Volere

Requirements Specification Template

Edition 16—2012

by James Robertson & Suzanne Robertson principals of the Atlantic Systems Guild

The Volere Requirements Specification Template is intended for use as a basis for your requirements specifications. The template provides sections for each of the requirements types appropriate to today's software systems. You may download the template from the Volere site and adapt it to your requirements gathering process and requirements tool. The template can be used with Requisite, DOORS, Caliber RM, IRqA and other popular tools see http://www.volere.co.uk/tools.htm

Contents

Project Drivers

Project Constraints

Functional Requirements

Non-functional Requirements

Project Issues

The Volere Requirements Knowledge Model (included with the download of Version 16 of this template) shows the formal structure and cross references between the components in the above table of contents.

Volere

Volere is the result of many years of practice, consulting, and research in requirements engineering and business analysis. We have packaged our experience in the form of a generic requirements process, requirements training, requirements consultancy, requirements audits, a variety of downloadable guides and articles, a requirements knowledge model and this requirements template. We also provide requirements specification-writing services.

The first edition of the Volere Requirements Specification Template was released in 1995. Since then, organizations from all over the world have saved time and money by using the template as the basis for discovering, organizing, and communicating their requirements.

The Volere web site www.volere.co.uk contains articles about the Volere techniques, experiences of Volere users and case studies, requirements tools, and other information useful to requirements practitioners.

The Volere requirements process is described in the book *Mastering the Requirements Process—Second Edition* by Suzanne Robertson and James Robertson, Addison-Wesley, 2006. ISBN 0-321-41949-9

For more about managing requirements see *Requirements Led Project Management* by Suzanne Robertson and James Robertson, Addison-Wesley, 2005.
ISBN 0-321-65904-X

Updates to this template and instructions for downloading are available at http://www.volere.co.uk

Public seminars on Volere are run on a regular basis in Europe, the United States, Australia, and New Zealand. For a schedule of courses, refer to www.volere.co.uk.

Requirements Types

For ease of use, we have found it convenient to think of requirements as belonging to a type. There are two reasons for the type: as an aid to finding the requirements, to be able to group the requirements that are relevant to a specific expert specialty.

Functional requirements are the fundamental or essential subject matter of the product. They describe what the product has to do or what processing actions it must take.

Non-functional requirements are the properties that the functions must have, such as performance and usability. Do not be deterred by the unfortunate name for this kind of requirements, they are as important as the functional requirements for the product's success.

Project constraints are restrictions on the product due to the budget or the time available to build the product.

Design constraints impose restrictions on how the product must be designed. For example, it might have to be implemented in the hand-held device being given to major customers, or it might have to use the existing servers and desktop computers, or any other hardware, software, or business practice.

Project drivers are the business-related forces. For example, the purpose of the project is a project driver, as are all of the stakeholders—each for different reasons.

Project issues define the conditions under which the project will be done. Our reason for including them as part of the requirements is to present a coherent picture of all factors that contribute to the success or failure of the project and to illustrate how managers can use requirements as input when managing a project.

Testing Requirements

The Volere philosophy is to start testing requirements as soon as you start writing them. You make a requirement testable by adding its fit criterion. This fit criterion measures the requirement, making it possible to determine whether a given solution fits the requirement. If a fit criterion cannot be found for a requirement, then the requirement is either ambiguous or poorly understood. All requirements can be measured, and all should carry a fit criterion.

Atomic Requirements Shell

Requirement Card 1 — Patient / Doctor Account Update

Requirement #: UC-01

Requirement Type: Functional

Event/BUC/PUC #: Patient / Doctor Account Update

Description:

The system shall allow patients and doctors to update their personal or professional account information securely.

Rationale:

Correct account data ensures accurate communication, record management, and identity verification.

Originator:

Patient, Doctor stakeholder groups.

Fit Criterion:

User submits changes → system validates → updates database → confirms success or displays validation errors.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Priority: High

Conflicts: None

Supporting Materials:

Profile UI mockups, workflow diagram.

History:

D3 Version 1

Requirement Card 2 — View EHR

Requirement #: UC-02

Requirement Type: Functional

Event/BUC/PUC #: View EHR

Description:

The system shall allow authorized users to view a patient's complete electronic health record.

Rationale:

Accurate clinical decisions depend on immediate access to medical history.

Originator:

Doctor stakeholder group.

Fit Criterion:

System loads and displays all EHR components (history, prescriptions, lab results) within 3 seconds.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Priority: High

Conflicts: None

Supporting Materials:

EHR UI mockups, data model.

History:

D3 Version 1

Requirement Card 3 — Send Prescription

Requirement #: UC-03

Requirement Type: Functional

Event/BUC/PUC #: Send Rx

Description:

The system shall allow doctors to create and send electronic prescriptions to pharmacies.

Rationale:

Reduces medical errors and speeds up pharmacy processing.

Originator:

Doctor stakeholder group.

Fit Criterion:

Prescription created → validated → digitally signed → visible in pharmacy dashboard.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Priority: High

Conflicts: None

Supporting Materials:

Rx workflow, pharmacy UI.

History:

D3 Version 1

Requirement Card 4 — Edit Medical Record

Requirement #: UC-04

Requirement Type: Functional

Event/BUC/PUC #: Edit Medical Record

Description:

The system shall allow doctors to edit patient medical records with automatic change logging.

Rationale:

Ensures accuracy and traceability of clinical information.

Originator:

Doctor stakeholder group.

Fit Criterion:

Record edited → validated → change logged → updated record shown instantly.

Customer Satisfaction: 4

Customer Dissatisfaction: 5

Priority: High

Conflicts: None

Supporting Materials:

Audit trail design, EHR wireframes.

History:

D3 Version 1

Requirement Card 5 — Book Appointment

Requirement #: UC-05

Requirement Type: Functional

Event/BUC/PUC #: Book Appointment

Description:

The system shall allow patients to book appointments based on real-time doctor availability.

Rationale:

Reduces manual scheduling load and improves patient access to care.

Originator:

Patient stakeholder group.

Fit Criterion:

Patient selects doctor + slot → system verifies availability → creates appointment → notifies both parties.

Customer Satisfaction: 5

Customer Dissatisfaction: 4

Priority: High

Conflicts: None

Supporting Materials:

Appointment sequence diagram, UI prototype.

History:

D3 Version 1

Requirement Card 6 — Create Insurance Claim

Requirement #: UC-06

Requirement Type: Functional

Event/BUC/PUC #: Create Insurance Claim

Description:

The system shall allow patients or admins to create insurance claims for medical visits or procedures.

Rationale:

Streamlines insurance processing and reduces administrative overhead.

Originator:

Administrator stakeholder group.

Fit Criterion:

User submits claim → mandatory fields verified → claim enters validation workflow.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Priority: Medium

Conflicts: None

Supporting Materials:

Insurance workflow diagram.

History:

D3 Version 1

Requirement Card 7 — Manage Roles

Requirement #: UC-07

Requirement Type: Functional

Event/BUC/PUC #: Manage Roles

Description:

The system shall allow administrators to assign and modify user roles and permissions.

Rationale:

Controlled access ensures system security and compliance.

Originator:

Administrator stakeholder group.

Fit Criterion:

Admin selects user → assigns/changes role → system updates permission set → audit entry recorded.

Customer Satisfaction: 4

Customer Dissatisfaction: 5

Priority: High

Conflicts: None

Supporting Materials:

Role hierarchy diagram, access control list.

History:

D3 Version 1

Requirement Card 8 — Create Billing Record

Requirement #: UC-08

Requirement Type: Functional

Event/BUC/PUC #: Create Billing Record

Description:

The system shall allow admins to create billing records for patient services.

Rationale:

Billing automation reduces errors and speeds up financial processing.

Originator:

Administrator stakeholder group.

Fit Criterion:

Service is selected → system calculates cost → billing record created → linked to patient account.

Customer Satisfaction: 4

Customer Dissatisfaction: 3

Priority: Medium

Conflicts: None

Supporting Materials:

Billing workflow chart.

History:

D3 Version 1

Requirement Card 9 — Verify Prescription

Requirement #: UC-09

Requirement Type: Included Use Case

Event/BUC/PUC #: Verify Prescription

Description:

The system shall validate prescription details before sending to a pharmacy.

