Technical Report - **Product specification**

MediSync

Course:	IES - Introdução à Engenharia de Software
Date:	Aveiro, 16/12/2024
Students:	113480: Afonso Ferreira 113482: João Neto 98059 : Pedro Ponte 115243: Ricardo Antunes
Project abstract:	MediSync is a multi-layered service that helps with hospital management by integrating various functionalities that help with room/patient management and provide real-time patient information to help nurses deliver optimal care.

Table of contents:

1 Introduction

The goal for the project was to propose, justify and implement a multi-service application that featured live data being fed into our system. Besides the actual code development, it also serves as a way to master collaborative work practices in agile project architecture.

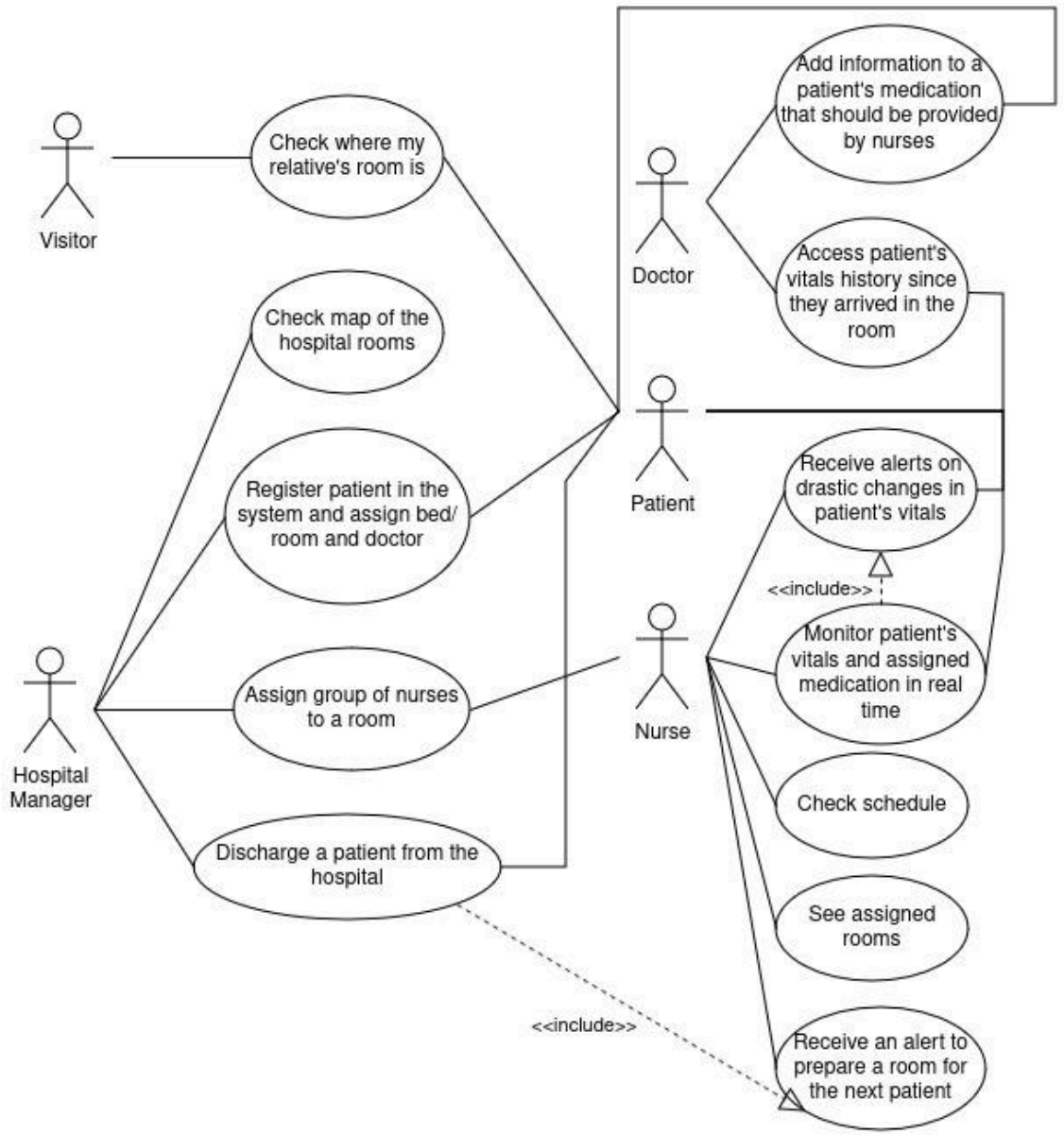
2 Product concept

Vision statement

MediSync is a web / app software that aims to aid hospital staff in organizing personnel and patient allocation in hospitalization bedrooms, in order to increase efficiency and organization, as well as aid said personnel in keeping tabs on the patient's condition while he / she is hospitalized and is undergoing treatment.

While laying out the expectations of our system, we had initially compromised with also maintaining coordination on surgeries and regular appointments, but we determined that covering so many aspects of hospital management would prove detrimental, and instead decided to focus on a single aspect of hospital management that could work perfectly, being able to expand to other fields in the future should that be the direction of the system.

Our system serves as an aid to the regular hospital systems, which are commonly known to be clunky and oftentimes inefficient, as many of us already experienced. Thus, our system is not particularly different from a subsystem you could find at a regular hospital, but it competes with that system as it is specifically gauged to maintaining coordination within the hospital's resources, initially on a smaller scale.



Gathering the requirements turned out to be a difficult process - as we laid out the expectations for what we wanted our system to do, we also needed to maintain the scope of our system relatively low (as mentioned above). We used a principle of “using a sniper instead of a shotgun”, so to say, we focused on wanting to make our main aim of the system as good as possible, rather than having many different mediocre features. Once we had those expectations laid out, we aimed at having 12 User Stories balanced between different personas (except with the visitor), and after that, we suggested many different stories and filtered those who we believed were the best at achieving the “Sniper” effect.

Personas and Scenarios

