Surgical Appointment and Resource Management

Request for Proposals

v.2024.10.23

Disclaimer
(1) Parts of this document have been generated using AI models.
(2) The domain and requirements are a simplified version of the reality to adjust to the semester

1 Scope

The scope of this project is to develop a prototype system for **surgical requests, appointment, and resource management**. The system will enable hospitals and clinics to manage surgery appointments, and patient records. It will also offer real-time 3D visualization of resource availability within the facility and optimize scheduling and resource usage. Furthermore, the project will address **GDPR compliance**, ensuring the system meets data protection and consent management requirements.

Each module of the system must consider the legal aspects of the **GDPR Regulation (EU) 2016/679** and guarantee that users can access the privacy policy and exercise all relevant rights under this regulation.

Since this is a prototype, not all modules will need full implementation. The project proposal must clearly specify which functionalities are implemented.

Given that the project spans **14 weeks**, broken into **three sprints**, it is important to carefully allocate tasks, milestones, and responsibilities to ensure that the students can develop a working system by the end of the course. Each sprint will include a planning phase, development phase, testing, and review.

2 Overview

In this project, you will develop a web-based **surgical appointment and resource management system**. The system will consist of several modules:

- Backoffice Web Application
- 3D Visualization Module
- Planning/Optimization Module
- GDPR Module
- Business Continuity Plan (BCP)

The goal is to expose students to full-stack web development, REST APIs, database management, 3D rendering, optimization using Prolog, and privacy law (GDPR).

You will work in groups of 4, with each group member responsible for different parts of the system. Each student will be responsible for a distinct part of the system, but collaboration is essential for integrating the various components into a functional product. All user stories across the modules must align, and any dependencies between modules (e.g., 3D visualization relying on scheduling data from the planning module) must be carefully managed.

The project will run over 14 weeks, split into **three sprints**. Each sprint will involve developing and integrating system components while ensuring GDPR compliance and business continuity.

Throughout the project, you will conduct tests to ensure that all modules (backoffice, 3D visualization, planning, GDPR, and business continuity) are integrated and function correctly. Special attention should be given to the interaction between the **3D visualization** and **planning** modules, ensuring that real-time updates (such as room availability) are consistent across the system.

Overall, the backofffice module will manage:

- Medical professionals (doctors, nurses)
- Patients
- Operation types
- Rooms
- Chirurgic requests

That information is feed into the planning module which will:

- Generate the schedule of the surgeries, i.e., appointments.
- Optimize those schedules according to different criteria and the medical professionals and rooms availability.

The 3d module will render the hospital floor.

The team must also consider GDPR aspects in all user stories/requirements.

The team must also consider failover and business continuity aspects for the system.

Assume all other medical and patient management is managed in other parts of the system not part of this prototype. This prototype's scope is concerned with the appointment of surgeries.

3 Main concepts, attributes, and rules[1]

3.1 User

Represents a user in the system, either (i) an admin managing the system, (ii) a healthcare staff (e.g. doctor, nurse, technician) or (iii) patients interacting with it.

- Attributes:
  - `Username`
  - `Role` (e.g. Admin, Doctor, Nurse, Technician, Patient)
  - `Email`
- Rules:
  - Backoffice users are registered by the admin in the IAM through an out-of-band process.
  - Patient users are self-registered using the IAM.
  - The user's IAM record is linked to the respective user and staff/patient record in the backoffice data.
  - All users authenticate using the IAM.

3.2 Patient

Represents individuals receiving medical care.

[1] Clarifications on the attributes and rules must be caried out by each team during the weekly session with the customer or through the "RFP clarification" forum (in Moodle)

- Attributes:
  - `First Name`
  - ‘Last Name’
  - ‘Full Name’
  - `Date of Birth`
  - `Gender`
  - `Medical Record Number` (unique identifier)
  - `Contact Information` (email, phone)
  - `Allergies/Medical Conditions` (optional)
  - `Emergency Contact`
  - `Appointment History` (list of previous and upcoming appointments)
- Rules:
  - A patient must be unique in terms of `Medical Record Number`, `Email` and ‘Phone’.
  - Sensitive data (like medical history) must comply with GDPR, allowing patients to control their data access.

3.3 Staff

Represents the professional providing healthcare.