Rationale:

Prevents invalid or incomplete prescriptions.

Originator:

Doctor stakeholder group.

Fit Criterion:

System checks dosage, conflicts, drug availability, and patient identity before approval.

Customer Satisfaction: 5

Customer Dissatisfaction: 5

Priority: High

Conflicts: None

Supporting Materials:

Prescription validation rules.

History:

D3 Version 1

Requirement Card 10 — Modify / Cancel Prescription

Requirement #: UC-10

Requirement Type: Extended Use Case

Event/BUC/PUC #: Modify / Cancel Prescription

Description:

The system shall allow doctors to modify or cancel existing prescriptions.

Rationale:

Medication changes are common and must be reflected quickly.

Originator:

Doctor stakeholder group.

Fit Criterion:

Doctor updates medication → system logs change → updated prescription replaces previous version → pharmacy notified.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Priority: Medium

Conflicts: None

Supporting Materials:

Rx modification workflow.

History:

D3 Version 1

Requirement Card 11 — Add Record Entry

Requirement #: UC-11

Requirement Type: Included Use Case

Event/BUC/PUC #: Add Record Entry

Description:

The system shall allow doctors to add new medical record entries for a patient.

Rationale:

New diagnoses, test results, or notes must be appended promptly and accurately.

Originator:

Doctor stakeholder group.

Fit Criterion:

Record entry created → timestamped → stored → visible in patient's EHR.

Customer Satisfaction: 5

Customer Dissatisfaction: 4

Priority: High

Conflicts: None

Supporting Materials:

EHR activity diagram.

History:

D3 Version 1

Requirement Card 12 — Reschedule / Cancel Appointment

Requirement #: UC-12

Requirement Type: Extended Use Case

Event/BUC/PUC #: Reschedule / Cancel Appointment

Description:

The system shall allow patients to reschedule or cancel their appointments.

Rationale:

Flexibility increases patient satisfaction and reduces no-shows.

Originator:

Patient stakeholder group.

Fit Criterion:

Patient selects a new slot or cancels → system updates schedule → both parties notified.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Priority: Medium

Conflicts: None

Supporting Materials:

Appointment lifecycle model.

History:

D3 Version 1

Requirement Card 13 — Validate Claim

Requirement #: UC-13

Requirement Type: Included Use Case

Event/BUC/PUC #: Validate Claim

Description:

The system shall validate insurance claims before approval.

Rationale:

Reduces fraudulent or incomplete claims.

Originator:

Administrator stakeholder group.

Fit Criterion:

Claim is checked for completeness, correctness, and policy coverage before approval.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Priority: Medium

Conflicts: None

Supporting Materials:

Insurance validation rules.

History:

D3 Version 1

Requirement Card 14 — Generate Report

Requirement #: UC-14

Requirement Type: Included Use Case

Event/BUC/PUC #: Generate Report

Description:

The system shall generate administrative and medical reports.

Rationale:

Reports support analytics, monitoring, and decision-making.

Originator:

Administrator stakeholder group.

Fit Criterion:

Admin selects report type → system generates PDF or dashboard → loads within 5 seconds.

Customer Satisfaction: 4

Customer Dissatisfaction: 3

Priority: Medium

Conflicts: None

Supporting Materials:

Report templates, sample outputs.

History:

D3 Version 1

Requirement Card 15 — Dispense Prescription

Requirement #: UC-15

Requirement Type: Included Use Case

Event/BUC/PUC #: Dispense Prescription

Description:

The system shall allow pharmacies to dispense prescriptions sent by doctors.

Rationale:

Ensures a complete prescription lifecycle and medical safety.

Originator:

Pharmacy stakeholder group.

Fit Criterion:

Pharmacy receives Rx → verifies → marks as dispensed → patient notified.

Customer Satisfaction: 4

Customer Dissatisfaction: 4

Priority: Medium

Conflicts: None

Supporting Materials:

Pharmacy module workflow.

History:

D3 Version 1

Requirement Card 16 — Create Lab Result

Requirement #: UC-16

Requirement Type: Included Use Case

Event/BUC/PUC #: Create Lab Result

Description:

The system shall allow medical staff to create and attach lab results to a patient’s record.

Rationale:

Lab results are critical for diagnosis and must be stored digitally.

Originator:

Doctor stakeholder group.

Fit Criterion:

Lab result entered → validated → attached to patient EHR → notification sent to patient and doctor.

Customer Satisfaction: 5

Customer Dissatisfaction: 4

Priority: High

Conflicts: None

Supporting Materials:

Lab entry workflow, EHR design.

History:

D3 Version 1

1. The Purpose of the Project

The first section of the template deals with the fundamental reason your client asked you to build a new product. That is, it describes the business problem the client faces and explains how the product is intended to solve the problem.

1a. The User Business or Background of the Project Effort

Content

Medalyze operates within the healthcare service ecosystem, where hospitals, clinics, doctors, pharmacies, and patients rely on fast and accurate communication. The current environment is fragmented—appointments, medical records, prescriptions, and billing are typically managed by separate, unconnected systems. This causes redundant data entry, delays, miscommunication, and inconsistent access to health information.

The Medalyze platform is being developed to centralize these operations, allowing patients to book appointments, access their records, and receive notifications, while healthcare professionals manage consultations, prescriptions, and billing from one unified system.

Motivation

Without centralization, healthcare institutions face growing administrative workloads, bottlenecks in service delivery, and avoidable errors. The project's justification comes from the need to improve efficiency, data accuracy, and patient experience. A unified system directly addresses these needs and supports institutions transitioning to modern digital workflows.

Considerations

The underlying business problem is serious: inefficient coordination between healthcare entities directly impacts patient satisfaction and delays medical decisions. The project also represents a significant opportunity to modernize healthcare management, reduce operational overhead, and provide new analytics capabilities that currently do not exist in most facilities.

Form

A textual description is sufficient to capture the core background. Supporting artifacts include existing workflow diagrams, prototypes of appointment and record pages, system overview diagrams, and organization charts showing the flow of medical data between stakeholders.

1b. Goals of the Project

Content

The goal of Medalyze is to provide a unified, centralized healthcare management platform that streamlines appointments, electronic health records, prescriptions, notifications, and billing for patients, doctors, and administrative staff. The system should improve coordination, reduce redundant work, and enhance overall healthcare service quality.

Motivation

Clear, measurable goals ensure the development remains aligned with the project’s original purpose. As new ideas emerge during implementation, these goals act as anchors to prevent unnecessary scope drift and keep stakeholders focused on the intended benefits.

Examples

- Provide patients with immediate access to appointments, records, and prescriptions.
- Give doctors accurate, up-to-date patient information for better decision-making.
- Improve hospital and clinic efficiency through centralized data handling.

Measurement

The project’s success will be measured by quantifiable improvements such as:
- Reduced appointment processing time (e.g., time from booking to confirmation).
- Decrease in duplicated or inconsistent patient data.
- Increased usage of the patient portal and engagement metrics.
- Faster prescription turnaround from doctor submission to pharmacy processing.

Form

Purpose: Centralize healthcare operations in one digital platform to eliminate fragmentation.
Advantage: Faster service delivery, higher data accuracy, and improved patient satisfaction across all participating entities.
Measurement: Track reductions in processing time, frequency of data redundancies, portal engagement rates, and prescription fulfillment speed.

2. The Stakeholders

This section describes the stakeholders—the people who have an interest in the product. It is worth your while to spend enough time to accurately determine and describe these people, as the penalty for not knowing who they are can be very high.

2a. The Client

Content

The primary client for Medalyze is the **Healthcare Information Systems Division** within the partnering medical institution. This includes the supervising academic body overseeing the development of the Medalyze conceptual prototype—represented by **Dr. Amal Elgammal**, who acts as the project's sponsor. She provides direction, validation, and approval for requirements and deliv

Motivation

The client is responsible for the acceptance of the final product. They ensure the system meets the needs of healthcare organizations, supports efficient operations, and aligns with strategic goals for digital transformation. Their involvement guarantees that Medalyze's functions and scope remain realistic, valuable, and appropriate for the targeted healthcare environment.

Considerations

For concept-level systems like Medalyze, the client may represent both the business oversight layer and internal stakeholders who expect the solution to improve operational efficiency. In a production scenario, the client could be hospital management, a health-tech organization, or a healthcare IT department requesting a unified management platform.