Personas

Peter is a 29 year old **nurse** at Wood Hospital and he has been working at the hospital for 5 years now. He provides care to multiple patients by monitoring their vital signs, giving injections, among other tasks. He is young so he has an ease with technology and the hospital machines which helps him with his job but he is frustrated by the current state of the Wood Hospital which is very outdated with old machines with no connectivity which makes his job harder and more demanding. He thinks that the patients deserve the best and optimal care possible while also being able to do other important tasks in the hospital

Motivation: Peter would like to have a way to check his timetable to be more efficient in his workplace and also be able to receive alerts from unusual vitals from one of his patients.

Nadia is a 43 year old that has been practicing **medicine** for almost two decades. She has a large number of patients, all very diverse with different conditions. She is analytical and likes to check the medical history of patients to see how they are reacting to the treatment. She also has a lot to do so she likes do be quick and communicate as fast as possible with staff

Motivation: Nadia is driven to find solutions that allow her to quickly access comprehensive patient information, make informed decisions, and communicate rapidly with staff to ensure timely treatments.

David has been a member of Wood Hospital since it opened, 25 years ago. He kept advancing in his career until he got to his current position of **hospital manager.** He controls all the patient rooms in the hospital, as well as the staff (nurses and doctors). Since he has to deal with a lot of people it makes it hard to communicate all the decisions and changes made to everyone in an efficient time. David's job is very demanding since the number of rooms is limited and he has to take into account multiple factors, for example, the fact that some patients can't be in the same room as others, because of contagious diseases.

Motivation: David wants to find a way to check room availability in room time, be able to assign patients and staff to each room in an efficient way, optimizing hospital resources.

Maria recently found out that her grandmother got sick and was put in a hospital with pneumonia and wants to visit her at Wood Hospital. She doesn't live in the town where the hospital is located, so she has never been there. She also doesn't have a great sense of direction and worries a lot about getting lost.

Motivation: Maria is motivated by the desire to quickly and easily find her grandmother's hospital room, so she can spend as much time as possible with her while minimizing the stress and confusion of navigating an unfamiliar hospital.