- Attributes:
  - `First Name`
  - ‘Last Name’
  - ‘Full Name’
  - `License Number` (unique identifier)
  - `Specialization` (e.g., cardiology, orthopedics). Each medical professional has only one specialty
  - `Contact Information` (email, phone)
  - `Availability Slots` (the list of time slots the staff defines as being available for appointments)
- Rules:
  - A staff must be unique in terms of `License Number`, `Email` and ‘Phone’.
  - Staff define the availability slots, e.g. slot 1: 2024-09-25:14h00-18h00; slot2: 2024-09-25:19h00/2024-09-26:02h00.
  - The availability slots remain unchanged when slots are used for an appointment.
  - Staff can handle multiple appointments but cannot be double-booked at the same time.

3.4 Operation request

The operation that is requested for later scheduling.

- Attributes:
  - `ID` (unique identifier)
  - `Patient ID` (linked to a specific patient)
  - `Doctor ID` (linked to a doctor that requests it)
  - `Operation Type ID` (the type of the operation to perform on the patient)
  - ‘Deadline date’ (the suggested deadline to perform the operation)
  - ‘Priority‘(the priority to perform the operation)

3.5 Operation Type

Represents predefined types of medical operations or procedures.

- Attributes:
  - `ID` (unique identifier)
  - `Name` (e.g., appendectomy, heart bypass)
  - `Required Staff by Specialization` (list of essential staff in respect to specialization)
  - `Estimated Duration` (of the operation type)

3.6 Appointment

Represents **scheduled** operation of a patient by a set of staff occurring in a room in a time slot.

- Attributes:
  - `ID` (unique identifier)
  - 'Request ID' (linked to the request that gave rise to this appointment)
  - `Room ID` (linked to a specific room)
  - `Date and Time` (of the operation)
  - `Status` (scheduled, completed, canceled)
- Rules:
  - Operations appointments must be assigned to a set of staff, a room in a time slot.
  - Operations cannot exceed the estimated time unless rescheduled.
  - An appointment cannot be scheduled if the staff or room is unavailable at that time.
  - The appointment type should match the staff's specializations and room availability.

3.7 Surgery Room

Represents surgery rooms in the healthcare facility for operations.

- Attributes:
  - `Room Number` (unique identifier)
  - `Type` (e.g., operating room, consultation room, ICU)
  - `Capacity` (maximum number of patients or staff)
  - `Assigned Equipment` (list of equipment in the room)
  - `Current Status` (available, occupied, under maintenance)
  - Maintenance slots (the slots defined for room maintenance)
- Rules:
  - Each room can host only one event at a time (either an appointment, surgery, or meeting).
  - The room's schedule must be managed based on doctor and equipment availability.
  - Assume that all rooms are fully equipped with the necessary equipment for every type of operation.

4 General Project Workflow

1. **Sprint planning**: At the beginning of each sprint, the group should meet to:
   a. Review the backlog of user stories and tasks.
   b. Discuss priorities and dependencies.
   c. Define the sprint goal and commit to the user stories the team believes it can complete within the sprint.
   d. Break down each user story into specific tasks and assign responsibilities to each student.

2. **Daily Standups/Weekly Meetings**: Each group should hold brief meetings (either daily or weekly) to ensure everyone is on track and aware of dependencies between modules.
3. **Code Reviews**: Regular peer code reviews should be held to maintain code quality and catch issues early.
4. **Continuous Integration**: Utilize a CI/CD pipeline for frequent testing and deployment to ensure that new features don't break existing functionality.
5. **Customer Feedback**: After each sprint, present progress to the "customer" (the professor) for feedback, ensuring that any adjustments are made before moving forward.
6. **Sprint retrospective**: At the end of each sprint, the group should hold a retrospective meeting to:
    a. Reflect on what went well, what could have gone better, and any challenges encountered during the sprint.
    b. Identify areas for improvement (both in the project workflow and technical implementation).
    c. Adjust workflows, tools, or processes for the next sprint to enhance productivity and collaboration.
7. **Documentation**: Each sprint should end with updated documentation, including a summary of progress, changes, and testing results.

5 Sprint 1

5.1 Backoffice module

5.1.1 As an Admin, I want to register new backoffice users (e.g., doctors, nurses, technicians, admins) via an out-of-band process, so that they can access the backoffice system with appropriate permissions.