Form

The client is positioned at the top of the project oversight structure as the primary approver of deliverables.

- The client is responsible for decisions related to project scope, approval of use cases, acceptance of architecture models, and validation of system requirements.

- Review checkpoints include:

- Requirement confirmation meetings
- Approval of use-case models
- Approval of prototype interface layouts
- Final acceptance of the deliverable package

2b. The Customer

Content

The primary customers for Medalyze are healthcare institutions and administrators who are responsible for approving and adopting the platform within their organization. This includes hospital managers, clinic directors, and pharmacy managers who oversee operational efficiency and patient satisfaction. They decide whether Medalyze meets their needs for appointment management, EHR integration, and prescription processing.

Motivation

The customer is responsible for approving the system and ensuring it aligns with organizational objectives such as efficiency, data consistency, and improved patient experience. Correct requirements can only be gathered by understanding the aspirations and operational challenges of these decision-makers.

Form

- Decisions on overall system adoption.

- Review and approval of design prototypes for UI/UX and workflow.

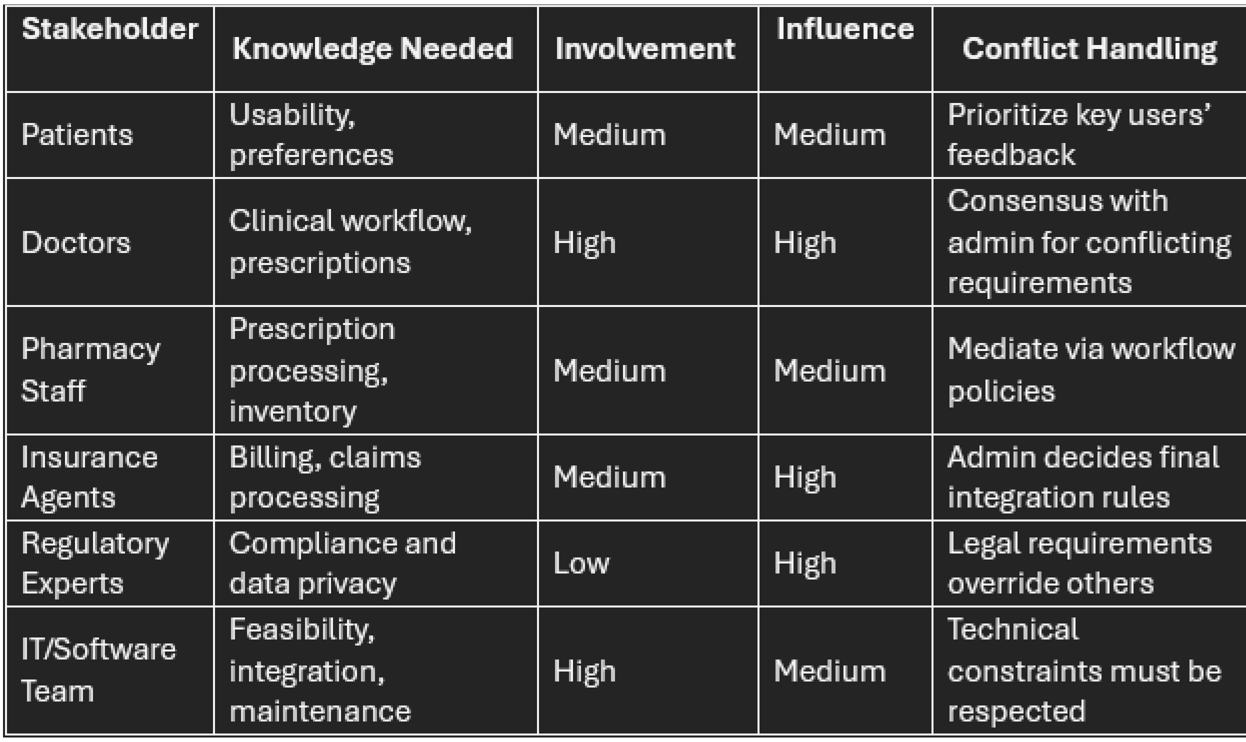
- Feedback on functional modules: Appointment Scheduling, EHR, Prescription Management, Billing.

- Monitoring progress indicators such as mockups, simulated workflows, and user interface prototypes.

2c. Other Stakeholders

Content

Below is the list of additional stakeholders who directly influence or are affected by the Medalyze system:



Stakeholder	Knowledge Needed	Involvement	Influence	Conflict Handling
Patients	Usability, preferences	Medium	Medium	Prioritize key users' feedback
Doctors	Clinical workflow, prescriptions	High	High	Consensus with admin for conflicting requirements
Pharmacy Staff	Prescription processing, inventory	Medium	Medium	Mediate via workflow policies
Insurance Agents	Billing, claims processing	Medium	High	Admin decides final integration rules
Regulatory Experts	Compliance and data privacy	Low	High	Legal requirements override others
IT/Software Team	Feasibility, integration, maintenance	High	Medium	Technical constraints must be respected

Motivation

Recognizing all stakeholders ensures that no critical requirements are missed and the system works for everyone involved.

Form

- Stakeholder map detailing roles, responsibilities, knowledge contribution, and influence level.

- Conflict resolution protocols for overlapping interests (e.g., patient data privacy vs. administrative reporting).

2d. The Hands-On Users of the Product

Content

- **Patients:**
    - Role: Book appointments, view health records, receive notifications.

    - Subject matter experience: Novice.
    - Technological experience: Journeyman.
    - Other characteristics: Wide age range, varying tech comfort, mobile and desktop access.
- **Doctors:**
    - Role: Manage patient records, issue prescriptions, review lab results.
    - Subject matter experience: Master.
    - Technological experience: Journeyman.
    - Other characteristics: Limited time, require efficiency and clarity in UI, mostly desktop usage.
- **Pharmacy Staff:**
    - Role: Process electronic prescriptions, update inventory, notify patients.
    - Subject matter experience: Journeyman.
    - Technological experience: Novice to Journeyman.
    - Other characteristics: Shift-based work, moderate computer skills, mobile or desktop.
- **Administrators:**
    - Role: Monitor system analytics, approve workflows, manage user accounts.
    - Subject matter experience: Master.
    - Technological experience: Journeyman.
    - Other characteristics: High-level access, data analysis needs, desktop-based.

Motivation

Defining users allows the system to meet usability, accessibility, and efficiency requirements.

Examples

Users can come from wide variety of (sometimes unexpected) sources. Consider the possibility of your users being clerical staff, shop workers, managers, highly trained operators, the general public, casual users, passersby, illiterate people, tradesmen, students, test engineers, foreigners, children, lawyers, remote users, people using the system over the telephone or an Internet connection, emergency workers, and so on.

Form

A simple list of user characteristics for each role:

- **Patient – Sara (Persona)**
  - Subject matter experience: Novice
  - Technological experience: Journeyman
  - Other characteristics: Mobile & desktop access, varying tech comfort, wide age range
- **Doctor – Ahmed (Persona)**
  - Subject matter experience: Master
  - Technological experience: Journeyman
  - Other characteristics: Time-constrained, mostly desktop user, requires efficiency
- **Pharmacy Staff**
  - Subject matter experience: Journeyman
  - Technological experience: Novice to Journeyman
  - Other characteristics: Shift-based work, moderate computer skills, mobile or desktop use
- **Administrator – Mohamed (Persona)**
  - Subject matter experience: Master
  - Technological experience: Journeyman
  - Other characteristics: Desktop-based, high-level access, data analysis responsibilities

2e. Personas

Content

- **Ahmed, 35, General Practitioner**

    - Lives in Cairo, married with 2 children. Hobbies: football and reading. Comfortable with technology, needs efficient EHR and prescription tools.
- **Sara, 28, Patient**
    - University student in Alexandria. Hobbies: painting, social media. Uses mobile devices heavily, prefers reminders and simple navigation for appointments and records.
- **Mohamed, 42, Hospital Administrator**
    - Lives in Giza, married. Hobbies: chess and travel. Requires dashboards for analytics and operational decision-making, mostly desktop user.

Motivation

Personas help ensure requirements meet the expectations of real user types with specific goals, frustrations, and workflows.

