrition/.

[7] *Scrum*, https://www.scrum.org/resources/what-is-scrum.

[8] *Android*, https://www.dotnettricks.com/learn/android/what-is-android-and-why-to-use-it.

[9] *Node.js*, https://medium.com/selleo/why-choose-node-js-b0091ad6c3fc.

[10] *Mongodb*, https://www.mongodb.com/why-use-mongodb.

[11] *The best language to build ai chatbot*, https://techcrunch.com/2017/12/20/choosing-the-best-language-to-build-your-ai-chatbot/.

[12] *Model-view-viewmodel (mvvm)*, https://www.wintellect.com/model-view-viewmodel-mvvm-explained/.

[13] *Model-view-controller (mvc)*, https://techterms.com/definition/mvc.

[14] *Flask*, https://pythonbasics.org/what-is-flask-python/.

[15] *Natural language processing with python's nltk package*, https://realpython.com/nltk-nlp-python/.

[16] *Natural language processing with spacy in python*, https://realpython.com/natural-language-processing-spacy-python/.

[17] *Deep learning: Feedforward neural network*, https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7.

Curiginal

Document Information

Analyzed document	Rapport_NesrineNouira_VF.pdf (D113287322)
Submitted	2021-09-23 16:41:00
Submitted by	
Submitter email	asma.mabrouk@esprit.tn
Similarity	1%
Analysis address	asma.mabrouk.esprit@analyse.urkund.com

Sources included in the report

W	URL: https://www.sciencedirect.com/science/article/pii/S2468024918303115. Fetched: 2021-09-24 13:24:00	2
SA	**ESPRIT / Mhamed-Hattab.pdf** Document Mhamed-Hattab.pdf (D40561847) Submitted by: imed.amri@esprit.tn Receiver: imed.amri.esprit@analyse.urkund.com	1
W	URL: https://www.clearlyagile.com/agile-blog/the-ultimate-guide-to-the-sprint-backlog Fetched: 2021-09-24 13:24:00	1
W	URL: https://studentnet.cs.manchester.ac.uk/resources/library/thesis_abstracts/MSc13/FullText/Tan-ChinLoong-fulltext.pdf Fetched: 2020-05-07 19:29:35	1
SA	**rapport_Ali_bayouth_DNI.pdf** Document rapport_Ali_bayouth_DNI.pdf (D75732011)	5
W	URL: http://www.hopmil.defense.tn/index.php/en. Fetched: 2021-09-24 13:24:00	1