Acceptance Criteria:

- Backoffice users (e.g., doctors, nurses, technicians) are registered by an Admin via an internal process, not via self-registration.

- Admin assigns roles (e.g., Doctor, Nurse, Technician) during the registration process.

- Registered users receive a one-time setup link via email to set their password and activate their account.

- The system enforces strong password requirements for security.

- A confirmation email is sent to verify the user's registration.

5.1.2 As a Backoffice User (Admin, Doctor, Nurse, Technician), I want to reset my password if I forget it, so that I can regain access to the system securely.

Acceptance Criteria:

- Backoffice users can request a password reset by providing their email.

- The system sends a password reset link via email.

- The reset link expires after a predefined period (e.g., 24 hours) for security.

- Users must provide a new password that meets the system's password complexity rules.

5.1.3 As a Patient, I want to register for the healthcare application, so that I can create a user profile and book appointments online.

Acceptance Criteria:

- Patients can self-register using the external IAM system.

- During registration, patients provide personal details (e.g., name, email, phone) and create a profile.

- The system validates the email address by sending a verification email with a confirmation link.

- Patients cannot list their appointments without completing the registration process.

5.1.4 As a Patient, I want to update my user profile, so that I can change my personal details and preferences.

Acceptance Criteria:

- Patients can log in and update their profile details (e.g., name, contact information, preferences).

- Changes to sensitive data, such as email, trigger an additional verification step (e.g., confirmation email).

- All profile updates are securely stored in the system.

- The system logs all changes made to the patient's profile for audit purposes.

5.1.5 As a Patient, I want to delete my account and all associated data, so that I can exercise my right to be forgotten as per GDPR.

Acceptance Criteria:

- Patients can request to delete their account through the profile settings.

- The system sends a confirmation email to the patient before proceeding with account deletion.

- Upon confirmation, all personal data is permanently deleted from the system within the legally required time frame (e.g., 30 days).

- Patients are notified once the deletion is complete, and the system logs the action for GDPR compliance.

- Some anonymized data may be retained for legal or research purposes, but all identifiable information is erased.

5.1.6 As a (non-authenticated) Backoffice User, I want to log in to the system using my credentials, so that I can access the backoffice features according to my assigned role.

Acceptance Criteria:

- Backoffice users log in using their username and password.

- Role-based access control ensures that users only have access to features appropriate to their role (e.g., doctors can manage appointments, admins can manage users and settings).

- After five failed login attempts, the user account is temporarily locked, and a notification is sent to the admin.

- Login sessions expire after a period of inactivity to ensure security.

5.1.7 As a Patient, I want to log in to the healthcare system using my external IAM credentials, so that I can access my appointments, medical records, and other features securely.

Acceptance Criteria:

- Patients log in via an external Identity and Access Management (IAM) provider (e.g., Google, Facebook, or hospital SSO).

- After successful authentication via the IAM, patients are redirected to the healthcare system with a valid session.

- Patients have access to their appointment history, medical records, and other features relevant to their profile.

- Sessions expire after a defined period of inactivity, requiring reauthentication.

5.1.8 As an Admin, I want to create a new patient profile, so that I can register their personal details and medical history.

Acceptance Criteria:

- Admins can input patient details such as first name, last name, date of birth, contact information, and medical history.

- A unique patient ID (Medical Record Number) is generated upon profile creation.

- The system validates that the patient's email and phone number are unique.

- The profile is stored securely in the system, and access is governed by role-based permissions.

5.1.9 As an Admin, I want to edit an existing patient profile, so that I can update their information when needed.

Acceptance Criteria:

- Admins can search for and select a patient profile to edit.

- Editable fields include name, contact information, medical history, and allergies.

- Changes to sensitive data (e.g., contact information) trigger an email notification to the patient.

- The system logs all profile changes for auditing purposes.

5.1.10 As an Admin, I want to delete a patient profile, so that I can remove patients who are no longer under car

Acceptance Criteria:

- Admins can search for a patient profile and mark it for deletion.

- Before deletion, the system prompts the admin to confirm the action.

- Once deleted, all patient data is permanently removed from the system within a predefined time frame.

- The system logs the deletion for audit and GDPR compliance purposes.