Form

- Each persona's information will be recorded in a **profile document** listing: name, age, role, location, hobbies, technology habits, and how they interact with the system.
- The profiles serve as **reference material for the project team**, helping everyone understand user needs, goals, and workflows.
- **Optional visual aids** (icons, sketches, or placeholders) can be added, but no real photos are required.
- Profiles can be displayed on cards, slides, or posters during team meetings to **keep the personas "alive"** and ensure design decisions stay user-focused.
- They may also be summarized in a **storyboard or table** highlighting persona characteristics and key interactions with the system.

2f. Priorities Assigned to Users

Content

This section assigns a priority to each user category, indicating their importance to the system's success.

Motivation

Prioritizing users ensures the most critical requirements drive the system design.

Form

- **Key Users** – Doctors, Administrators
    - Critical to operational success
    - Their requirements take precedence over others
- **Secondary Users** – Patients
    - Frequent users, but individual preferences have less impact on overall success
- **Unimportant Users** – External auditors, temporary visitors
    - Infrequent access, low priority, minimal influence on design

2g. User Participation

Content

This section defines the expected participation of users in providing requirements and feedback during development.

Motivation

Clear user participation ensures requirements are complete and accurate. Without specifying time and type of input, projects risk missing key usability or workflow information.

Form

- **Doctors**
    - Contribution: Clinical workflow input, EHR mockup review
    - Estimated participation: 3–5 hours per week
- **Patients**
    - Contribution: Usability feedback, survey responses
    - Estimated participation: 1–2 hours per week

- **Administrators**
  - Contribution: Dashboard and workflow review
  - Estimated participation: 2–3 hours per week

2h. Maintenance Users and Service Technicians

Content

Maintenance users are responsible for keeping the system running and performing updates or bug fixes.

Motivation

Defining maintenance users helps identify requirements related to system maintenance, reliability, and future changes that might otherwise be missed.

Form

- **IT Personnel**
  - Responsibilities: System updates, backups, troubleshooting
  - Experience: Journeyman to Master in system maintenance
- **Software Developers**
  - Responsibilities: Architecture updates, bug fixes, feature enhancements
  - Experience: Master in system development and integration

3. Mandated Constraints

This section describes constraints on the eventual design of the product.

Constraints are global—they are factors that apply to the entire product. The product must be built within the stated constraints. Often you know about the constraints, or they are mandated before the project gets under way. They are probably determined by management and are worth considering carefully—they restrict what you can do and so shape the product. Constraints, like other types of requirements have a description, rationale, and fit criterion, and generally are written in the same format as functional and non-functional requirements.

3a. Solution Constraints

Content

- Medalyze shall use **React for the frontend** (v18+) and **Scala with http4s** for backend services.
- Database must be **PostgreSQL** for structured EHR and appointment data.
- All user authentication shall comply with **OAuth2** standards.
- Notifications shall be delivered via **email and SMS APIs** already approved by the hospital.

Motivation

These solution constraints are non-negotiable because the client's IT infrastructure, existing expertise, and licensing agreements dictate the technology stack. Using alternative technologies would make deployment or integration impossible.

Form

- Each constraint will be included in the requirements spreadsheet as an atomic requirement with:
  - **Description:** what the constraint is
  - **Rationale:** why it is non-negotiable
  - **Fit criterion:** how compliance will be tested
- Optional diagrams may illustrate system architecture showing how these mandated technologies interact.
- **Example:**
  - **Description:** The backend services shall be implemented in Scala using http4s.
  - **Rationale:** The hospital IT team is experienced with Scala and http4s, and integration with existing services requires it.
  - **Fit criterion:** All endpoints shall be implemented in http4s and pass functional testing.

3b. Implementation Environment of the Current System

Content

- The system will operate on **hospital/clinic servers** running Linux or Windows Server.
- Client devices include desktops (doctors and admins) and mobile devices (patients).
- Printers and barcode scanners are used in pharmacies and billing stations.
- Existing EHR modules and insurance claim systems are present as adjacent systems.

Motivation

Understanding the operational environment ensures that Medalyze interacts correctly with all hardware, software, and organizational systems already in use.

3c. Partner or Collaborative Applications

Content

- Medalyze collaborates with **insurance claim systems**, **pharmacy management systems**, and **SMS/email delivery APIs**.
- Existing EHR modules in hospitals must be integrated for seamless record access.

Motivation

Documenting these partner applications highlights design constraints and integration challenges, ensuring compatibility and smooth data flow.

3d. Off-the-Shelf Software

Content

- Medalyze will use the following OTS software:
  - **PostgreSQL** for the database.

    - **Twilio API** for SMS notifications.
    - **SendGrid API** for email notifications.
    - **Bootstrap/Tailwind CSS** for UI components.
- Any hardware devices such as barcode scanners or printers in pharmacies are also considered OTS components.

Motivation

Using these OTS products constrains the design because the system must operate within the capabilities, APIs, and interfaces of these products. Conflicting requirements must either be adjusted or handled through custom integration.

3e. Anticipated Workplace Environment

Content

- Patients will use Medalyze primarily from mobile devices at home or on the go.
- Doctors and administrators will use desktops in hospitals and clinics, often in busy or noisy environments.
- Pharmacies have shift-based staff using computers and barcode scanners.
- Office layouts may involve shared desks, limited space, and variable lighting.

Motivation

Understanding workplace constraints ensures the system is usable, visible, and efficient in real conditions. For example, notifications should be clearly visible and audible despite background noise.

3f. Schedule Constraints

Content

- Project must be delivered in phases aligned with the academic schedule and hospital testing periods.
- Key milestones include: requirements finalization, prototype completion, and final submission.

Motivation

Deadlines ensure that the project is usable and testable in the intended operational environment. Missing deadlines could prevent integration with hospital schedules and disrupt planned demonstrations.

Form

Written statement:

- Deadline: End of semester / end of academic term
- Reason: Project submission and hospital prototype testing schedule
- Effect of not meeting: Delayed feedback, incomplete evaluation, and lower grade/performance review

3g. Budget Constraints

Content

- The project is constrained by academic resources and available free/open-source software. No paid licenses will be purchased for this deliverable.
- Hardware such as lab PCs, mobile devices, and barcode scanners is available but limited.

Motivation

Budget limitations constrain the number of features that can be implemented and the technologies that can be used. Overambitious designs must be avoided.

Form

Written statement:

- Budget: $0–$200 (mostly academic resources and free APIs)
- Source: University-provided lab resources and free/open-source software

3h. Enterprise Constraints

Content

- Medalyze must comply with university IT policies for network access, data storage, and software usage.
- All patient data must be handled according to standard healthcare privacy guidelines.
- The system must run on the university lab hardware and devices used in partner clinics.

Motivation

Enterprise requirements may seem arbitrary but are mandatory. Compliance ensures system deployment within the organization and adherence to legal and strategic policies.

4. Naming Conventions and Terminology

It has been our experience that all projects have their own unique vocabulary usually containing a variety of acronyms and abbreviations. Failure to understand this projectspecific nomenclature correctly inevitably leads to misunderstandings, hours of lost time, miscommunication between team members, and ultimately poor-quality specifications.

4a. Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

Content

This section defines the terminology and acronyms used in Medalyze to avoid misunderstandings between team members and stakeholders:

- **EHR (Electronic Health Record):** Digital version of a patient's medical history, including diagnoses, lab results, and prescriptions.
- **Rx:** Prescription issued by a doctor for a patient's medication.
- **Appointment Slot:** A specific time period during which a patient can book a consultation with a doctor.
- **API (Application Programming Interface):** A set of protocols and tools allowing different software components to communicate.
- **Admin:** System administrator responsible for managing accounts, monitoring analytics, and overseeing operations.
- **Patient Portal:** The interface through which patients can access their appointments, records, and notifications.
- **Prescription Management System:** Module allowing doctors to issue prescriptions electronically and pharmacies to process them.
- **Notification System:** Component responsible for sending reminders about appointments, lab results, and medications via email or SMS.
- **HIPAA:** Health Insurance Portability and Accountability Act (or relevant local health privacy regulations), governing patient data privacy.
- **Pharmacy Staff:** Personnel responsible for filling prescriptions and managing pharmacy inventory.