Scenarios

Scenario 1

Peter starts his shift at Wood Hospital, working with outdated machines that require him to manually check and record patient vitals. While giving medications to a patient, he misses an unusual heart rate in another patient due to the lack of real-time alerts. A month later, MediSync is introduced, a system that notifies him immediately of any unusual vitals from his patients while also recording their data in real-time. Besides that in the same platform he can also check his schedule which allows him to do this work efficiently.

Scenario 2

Nadia starts her day by providing an appointment to an hospitalized patient. She quickly gets all the information about the patient in MediSync and realizes that the health of the patient is not improving, deciding to increase the dosage of medication. She updates the respective dosage in the MediSync service that will make sure the nurses are aware of the changes.

Scenario 3

David opens his computer with the MediSync service and quickly realizes that a lot of patients are going to be coming in and need a room to stay in. He checks which rooms are available and easily separates patients with similar contagious conditions that can be given and the same room and which can't. He quickly gives each patient and a room and then quickly assigns a nurse and a doctor to each patient so they are well taken care of.

Scenario 4

Maria has just arrived at the hospital to visit her grandmother and quickly gets overwhelmed with the amount of rooms and people. She opens the MediSync service and selects an option to "Check Room" she inserts a phone number, which is in the list of visitors and quickly gets a code as a text message. She inserts the code and a map of the hospital with the room of her grandmother is shown giving her clear directions

Product requirements (User stories)

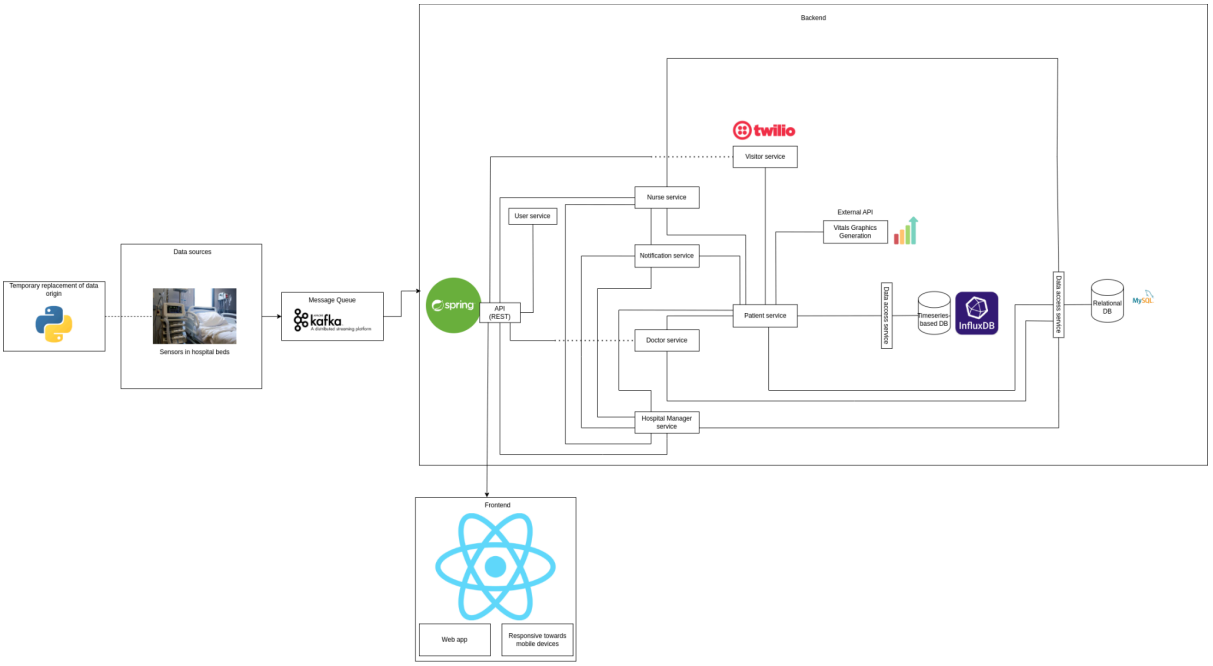
- **Hospital Room and Bed Management**
  - **As** David, a hospital manager, **I want to** be able to check a map of the rooms and the respective beds, to check bed/room availability and which rooms have patients with contagious diseases **so that** I can optimize patient placement and hospital management