5.1.11 As an Admin, I want to list/search patient profiles by different attributes, so that I can view the details, edit, and remove patient profiles.

Acceptance Criteria:

- Admins can search patient profiles by various attributes, including name, email, date of birth, or medical record number.

- The system displays search results in a list view with key patient information (name, email, date of birth).

- Admins can select a profile from the list to view, edit, or delete the patient record.

- The search results are paginated, and filters are available to refine the search results.

5.1.12 As an Admin, I want to create a new staff profile, so that I can add them to the hospital's roster.

Acceptance Criteria:

- Admins can input staff details such as first name, last name, contact information, and specialization.

- A unique staff ID (License Number) is generated upon profile creation.

- The system ensures that the staff's email and phone number are unique.

- The profile is stored securely, and access is based on role-based permissions.

5.1.13 As an Admin, I want to edit a staff's profile, so that I can update their information.

Acceptance Criteria:

- Admins can search for and select a staff profile to edit.

- Editable fields include contact information, availability slots, and specialization.

- The system logs all profile changes, and any changes to contact information trigger a confirmation email to the staff member.

- The edited data is updated in real-time across the system.

5.1.14 As an Admin, I want to deactivate a staff profile, so that I can remove them from the hospital's active roster without losing their historical data.

Acceptance Criteria:

- Admins can search for and select a staff profile to deactivate.

- Deactivating a staff profile removes them from the active roster, but their historical data (e.g., appointments) remains accessible.

- The system confirms deactivation and records the action for audit purposes.

5.1.15 15. As an Admin, I want to list/search staff profiles, so that I can see the details, edit, and remove staff profiles.

Acceptance Criteria:

- Admins can search staff profiles by attributes such as name, email, or specialization.

- The system displays search results in a list view with key staff information (name, email, specialization).

- Admins can select a profile from the list to view, edit, or deactivate.

- The search results are paginated, and filters are available for refining the search results.

5.1.16 As a Doctor, I want to request an operation, so that the Patient has access to the necessary healthcare.

Acceptance Criteria:

- Doctors can create an operation request by selecting the patient, operation type, priority, and suggested deadline.

- The system validates that the **operation type** matches the doctor's specialization.

- The operation request includes:

  - Patient ID

  - Doctor ID

  - Operation Type

  - Deadline

  - Priority

- The system confirms successful submission of the operation request and logs the request in the patient's medical history.

5.1.17 As a Doctor, I want to update an operation requisition, so that the Patient has access to the necessary healthcare.

Acceptance Criteria:

- Doctors can update operation requests they created (e.g., change the deadline or priority).

- The system checks that only the requesting doctor can update the operation request.

- The system logs all updates to the operation request (e.g., changes to priority or deadline).

- Updated requests are reflected immediately in the system and notify the **Planning Module** of any changes.

5.1.18 As a Doctor, I want to remove an operation requisition, so that the healthcare activities are provided as necessary.

Acceptance Criteria:

- Doctors can delete operation requests they created if the operation has not yet been scheduled.

- A confirmation prompt is displayed before deletion.

- Once deleted, the operation request is removed from the patient's medical record and cannot be recovered.

- The system notifies the **Planning Module** and updates any schedules that were relying on this request.

5.1.19 As a Doctor, I want to list/search operation requisitions, so that I see the details, edit, and remove operation requisitions.

Acceptance Criteria:

- Doctors can search operation requests by patient name, operation type, priority, and status.

- The system displays a list of operation requests in a searchable and filterable view.

- Each entry in the list includes operation request details (e.g., patient name, operation type, status).

- Doctors can select an operation request to view, update, or delete it.

5.1.20 As an Admin, I want to add new types of operations, so that I can reflect on the available medical procedures in the system.

Acceptance Criteria:

- Admins can add new operation types with attributes like:

  - Operation Name

  - Required Staff by Specialization

  - Estimated Duration

- The system validates that the operation name is unique.

- The system logs the creation of new operation types and makes them available for scheduling immediately.

5.1.21 As an Admin, I want to edit existing operation types, so that I can update or correct information about the procedure.

Acceptance Criteria:

- Admins can search for and select an existing operation type to edit.

- Editable fields include operation name, required staff by specialization, and estimated duration.

- Changes are reflected in the system immediately for future operation requests.