Motivation

Defining all terms and acronyms ensures clear communication among stakeholders, prevents misinterpretation, and serves as the foundation for the data dictionary and more detailed system specifications. It reduces wasted time and helps ensure consistency across documentation, design, and development.

Form

- A glossary document containing all terms and acronyms with definitions.
- Include references to industry standards where applicable (e.g., HIPAA).
- Optionally, maintain the glossary as a living document, updated as new terms arise during the project.

5. Relevant Facts and Assumptions

Relevant facts are external factors that have an effect on the product but are not covered by other sections in the requirements template. They are not necessarily translated into requirements but could be. Relevant facts alert the developers to conditions and factors that have a bearing on the requirements.

5a. Relevant Facts

Content

- Hospitals and clinics currently use **paper-based or semi-digital appointment and record systems**, causing inefficiencies.
- Pharmacy systems vary in their degree of automation, requiring integration for digital prescriptions.
- Patients increasingly expect **mobile access** to their health records and notifications.
- Existing EHR modules and insurance claim systems are partially digital and may have limited APIs.

Motivation

Relevant facts provide context to the specification, helping developers understand the business environment and constraints, even if not directly translated into requirements. Awareness of these facts ensures that system design addresses real-world conditions.

5b. Business Rules

Content

- Doctors may only access records for patients under their care.
- Appointment slots are limited by doctors' availability and clinic capacity.
- Prescriptions must comply with local health regulations.
- Insurance claims must be validated before billing can be finalized.

Motivation

Business rules help identify triggers for system requirements and ensure compliance with legal, operational, and organizational policies. They provide context for workflows and data access rules.

5c. Assumptions

Content

- The existing workflows at hospitals and clinics are mostly stable during the project.
- Necessary software tools such as React, Scala, PostgreSQL, and Twilio/SendGrid APIs will be available.
- Partner systems (existing EHR modules, insurance claim systems, pharmacy systems) will provide functional APIs for integration.
- Users (doctors, admins, patients, pharmacy staff) will engage with the system according to their defined roles.
- No major policy or regulatory changes will occur during the development phase.

Motivation

Stating assumptions clarifies expectations, reduces misunderstandings, and highlights potential risks if an assumption proves false. They guide the development team in planning and prioritizing work.

6. The Scope of the Work

The scope of the work determines the boundaries of the business area to be studied and outlines how it fits into its environment. Once you understand the work and its constraints, you can establish the scope of the product see Section 8 of the template.

6a. The Current Situation

Content

The current healthcare workflow in many clinics and hospitals relies on a fragmented mix of manual and partially automated processes. Appointment scheduling is often handled through phone calls or simple web forms that do not integrate with doctors' schedules. Electronic Health Records (EHR) are stored in separate systems or maintained manually, limiting real-time access for patients and providers. Lab results are frequently delivered on paper or through isolated laboratory portals that do not sync with patient medical records.

Prescription processing lacks standardization; doctors may issue handwritten prescriptions or use systems that do not integrate with pharmacy software, leading to delays, errors, and inefficiencies. Billing and insurance validation processes typically involve manual review and communication between departments, increasing processing time and the risk of discrepancies.

Medalyze aims to replace or enhance these disparate processes by providing a unified, integrated platform that supports appointments, EHR access, lab result management, e-prescriptions, pharmacy processing, reporting, role management, and insurance validation. Understanding how these current processes operate is essential before introducing new system capabilities or redesigning workflows.

Motivation

If your project intends to make changes to an existing manual or automated system, you need to understand the effect of proposed changes. The study of the current situation provides the basis for understanding the effects of proposed changes and choosing the best alternatives. Business process modelling does not always lead to building software. Instead, some changes in procedures and the way roles are allocated might be the best way of making a necessary improvement.

Form

The current processes can be modeled using business activity diagrams diagrams to illustrate the workflow and dependencies among departments such as clinics, pharmacies, laboratories, and administrative units. These models capture how patients, doctors, administrators, pharmacy staff, and insurers currently interact, helping visualize where Medalyze will streamline and integrate operations.

6b. The Context of the Work

Content

For Medalyze, the work context defines the boundaries of the healthcare operations that the system aims to support and integrate. This includes all processes related to appointments, EHR access, lab result management, prescription issuance, pharmacy processing, administrative reporting, role management, and insurance validation. To build a product that fits naturally into this environment, it is essential to understand not only the Medalyze platform but also the broader healthcare ecosystem in which it operates.

Adjacent systems include laboratories, pharmacies, insurance providers, hospital information systems, patient communication channels, and external authentication services. Each of these interacts with Medalyze through data exchanges such as lab results, prescription processing updates, claim status responses, or login credentials.

Understanding these adjacent systems—how they work, what information they provide or require, and how they integrate into existing workflows—is necessary to define the scope of Medalyze and ensure seamless interoperability between all components of the healthcare environment.

Motivation

The primary motivation is to establish a clear boundary around the healthcare work processes being studied. By identifying all external actors and systems that exchange information with Medalyze, we ensure that the requirements are accurately defined and the final product aligns with real-world operations.

Without a clearly defined work context, Medalyze may overlook crucial dependencies such as lab data formats, pharmacy system constraints, or insurance verification rules. A well-defined context model prevents misalignment, ensures effective integration, and supports the correct identification of requirements for interoperability, security, and workflow consistency across clinical and administrative entities.

Examples

In the Medalyze context model, adjacent systems include:

- Laboratory Information Systems (LIS): Sends lab results and abnormality indicators to Medalyze.
- Pharmacy Management Systems: Receives e-prescriptions and returns processing status (Completed, Out of Stock).
- Insurance Provider Systems: Sends insurance validation results and claim statuses.
- Hospital Information Systems (HIS): Shares patient demographic information and doctor schedules.
- Notification Services: Sends appointment reminders, prescription updates, and alerts to patients.
- Authentication and Identity Services: Supports secure login for doctors, patients, and administrative staff.

Inputs and outputs between Medalyze and these systems include lab result files, prescription data packets, billing records, eligibility checks, scheduling information, and user identity tokens. These exchanges serve as the foundation for discovering integration requirements, defining workflows, and determining necessary interfaces.

Considerations

All names used in the Medalyze context diagram must follow the project's naming conventions and be formally defined in the data dictionary (Section 7) to ensure clarity and consistency. Without precise definitions, there is a risk of misunderstanding interfaces such as "LabResult Data," "Prescription Status Update," or "Insurance Verification Response."

Stakeholders—including clinicians, administrators, pharmacists, and IT partners—must agree on the meaning and structure of each interface. Their agreement ensures that the context model accurately reflects operational realities and can be used to guide integration specifications and system architecture decisions.

Form

The context of the work is presented as:

- A context diagram showing all inputs and outputs flowing between Medalyze and adjacent systems (laboratories, pharmacies, insurers, HIS, etc.)
  or
- A tabular list enumerating each input and output exchanged, along with the responsible adjacent system.

Each input/output will later be fully defined in the data dictionary (Section 7b) to ensure accuracy and consistency in system implementation.

6c. Work Partitioning

Content

The Medalyze work partitioning identifies all business events—real world triggers that require the Medalyze system to respond. These events include actions initiated by patients, doctors, administrators, pharmacies, laboratories, and insurers, as well as time-based events such as automatic reminders or report generation.

Each event results in a Business Use Case (BUC) that represents a discrete, self-contained segment of the overall healthcare workflow. These BUCs collectively define the total functionality of Medalyze.
For each business event, the event list includes:

- Event Name
- Triggering Input from an adjacent system (matches context diagram terminology)
- Resulting Output(s) to adjacent systems
- Brief Summary of the BUC
- Relevant Business Data Classes, based on Medalyze's domain model

This structure ensures that all interactions between Medalyze and external systems are understood before detailed requirements are created.

Motivation

Partitioning Medalyze into business events makes the analysis manageable and ensures requirements are captured logically and consistently. Each BUC can be studied independently, but all are unified through shared business data such as **Patient**, **Appointment**, **LabResult**, **Prescription**, and **BillingRecord**.

By identifying the full list of business events, the project ensures:

- No essential functionality is missed
- Each business process is traceable to a specific trigger
- Clear boundaries exist for analysis, design, and implementation
- Integration points with laboratories, pharmacies, insurers, and hospital systems are well understood

This structured approach reduces ambiguity and provides a foundation for discovering accurate, complete, and traceable requirements.

