AASS Semestral project

Lukáš Častven, Michal Kilian

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

`xcastven@stuba.sk, xkilian@stuba.sk`

March 29, 2025

1 Application overview

The name of our application is Healendar (heal + calendar).

2 Stakeholders

Stakeholder	Description
Patient	Wants to book appointment with doctor.
Nurse	Accepts or rejects appointment requests, requests more information from patients, and selects specialized rooms.
Doctor	Needs to see their workload and can override nurse decisions regarding appointments.
Other hospital staff	Needs to see the availability of other doctors and specialized rooms for their patients.
Hospital	Desires a new application to organize doctors and appointments.

Table 1: Stakeholders and Their Descriptions

3 Motivation

The following motivation diagram contains the drivers, assessments, goals, outcomes, and requirements related to the 4 identified stakeholders. Patient, Hospital, Nurse, Doctor.

The core problems, "High operational costs", "Long waiting times", and a "Bad reputation for possible new hires", drive the hospital for change. These drivers are associated with issues such as "Ineffective allocation", "Medical staff frustration", "Unavailability of specialized rooms", and patients' "Waiting times."
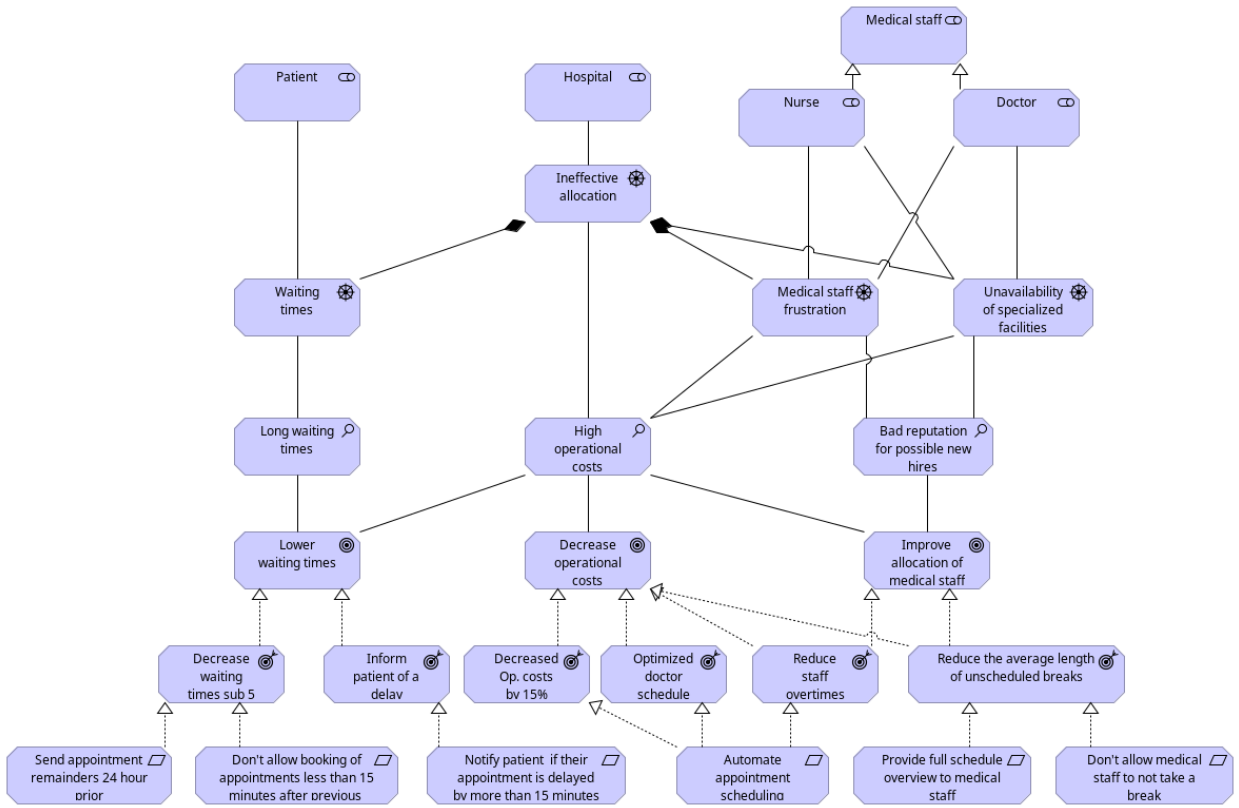
Figure 1: Motivation layer of Healendar