- Historical data is maintained, but new operation requests will use the updated operation type information.

5.1.22 As an Admin, I want to remove obsolete or no longer performed operation types, so that the system stays current with hospital practices.

Acceptance Criteria:

- Admins can search for and mark operation types as inactive (rather than deleting them) to preserve historical records.

- Inactive operation types are no longer available for future scheduling but remain in historical data.

- A confirmation prompt is shown before deactivating an operation type.

5.1.23 As an Admin, I want to list/search operation types, so that I can see the details, edit, and remove operation types.

Acceptance Criteria:

- Admins can search and filter operation types by name, specialization, or status (active/inactive).

- The system displays operation types in a searchable list with attributes such as name, required staff, and estimated duration.

- Admins can select an operation type to view, edit, or deactivate it.

6 Sprint 2

6.1 Integration

The user interface of the system will integrate the interaction of the different modules in a seamless way.

6.1.1 As user, I want to have an integrated UI for all modules of the system so that I don't need to switch between application urls

6.1.2 As user I want the application menu to adjust according to my role so that it only presents me the options I may access

6.1.3 As healthcare staff I want the information show on the 3D visualization module about room availability is in sync with the schedule that was generated by the planning module

6.1.4 As Admin I want the information about healthcare staff, operation types, and operation requests used in the planning module is in sync with the information entered in the backoffice module

6.1.5 As Admin I want the information about staff's availability and operation schedule is in sync with the plan generated by the planning module

6.2 Backoffice module

The scope of the sprint 2 for the backoffice module is mainly to implement the user interface for each requirement of sprint 1.

6.2.1 As a Patient, I want to register for the healthcare application, so that I can create a user profile and book appointments online.

6.2.2 As a Patient, I want to update my user profile, so that I can change my personal details and preferences.

6.2.3 As a Patient, I want to delete my account and all associated data, so that I can exercise my right to be forgotten as per GDPR.

6.2.4 As a (non-authenticated) Backoffice User, I want to log in to the system using my credentials, so that I can access the backoffice features according to my assigned role.

6.2.5 As a Patient, I want to log in to the healthcare system, so that I can access my appointments, medical records, and other features securely.

6.2.6 As an Admin, I want to create a new patient profile, so that I can register their personal details and medical history.

6.2.7 As an Admin, I want to edit an existing patient profile, so that I can update their information when needed.

6.2.8 As an Admin, I want to delete a patient profile, so that I can remove patients who are no longer under care

6.2.9 As an Admin, I want to list/search patient profiles by different attributes, so that I can view the details, edit, and remove patient profiles.

6.2.10 As an Admin, I want to create a new staff profile, so that I can add them to the hospital's roster.

6.2.11 As an Admin, I want to edit a staff's profile, so that I can update their information.

6.2.12 As an Admin, I want to deactivate a staff profile, so that I can remove them from the hospital's active roster without losing their historical data.

6.2.13 15. As an Admin, I want to list/search staff profiles, so that I can see the details, edit, and remove staff profiles.

6.2.14 As a Doctor, I want to request an operation, so that the Patient has access to the necessary healthcare.

6.2.15 As a Doctor, I want to update an operation requisition, so that the Patient has access to the necessary healthcare.

6.2.16 As a Doctor, I want to remove an operation requisition, so that the healthcare activities are provided as necessary.

6.2.17 As a Doctor, I want to list/search operation requisitions, so that I see the details, edit, and remove operation requisitions.

6.2.18 As an Admin, I want to add new types of operations, so that I can reflect on the available medical procedures in the system.

6.2.19 As an Admin, I want to edit existing operation types, so that I can update or correct information about the procedure.

6.2.20 As an Admin, I want to remove obsolete or no longer performed operation types, so that the system stays current with hospital practices.

6.2.21 As an Admin, I want to list/search operation types, so that I can see the details, edit, and remove operation types.

6.3 Planning module

6.3.1 As an Admin, I want to obtain the better scheduling of a set of operations (surgeries) in a certain operation room in a specific day.

The better scheduling is considered as the sequence of operations that finishes early. Note that surgeries have constraints (e.g. number of doctors or other staff), namely concerning the time availability of staff during the day. The approach may be generating all surgeries' sequences and select the better, and this is possible till a certain dimension (number of surgeries).