  - **As** David, a hospital manager, **I want to** be able to register a patient in the system and assign a bed/room and a doctor to the patient **so that** the hospital can be more organized
  - **As** David, a hospital manager, **I want to** be able to easily discharge a patient from the hospital **so that** I can optimize bed/room usage and help patients quickly
  - **As** Peter, a nurse, I want to receive an alert when a patient is discharged, **so that** I know which rooms/beds need to be prepared.
- **Staff Assignment and Scheduling**
  - **As** David, a hospital manager, **I want to** assign a group of nurses to a specific room and a specific time frame by selecting the room, the time frame and selecting their names from a list of available nurses **so that** I can ensure adequate staffing and that each room has staff assigned to it.
  - **As** Peter, a nurse, **I want to** check my assignment schedule for the day **so that** I can know where I'll be needed next.
  - **As** Peter, a nurse, **I want to** see what rooms I am assigned to **so that** I can be prepared and present in the correct locations to provide care for patients.
- **Patient Health Monitoring and Alerts**
  - **As** Peter, a nurse, **I want to** be able to monitor the vital signs and assigned medication of the patients in the rooms I'm assigned too in real-time **so that** I can ensure their health status is stable and respond promptly to any changes
  - **As** Peter, a nurse, **I want to** be able to receive an alert if something unusual like a drastic change in heartbeat happens to my patient, **so that** I can help them as fast as possible.
  - **As** Nadia, a doctor, **I want to** be able to access a patient's vital signs history since they arrived at their room **so that** I can have a bigger picture of how their health progressed since they arrived
  - **As** Nadia, a doctor, **I want to** be able to add information to each patient about necessary medication that needs to be provided by the nurses at a specific time **so that** they can make sure that the treatment is made while the patients are in their rooms.
- **Visitor Management**
  - **As** Maria, a visitor, **I want to** be able to check where my relative's room is by receiving a secret code from the application sent by the reception **so that** when I enter the hospital to visit them, I know where to go.