4 Value stream

The value stream of scheduling an appointment realizes the outcome of optimized doctor schedule.
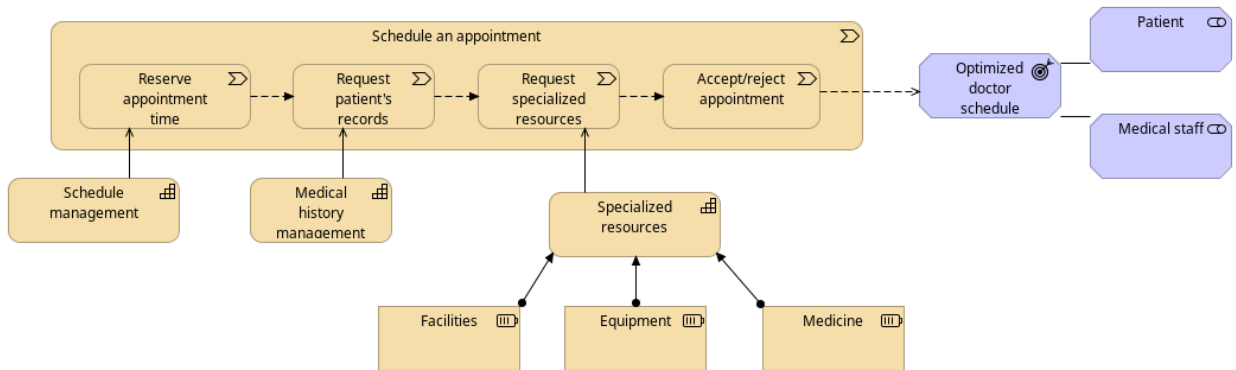

Figure 2: Value stream: Scheduling appointment

5 Business layer

Following diagram models three business processes, which serve the patient and medical staff to manage schedule.
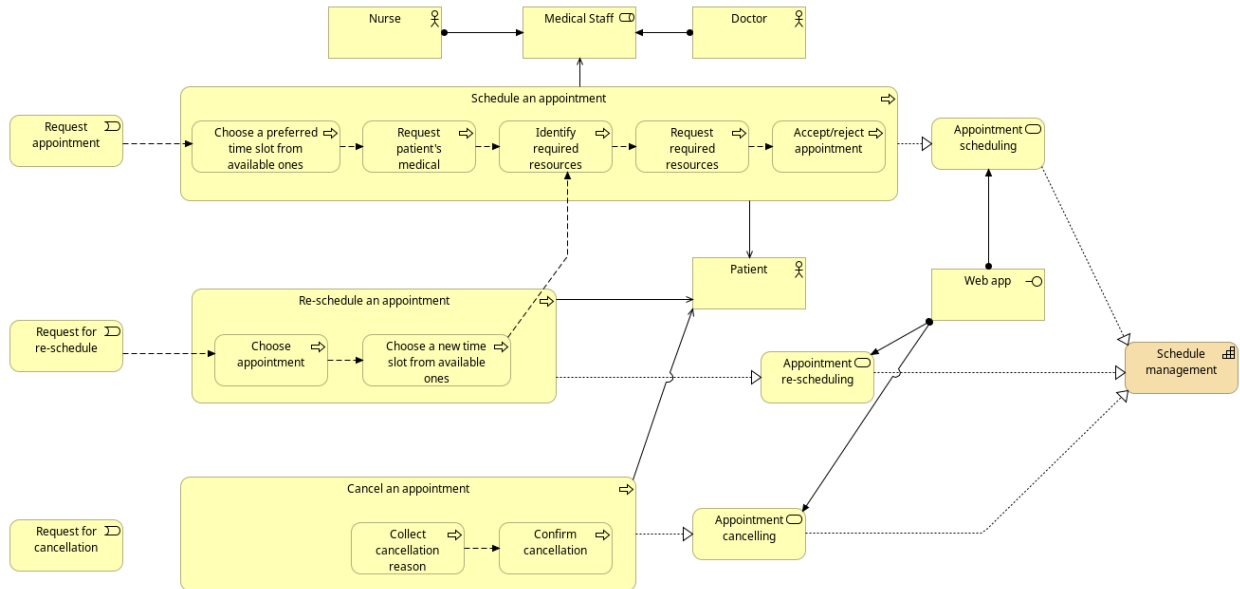