Example

Event No.	***Event Name***	***Triggering Input***	***Output(s)***	***Summary of BUC***	***Relevant Business Data***
1	*Patient Requests Appointment*	*Appointment Request (in)*	*Appointment Confirmation (out)*	*Patient chooses doctor/date; system checks availability and schedules appointment.*	*Patient, Doctor, Appointment*
2	*Doctor Updates Availability*	*Schedule Update (in)*	*Updated Schedule (out)*	*Doctor modifies availability; system refreshes booking options.*	*Doctor, Appointment*
3	*Lab Sends Test Result*	*LabResult Data (in)*	*LabResult Added Confirmation (out)*	*Laboratory submits patient test results; system stores them and flags abnormalities.*	*LabResult, MedicalRecord, Patient*
4	*Doctor Issues e-Prescription*	*Prescription Request (in)*	*Prescription File (out)*	*Doctor creates prescription; system forwards it to pharmacy.*	*Prescription, Medication, Patient, Doctor*
5	*Pharmacy Processes Prescription*	*Prescription File (in)*	*Prescription Status Update (out)*	*Pharmacy checks medication availability and marks status.*	*Prescription, PharmacyStaff, Pharmacy*
6	*Insurance Provider*	*Claim Request (in)*	*Claim Status (out)*	*System sends billing data; insurer returns*	*BillingRecord, Patient*

	Validates Claim			*approval/denial status.*	
7	*Administrator Generates Report*	*Report Request (in)*	*Utilization or Performance Report (out)*	*Admin selects report type; system compiles and outputs report.*	*Report, Administrator*
8	*System Sends Appointment Reminder (Time Event)*	*Time Trigger*	*Reminder Notification (out)*	*At scheduled times, system sends appointment reminders to patients.*	*Appointment, Notification*
9	*Patient Views EHR*	*EHR Access Request (in)*	*EHR Data (out)*	*Patient requests access; system retrieves medical history and lab results.*	*MedicalRecord, LabResult, PatientRecord*
10	*Doctor Updates Medical Record*	*Record Update Request (in)*	*Update Confirmation (out)*	*Doctor adds notes or updates patient information in the EHR.*	*MedicalRecord, PatientRecord*
11	*Patient Requests Billing Review*	*Billing Inquiry (in)*	*Billing Details (out)*	*Patient requests billing status; system retrieves their billing info.*	*BillingRecord, Patient*

Considerations

Listing Medalyze business events helps validate the work context and often reveals missing or unclear interfaces. As events are analyzed, adjustments may be required to the context diagram, especially when identifying new inputs, outputs, or external actors.

Business events also serve as the **foundation for gathering detailed requirements**, organizing analysis tasks, and ensuring that each BUC is traceable to specific system interactions. They also guarantee consistency between the event list, context model, and later specifications.

All event names, input/output flows, and data classes must comply with the naming conventions used in Medalyze and be defined precisely in the **data dictionary (Section 7)**.

Form

The work partitioning is presented as a **business event list/table**, with each row documenting:

- Event number
- Event name
- Triggering input
- Outputs
- Brief BUC summary
- Relevant business data classes

All names must match the work context diagram (Section 6b) and the data dictionary (Section 7b).

6d. Specifying a Business Use Case (BUC)

Content

For Medalyze, a Business Use Case (BUC) specification defines how the system and its associated human actors respond when a specific business event occurs. Each BUC describes, in detail, the workflow triggered by one of the previously identified Medalyze business events—such as a patient requesting an appointment, a lab sending results, or a pharmacy processing an e-prescription.

A BUC outlines the sequence of business actions, decision points, data interactions, and the movement of information between Medalyze and adjacent systems. This specification ensures that each event is handled consistently, accurately, and in line with clinical and administrative requirements.

Motivation

Specifying BUCs in detail allows stakeholders to fully understand the required business response for each event before defining software requirements. This understanding provides several benefits:

- Ensures clarity on how Medalyze should behave in real operational scenarios
- Helps identify which steps should be automated by Medalyze and which remain manual
- Reveals missing inputs, outputs, or data dependencies in the business event list
- Supports accurate requirements discovery and system design
- Creates a solid foundation for discussing workflow improvements, integrations, and automation opportunities

By defining the business behavior behind each event, the project team ensures that Medalyze will support healthcare workflows effectively and align with clinical operations.

Example

- **Patient Requests Appointment (BUC-01):** Describes how the system checks doctor availability, confirms eligibility, schedules the appointment, and notifies stakeholders.
- **Lab Sends Test Result (BUC-03):** Explains how lab data is validated, stored, linked to the medical record, flagged for abnormalities, and made accessible to the patient and doctor.
- **Pharmacy Processes Prescription (BUC-05):** Details how prescriptions are retrieved, medication stock is verified, processing status is assigned, and notifications are sent.

These BUC specifications serve as narrative or model-based descriptions that illustrate how Medalyze must respond to each business event.

Considerations

While specifying a Medalyze BUC, it is essential to stay within the boundaries of the defined **inputs and outputs** for the corresponding business event (from Section 6c). If, during BUC elaboration, new inputs or outputs surface—such as additional lab flags, status codes, or patient identifiers—this indicates the need to:

- Update the relevant business event entry
- Adjust the work context model (Section 6b)
- Revise naming conventions or data definitions (Section 7)

This ensures full traceability and prevents inconsistencies in system design and requirements development.

Form

A Medalyze BUC can be specified using a combination of descriptive and visual modeling techniques, such as:

- Activity diagrams
- Step-by-step BUC scenarios
- Process flow diagrams
- System sequence diagrams
- Swimlane diagrams
- Interview-based notes or structured narratives

Regardless of the modeling technique used, the inputs and outputs defined in the BUC **must match exactly** those documented in the Business Event List (Section 6c) and the Work Context Diagram (Section 6b) to preserve consistency and traceability across all artifacts.

7. Business Data Model and Data Dictionary

7a. Business Data Model

Content

A specification of the essential subject matter, business objects, entities, and classes that are germane to the product. It might take the form of a first-cut class model, an entity-relationship model, or any other kind of data model.

Motivation

To clarify the system's subject matter, thereby triggering recognition of requirements not yet considered. To discover missing requirements you can cross check the data model and the events using a Create, Reference, Update, Delete (CRUD) table. The data model is a specification for all of the business data that is relevant to the scope of the work.

Example

This is a model of the business system's business subject matter using the Unified Modelling Language (UML) class model notation. This is all the data that is Created, Referenced, Updated and Deleted by processes within the scope of the work being studied. See section 6 for more about the scope of the work.

7a. Business Data Model

Content

The business data model for Medalyze specifies the essential subject matter and business objects that define the healthcare environment in which the system operates. This includes key entities such as **Patient, Doctor, Appointment, MedicalRecord, LabResult, Prescription, Medication, BillingRecord, PharmacyStaff, Report, and Pharmacy**.

The model may be represented as a first-cut UML class diagram or any structured form that captures the relationships, attributes, and associations among these domain classes. It reflects the business data that Medalyze must handle, independent of implementation details or database design.

This model forms the foundation for understanding the core healthcare concepts Medalyze must manage across clinical, administrative, and operational workflows.

Motivation

Developing the Medalyze business data model clarifies the system's subject matter and ensures that all necessary business concepts are recognized early in the requirements process. By mapping the entities and their relationships, the project team can identify missing or overlooked requirements and validate interactions across business events.

The data model also supports cross-checking with the Business Event List using the **CRUD matrix**, ensuring every class can be Created, Referenced, Updated, and Deleted appropriately across Medalyze's use cases. This helps confirm that Medalyze supports all essential healthcare operations and that no important data structures are ignored.

Ultimately, the business data model serves as the core specification of all data relevant to the scope of the work and provides a bridge between requirements and system design.