3 Architecture notebook

Key requirements and constraints

As a system that will be deployed at hospitals, we will surely have to deal with adaptability to legacy systems. The sensors in the current hospital beds currently do not have the technology necessary to transmit the information they capture (specifically regarding vitals) towards our system. This is a hardware dependency that we can't (and won't) control in this early part of development. Similarly, as our system would be part of critical infrastructure, it needs to be available as long as possible. To this end, there need to be solutions for situations such as database connection issues, and with every deployment we absolutely need to make sure that it never encounters a bug while deployed, as such, we believe that the best approach for CD would be the Blue-Green Deployment strategy. With this, if we detect that there is an issue with a newly deployed version, we can always backtrack to the previous version. This is a viable strategy due to the fact that almost all of our changes after the first deployment are going to be impacting the interface of the system, rather than the inner-working of it. Our system will be provided as a web app, but it will also be responsive to other screen configurations, to accommodate towards mobile phones. This is done because nurses will then be able to have the application open in their phones, being able to check the information they need while walking or if they can't access a computer at the time.

Architectural view



Data sources: We have 1 data source, which consists of all the sensors near the hospital bed. These sensors collect the information regarding a patient such as heartbeat, oxygen

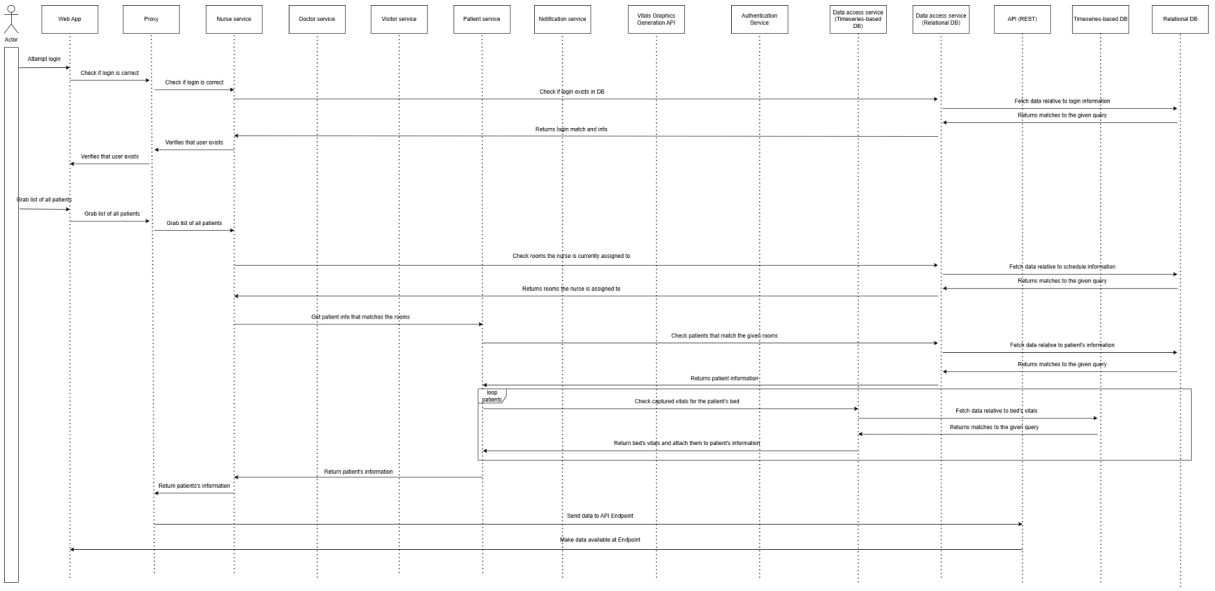
saturation and other information.

Message Queue: We will use *Apache Kafka* to handle communications between the sensors and the backend. This message queue is very reliable as it can store data as long as necessary, unlike other message queues. Reliability is a very important asset in this system, therefore this feature is very important as well

Backend: Our backend will comprise 2 databases, one of them relational, the other timeseries-based. The timeseries-based database will store the live information related to patients, specifically regarding their vital signs. It will store every information that comes grouped by bed. The relational database will serve to manage who can see what (by establishing relationships between every entity), as well as serve for authentication. In short, we will access this database to know what to search in the timeseries-based database. We will have our own API which will serve the front-end through its endpoints, which we will implement using Spring Boot.. We also have services that will process the data regarding the hospital bed sensors (and update the databases accordingly), as well as when a visitor / staff does an action that affects other users. We will use an external API to generate graphs based on the data collected in the timeseries DB, as well as an authentication tool (twillio) that sends a confirmation code to patients to verify their identity in a first stage. We will also have a notification service that delivers all types of notifications to other services.

Frontend: We will use React to develop the website interface. We will access the API's endpoints to gather the information from the databases. Our web app will not have a mobile version, but it will be compatible with mobile screens.