Figure 3: Business layer

6 Mockups

The patient interface, Figure 4, shows a design centered on appointment scheduling. Figure 5 illustrates the UI for the same use case, but with more control and additional resources requests for the medical staff.
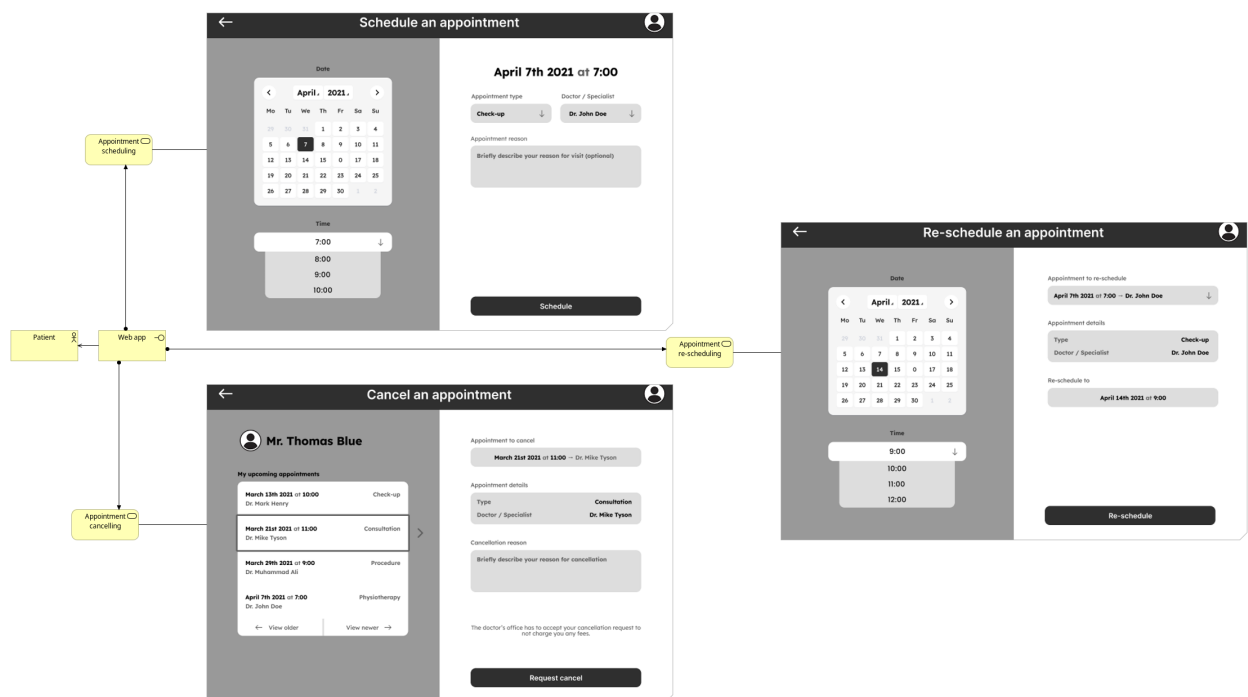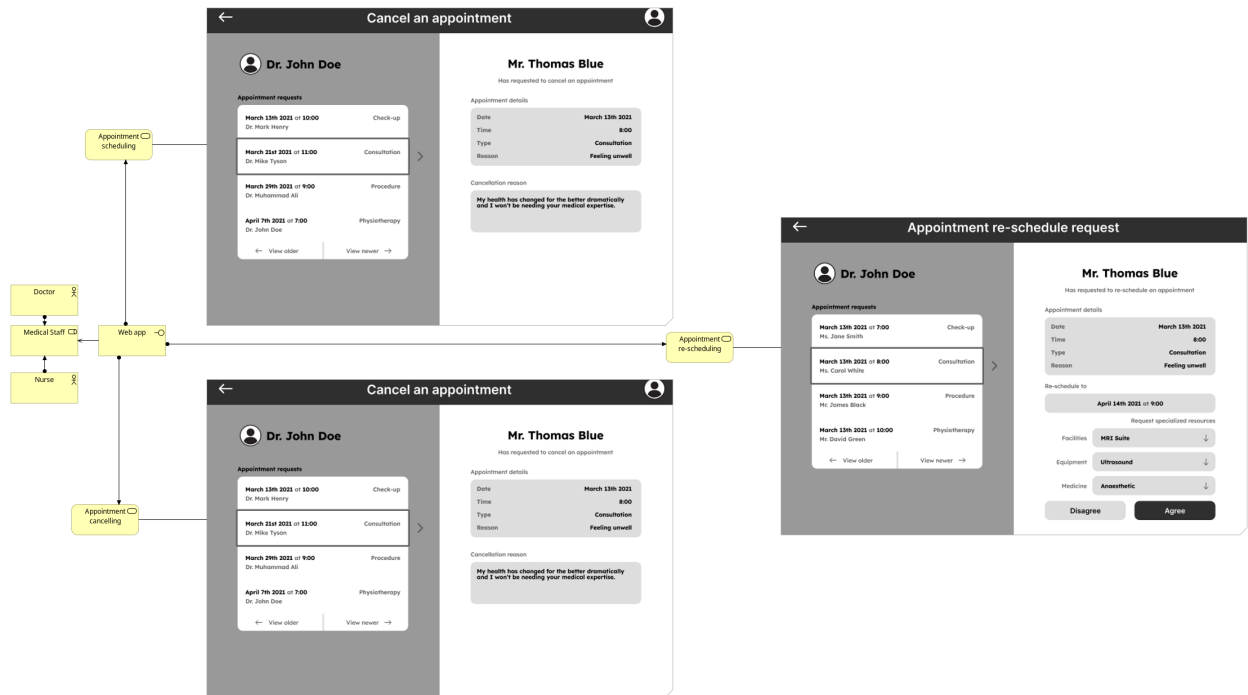
Figure 4: Patient's UI mockups