The user must have a user interface to start the process (enter any additional parameters the planning algorithm needs, e.g., room number, date). The system will then generate the plan and show it to the user on the screen. It is acceptable that the UI blocks while waiting for the planning module response.

6.3.2 As an Admin, I want to know till what dimension in terms of number of surgeries is possible to ask for the better solution.

Perform a complexity analysis of the problem to understand to which dimension it is feasible to ask for the better solution. Document your results and findings.

6.3.3 As an Admin, I want to obtain a good schedule, not necessarily the better, in useful time to be adopted.

The system generates a "good" (non-optimal but efficient) schedule using heuristics or informed methods (e.g., greedy algorithms, rule-based scheduling).

The system prioritizes generating a schedule that is close to optimal but does so within a reasonable time frame (e.g., under 30 seconds).

The user must have a user interface to start the process (enter any additional parameters the planning algorithm needs, e.g., room number, date, which heuristic to use). The system will then generate the plan and show it to the user on the screen. It is acceptable that the UI blocks while waiting for the planning module response.

6.4 Business continuity module[2]

6.4.1 As system administrator, I want the deployment of one of the RFP modules in a DEI VM to be systematic, validating it on a scheduled basis with the test plan.

6.4.2 As system administrator, I only want clients on the DEI's internal network (wired or via VPN) to be able to access the solution.

6.4.3 As system administrator, I want the clients listed in the requirement 6.3.2 to be able to be defined by simply changing a text file.

6.4.4 As an administrator, I want to identify and quantify the risks involved in the recommended solution.

6.4.5 As system administrator, I want to define the MBCO (Minimum Business Continuity Objective) to propose to stakeholders

6.4.6 As system administrator, I want a backup strategy to be proposed, justified and implemented that minimizes RPO (Recovery Point Objective) and WRT (Work Recovery Time).

6.4.7 As system administrator I want to define a public folder for all users registered on the system, where they can read whatever is placed there

6.4.8 As system administrator I want to get users with more than 3 incorrect accesses attempts

[2] Each student must perform 2 requirements.

6.5 3D visualization module

6.5.1 As a healthcare staff member, I want to see a 3D representation of the hospital/clinic floor. Its description should be imported from a JSON (JavaScript Object Notation) formatted file. The floor must consist of several surgical rooms. Each room must be enclosed by walls and include a door and a surgical table. There should be no representation of the ceiling. If a room is being used at any given time, a 3D model of a human body should be lying on the table. Models can either be created or imported.

6.5.2 As a healthcare staff member, I want to see appropriate textures (that is, suitable for use in representing a hospital or clinic) mapped onto the floor, walls, and so on.

6.5.3 As a healthcare staff member, I want to see the hospital/clinic floor illuminated with ambient and directional light.

6.5.4 As a healthcare staff member, I want to control the camera with the mouse.

Controls can either be created or imported (e.g., three.js addon OrbitControls or equivalent):

- Left button: unused for now. It will be defined in the next sprint
- Right button: orbit
- Wheel: Zoom or dolly

6.6 GDPR Module

The GDPR module will ensure that the team fully understands the scope of its intervention and its impact in personal data.

6.6.1 As the entity that will implement the technical solution, I want to be sure that the team has good knowledge of the project, how it may affect patients' personal data and if the processing is done according to the law.

Acceptance Criteria:

-The team must explain the project and its functionalities.

- The team must be able to identify personal data that may be processed by the prototype.

- The team must be able to identify the processing to which the data will be subjected.

- The team must identify the lawful basis(s) for processing personal data.

6.6.2 As a System, I want to notify both users and the responsible authority in case of a data breach, so that I comply with GDPR's breach notification requirements.

Acceptance Criteria:

- The system automatically detects potential data breaches and immediately notifies both the affected users and the relevant GDPR authority.

- Breach notifications to users include:

  - Details of the breach (e.g., what data was compromised).

  - Steps being taken to mitigate the breach.

  - Recommendations for users (e.g., changing passwords, monitoring for suspicious activity).

- Notifications to the GDPR authority include detailed logs of the breach and actions taken.

- Breach notifications are sent within the legally required timeframe (e.g., 72 hours).

- The system logs all breach notifications and subsequent actions taken for auditing and compliance purposes.