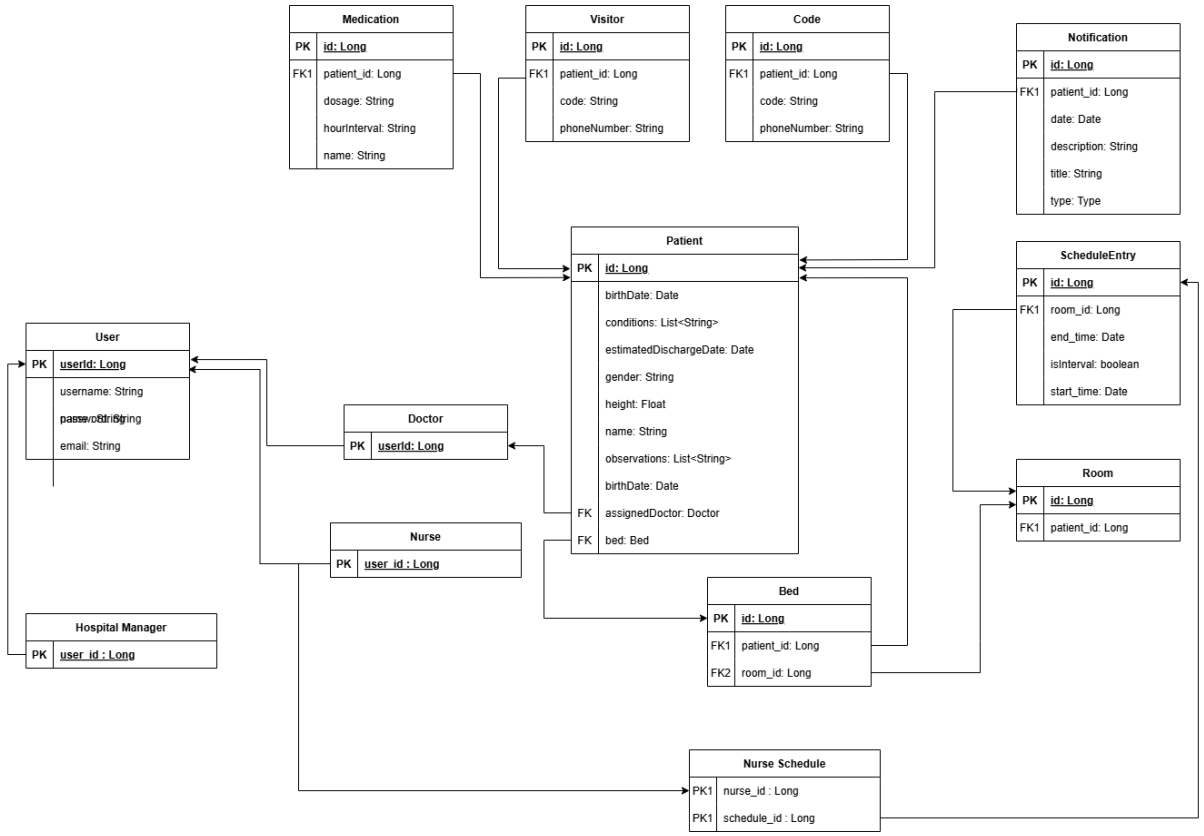
Module interactions



This example of an interaction shows how the system would behave when a user (nurse) logs in the system and goes to a page that contains all the information pertaining to patients they're assigned to. In the login stage, the proxy grabs the information form the Nurse's service module, which goes to the Relational Database to gather the information it needs regarding credentials. If the database finds a match, it will be returned to this service, which gives it back to the proxy making the communication between the frontend and the backend.

With the login information saved and the user authenticated, when the user wants to get the information regarding their patients, the Nurse's module will first go to the relational DB to fetch which rooms this nurse has access to. After grabbing that information, it turns to the Patient's service module which is responsible for fetching all patient-related information, no matter for who it is. After fetching this information, it gives it to the Doctor's service module, which sends it to the API, and the API returns the JSON to the frontend. When making things this way, it is impossible for a user without access to grab data from patients, unless they directly access the DB, which is hidden behind many layers.

4 Information perspective



4.1 - API Endpoints

Documentation done using Swagger. To access the documentation run the project and access port 8082, or use the link in the repository README.

5 Future Work

Since an application like this never stops growing, after the release of the product some extra features could be developed to improve the product:

1. Schedules for doctors
2. Edit and remove schedules
3. Only allow visitors in the hospital with booking
4. Include monitorization on kafka messages, user requests, database queries, etc.
5. Improve password security
6. Have more rooms for specific diseases, for example a burn unit, oncological patients, etc.
7. Add doctor specializations and automatic patient attribution depending on their conditions

6 References and resources

- Chartjs
- Kafka
- React
- OpenAPI
- Information about vitals (temperature/blood pressure/pulse, blood oxygen level)
- Spring
- Python requests