Example

Example — Medalyze Business Data Model Overview

A UML-style class model for Medalyze would include the following key business data classes:

- **Patient**
- ○ Attributes such as patientID, name, dateOfBirth, contactInfo, allergies
- **Doctor**
- ○ doctorID, specialty, contactInfo
- **Appointment**
- ○ appointmentID, date, time, status
- **MedicalRecord**
- ○ recordID, historySummary, doctorNotes
- **LabResult**

  - labResultID, resultType, resultValue, abnormalFlag
- **Prescription**
  - prescriptionID, issueDate, status
- **Medication**
  - medicationID, name, dosage, quantity
- **BillingRecord**
  - billingID, amount, insuranceStatus
- **PharmacyStaff**
  - staffID, role
- **Administrator**
  - adminID, role
- **Report**
  - reportID, generatedDate, reportType
- **Pharmacy**
  - pharmacyID, name, location

Each rectangle in the class model represents one of these business classes.
Each attribute will be formally defined in the **data dictionary (Section 7b)**.

The model expresses the relationships among classes, such as:

- Patient ↔ Appointment
- MedicalRecord ↔ LabResult
- Prescription → Medication (composition)
- Doctor ↔ Prescription
- PharmacyStaff ↔ Prescription
- Patient ↔ BillingRecord

This representation ensures consistency, clarity, and traceability across requirements.

Considerations

When building the business data model for Medalyze, analysts should examine whether existing healthcare standards, terminology frameworks, or reference models (e.g., EHR structures, lab reporting formats, pharmacy record schemas) can serve as useful inspiration. These may help ensure that Medalyze aligns with common healthcare practices and integrates smoothly with external systems.

It is also important to ensure that each class represents **business-level concepts**, not technical implementations. The goal is to capture the logical organization of healthcare information and the real-world relationships between business entities.
All class names, attribute names, and relationship labels must be clear, unambiguous, and consistent with the project's naming conventions.

Form

The Medalyze business data model may be presented using:

- **UML class diagrams** (recommended)
- Entity Relationship (ER) diagrams
- A structured table listing:
- ○ Class name
- ○ Attributes
- ○ Relationships with other classes

The key requirement is that the model reflects **business-level data**, capturing the logical partitioning of information within the Medalyze work context. Attribute definitions for each class will appear in the **data dictionary (Section 7b)**.

7b. Data Dictionary

The glossary introduced in Section 4 establishes basic terminology. As Medalyze's scope becomes clearer, these terms are formalized in a **data dictionary**, which defines the content and meaning of all classes, attributes, relationships, and data flows used across the system. These definitions serve as the foundation for all atomic requirements and interface specifications.

Content

The Medalyze data dictionary defines the structure and meaning of:

- **Classes** in the business data model
- **Attributes** belonging to each class
- **Relationships** between classes (e.g., Patient–Appointment, Prescription–Medication)
- **Inputs and outputs** shown in the Work Context and Business Event List
- **Elements inside each input/output**, such as identifiers, status codes, timestamps, and clinical values

When Medalyze moves into implementation, technical interface specifications—such as formats, validation rules, and data types—will be added to the dictionary.

Motivation

Accurate definitions of Medalyze's information flows are essential because:

- The **work context diagram** establishes the scope of clinical and administrative data exchanged with external systems.
- The **product scope diagram** (Sections 8a and 8b) defines what Medalyze must manage internally.
- These boundaries can only be fully correct when all data entering or leaving the system is clearly defined.

The data dictionary ensures that developers, analysts, testers, and stakeholders share the same understanding of Medalyze's data structures, thereby avoiding inconsistencies and misinterpretations during design and implementation.

Examples

Classes:

Name	Content	Type
Appointment	appointmentID + date + time + status	Class
BillingRecord	billingID + amount + insuranceStatus	Class
Doctor	doctorID + specialty + contactInfo	Class

LabResult	labResultID + resultType + resultValue + abnormalFlag	Class
MedicalRecord	recordID + doctorNotes + historySummary	Class
Medication	medicationID + name + dosage + quantity	Class
Patient	patientID + name + dateOfBirth + contactInfo + allergies	Class
PatientRecord	recordEntryID + entryType + entryDate	Class
Pharmacy	pharmacyID + name + location	Class
PharmacyStaff	staffID + role	Class
Prescription	prescriptionID + issueDate + status	Class
Report	reportID + reportType + generatedDate	Class

Dataflows (Inputs/Outputs):

Name	**Content**	**Type**
Appointment Request	patientID + doctorID + preferredDate + preferredTime	Dataflow
Appointment Confirmation	appointmentID + date + time + status	Dataflow
LabResult Data	labResultID + patientID + resultType + resultValue + abnormalFlag	Dataflow
Prescription File	prescriptionID + patientID + doctorID + medicationList + issueDate	Dataflow
Prescription Status Update	prescriptionID + status + pharmacyID	Dataflow
Claim Request	billingID + patientID + amount + diagnosisCode	Dataflow
Claim Status	billingID + insuranceStatus	Dataflow
Report Request	reportType + timePeriod	Dataflow
Utilization Report	reportID + generatedDate + summaryMetrics	Dataflow

Reminder Notification	appointmentID + date + time + patientContact	Dataflow

Attributes / Elements

Name	**Definition**	**Type**
abnormalFlag	Indicates whether a lab result falls outside normal ranges	Attribute/Element
amount	Total cost associated with a service or visit	Attribute/Element
appointmentID	Unique identifier for an appointment	Attribute/Element
contactInfo	Phone, email, or preferred communication method of a user	Attribute/Element
date	Calendar date of an appointment or event (YYYY-MM-DD)	Attribute/Element
doctorNotes	Free-text notes recorded by a doctor within a medical record	Attribute/Element
dosage	Prescribed dosage instructions for a medication	Attribute/Element
entryDate	Date a patient record entry was added	Attribute/Element
entryType	Type of entry (e.g., Visit, Diagnosis, Procedure)	Attribute/Element
generatedDate	Date a report is generated	Attribute/Element
insuranceStatus	Result of claim validation (Approved, Denied, Pending)	Attribute/Element
issueDate	Date a prescription was issued (YYYY-MM-DD)	Attribute/Element
labResultID	Identifier assigned to a specific lab test result	Attribute/Element
medicationID	Unique identifier for a medication item	Attribute/Element
name	Full name of a patient, doctor, or pharmacy	Attribute/Element
patientID	System-generated identifier for a patient	Attribute/Element
resultType	Type of lab test (e.g., CBC, Glucose, Lipid Panel)	Attribute/Element

resultValue	Numeric or qualitative value of a lab test	Attribute/Element
specialty	Doctor’s field of expertise (e.g., Cardiology, Dermatology)	Attribute/Element
status	State of an appointment or prescription (Confirmed, Cancelled, Completed, OutOfStock)	Attribute/Element
time	Time of an appointment (HH:MM, 24-hour clock)	Attribute/Element

Considerations

The Medalyze data dictionary forms a bridge between analysts and implementers. As development proceeds, implementation teams add technical details such as:

- data formats (JSON, XML, HL7, FHIR)
- validation rules
- data types and constraints
- API request/response structures

During the analysis process, some terms originally listed in the glossary may evolve into formal data attributes or dataflows. When this occurs, entries should be moved into the data dictionary to ensure consistency.

It is essential that the data dictionary remains fully aligned with all requirements documents, models, and diagrams so that every stakeholder refers to the same definitions throughout the project lifecycle.

Form

The Medalyze data dictionary may be maintained in:

- a spreadsheet
- a structured table
- a requirements management system
- a modeling tool supporting class diagrams

Regardless of format, the dictionary must support easy cross-reference of terms across requirements, diagrams, events, and interface definitions, ensuring consistency and traceability throughout the project.

8. The Scope of the Product

8a. Product Boundary

A use case diagram for Medalyze defines the separation between the **actors** (patients, doctors, administrators, pharmacies, laboratories, insurers, and external systems) and the **product itself**. Determining the product boundary means deciding which parts of each Business Use Case (BUC) should be handled by the Medalyze system and which parts remain the responsibility of human users or external systems.

This decision process considers:

- The abilities and limitations of Medalyze users (patients, clinicians, pharmacy staff, administrators)
- Environmental and regulatory constraints (data privacy, healthcare regulations, system interoperability)
- Project goals (streamlining workflows, reducing manual processes, improving accessibility)
- Known clinical and administrative workflows

The product boundary defines exactly what Medalyze automates and ensures the product fits smoothly within the broader healthcare environment.