Figure 5: Medical staff's UI mockups

7 Architecture 1 application layer

We will implement a user interface and a backend components to support the "Schedule an appointment" business process. The implementation will consist of a monolithic backend written in Go and a web application frontend developed using Stencil.js.
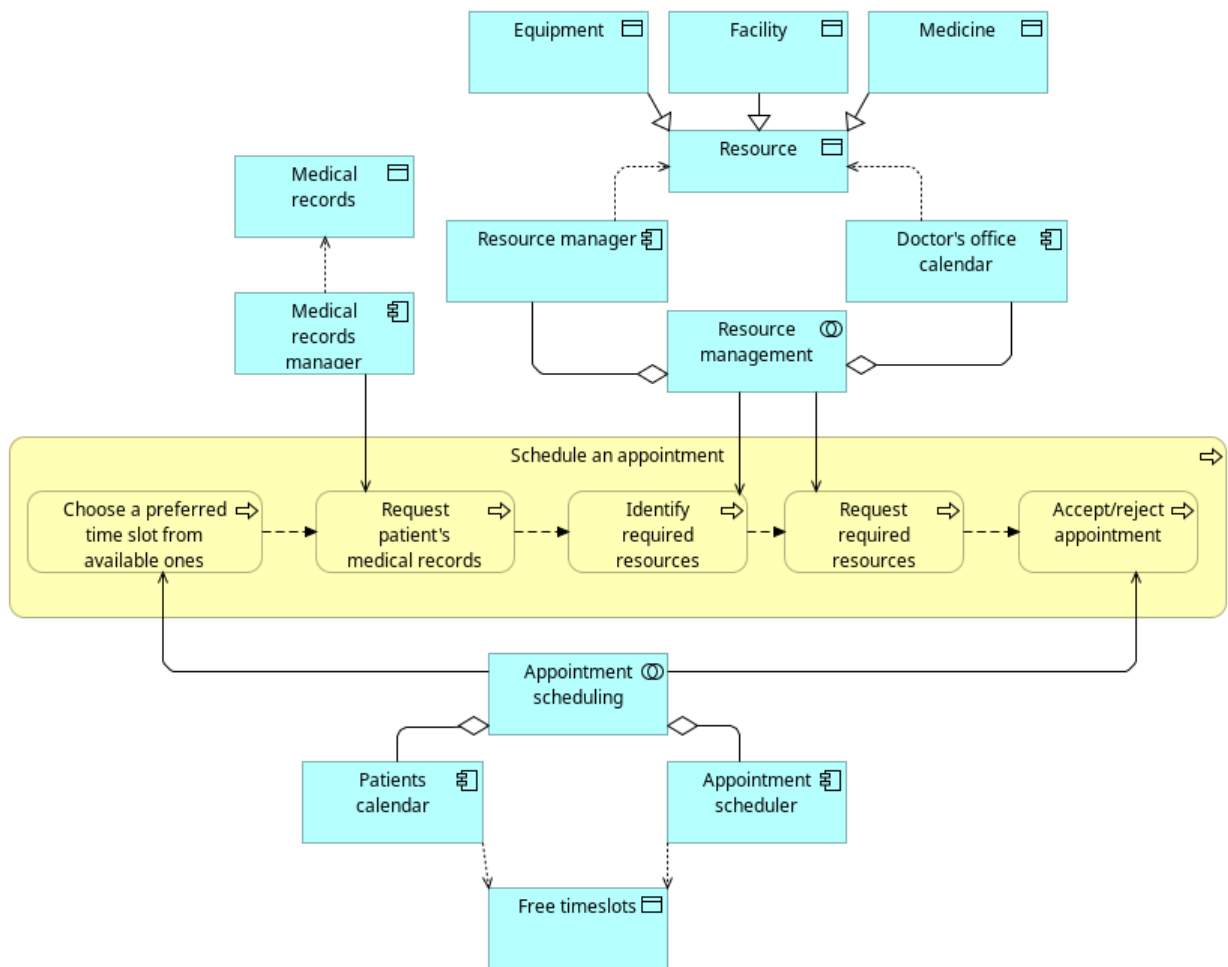
Figure 6: Application layer architecture supporting appointment scheduling