Application to Medalyze

The Product Use Case Diagram (PUC diagram) places human and automated actors—such as Patients, Doctors, Pharmacy Staff, Laboratories, Insurance Providers, and External Notification

Services—**outside** the product boundary. Inside the boundary are the **Product Use Cases** (PUCs) that Medalyze is responsible for, such as:

- Schedule Appointment
- Manage Medical Records
- Retrieve Lab Results
- Issue Electronic Prescriptions
- Process Prescription Updates
- Validate Insurance Claims
- Generate Reports
- Manage User Roles and Permissions

Each PUC links directly back to its corresponding Business Use Case (BUC), ensuring full traceability.

The product boundary decisions reflect where automation significantly improves efficiency, accuracy, and user experience—for example, automating appointment reminders, insurance validations, and prescription routing to pharmacies—while leaving clinical decision-making and human interactions outside the product.

In many cases, Medalyze collaborates with external systems: Laboratories supply lab results, pharmacies return prescription statuses, and insurance providers respond to claim validations. These are represented as adjacent systems connected through defined input/output flows.

Example

The Medalyze product boundary may include the following PUCs derived from BUCs:

BUC No.	Product Use Case (PUC)	Primary Actor
1	Schedule Appointment	Patient

2	Update Doctor Availability	Doctor
3	Store and Present Lab Results	Laboratory
4	Issue E-Prescription	Doctor
5	Receive Prescription Processing Status	Pharmacy Staff
6	Validate Insurance Claim	Insurance Provider System
7	Generate Admin Report	Administrator
8	Send Automated Appointment Reminder	Notification Service
9	Provide EHR Access	Patient / Doctor
10	Update Medical Record	Doctor
11	Retrieve Billing Information	Patient

The PUC diagram visually summarizes these interactions, showing Medalyze at the center with clearly labeled interfaces to each actor. Arrows indicate usage and data direction for maximum traceability.

This model makes it clear which activities are system-automated and which rely on external systems or human actions.

Form

The product boundary for Medalyze may be represented using:

- **A Product Use Case (PUC) Diagram** showing product use cases inside a boundary and actors outside

- **A Product Scope Diagram** summarizing the external interfaces (laboratories, pharmacies, insurers, etc.)

- **Individual PUC specifications and interface definitions**

- **Prototypes or UI mockups** illustrating how the system interacts with users

  These diagrams and specifications collectively define how Medalyze fits into the healthcare ecosystem and how each actor exchanges information with the system.

8b. Product Use Case Table

The Product Use Case (PUC) Table provides a structured summary of all Medalyze product use cases and the interfaces between the product and its actors.
It handles larger numbers of PUCs more effectively than a diagram and provides precise definitions of inputs and outputs at the product boundary.

Medalyze Use Case (PUC) Summary Table

PUC No.	PUC Name	Actor(s)	Input & Output
1	Schedule Appointment	Patient	**Input:** Appointment Request (in) **Output:** Appointment Confirmation (out)
2	Update Doctor Availability	Doctor	**Input:** Schedule Update (in) **Output:** Updated Schedule (out)
3	Store and Present Lab Results	Laboratory System	**Input:** LabResult Data (in) **Output:** LabResult Added Confirmation (out)

4	Issue Electronic Prescription	Doctor	**Input:** Prescription Request (in) **Output:** Prescription File (out)
5	Process Prescription Status	Pharmacy Staff / Pharmacy System	**Input:** Prescription File (in) **Output:** Prescription Status Update (out)
6	Validate Insurance Claim	Insurance Provider System	**Input:** Claim Request (in) **Output:** Claim Status (out)
7	Generate Administrative Report	Administrator	**Input:** Report Request (in) **Output:** Utilization or Performance Report (out)
8	Send Automated Appointment Reminder	Notification Service	**Input:** Time Trigger (in) **Output:** Reminder Notification (out)
9	Provide EHR Access	Patient, Doctor	**Input:** EHR Access Request (in) **Output:** EHR Data (out)
10	Update Medical Record	Doctor	**Input:** Record Update Request (in) **Output:** Record Update Confirmation (out)
11	Retrieve Billing Information	Patient	**Input:** Billing Inquiry (in) **Output:** Billing Details (out)

8c. Individual Product Use Cases

This section defines the **detailed behavior** of each Product Use Case (PUC) listed in the Medalyze Product Use Case Table. Each PUC focuses on how the Medalyze system responds to specific inputs from actors such as patients, doctors, laboratories, pharmacies, administrators, or external systems.

For each PUC, a detailed scenario or model can be created to show:

- the sequence of interactions between the actor and Medalyze
- the inputs Medalyze receives
- the internal processing steps or validations
- the outputs Medalyze produces
- any alternatives, exceptions, or error conditions
- how the PUC supports the corresponding Business Use Case (BUC)

These descriptions help clarify the product's functionality and ensure that all stakeholders have a shared understanding of how Medalyze behaves in each operational context.

Depending on the complexity of the PUC, the analyst may choose to use narrative text, diagrams, prototypes, or other modeling forms.

Form

- For Medalyze, each Product Use Case can be represented using any of the following formats:

- **A text-based scenario**, describing the step-by-step interaction between actors and Medalyze
- **A storyboard**, illustrating how the user interface supports the PUC
- **A formal use case specification**, including:
  - main success scenario
  - alternative flows
  - exception flows
  - preconditions and postconditions
- **A sequence diagram**, modeling communication between actors and system components
- **An activity diagram**, showing workflow decisions and process branches
- **A dataflow diagram**, representing how data moves through Medalyze during the PUC
- These models must be consistent with the inputs and outputs defined in the PUC Table (8b) and must remain traceable to the Business Use Cases (6c).

9. Functional Requirements

9a. Functional Requirements

Content

This section specifies each **atomic functional requirement** that Medalyze must satisfy. Each requirement uses the Volere requirement shell, ensuring that its attributes—such as description, rationale, fit criterion, originator, and dependencies—are clearly defined.

Atomic functional requirements describe **precisely what Medalyze must do**, at the smallest actionable level. These requirements trace back to:

- Business Events (Section 6c)
- Product Use Cases (Section 8b)
- Individual PUC specifications (Section 8c)

Each requirement must be testable, unambiguous, and directly linked to the work Medalyze must support in the healthcare environment.

Motivation

The purpose of the functional requirements section is to specify in detail the operations Medalyze must perform to support scheduling, clinical workflows, prescriptions, billing, reporting, and system administration.

These requirements ensure that:

- Medalyze performs all essential clinical and administrative actions
- System behavior is consistent, predictable, and verifiable
- Developers and testers have clear instructions
- The product aligns with stakeholder needs and constraints

Functional requirements represent the core behavior of the Medalyze system and form the backbone of implementation and validation activities.

Examples

Requirement #: 75 Requirement Type: Functional Event/BUC/PUC #: BUC 1 / PUC 1

Description: The product shall allow a patient to request an appointment by selecting a doctor, date, and time, and shall confirm the appointment if the selected slot is available.

Rationale: To ensure patients can reliably schedule appointments and receive confirmation without administrative delays.

Originator: Clinical Workflow Stakeholder Group

- Fit Criterion: The appointment shall be recorded in the system database immediately upon confirmation.
- The selected time slot must be marked as unavailable to other users within 5 seconds.
- A confirmation must be sent to the patient containing appointmentID, date, time, and doctor information.

Customer Satisfaction: 4 Customer Dissatisfaction: 5

Dependencies:

- Doctor availability data
- Patient account data
- Notification service configuration

Supporting Materials:

- Work Context Diagram
- Product Use Case Table
- Data Dictionary

Volere

History: Created February 29,2010— Medalyze Requirements Team

Fit Criterion

Every Medalyze functional requirement must include a **fit criterion**, which defines objectively measurable conditions indicating whether the requirement has been satisfied.

Fit criteria may include:

- time thresholds
- accuracy thresholds
- data validation rules
- correctness of system outputs
- consistency with data dictionary definitions

These criteria help testers determine precisely whether Medalyze fulfills each requirement as intended.

Considerations

Because the Medalyze project includes an extensive Business Event List and Product Use Case Table, these artifacts should be used to **trigger and group functional requirements**.

Traceability is essential:

- Each functional requirement should link to a BUC and/or PUC.
- New requirements discovered during analysis should be assigned unique identifiers.
- Grouping requirements by PUC ensures full coverage.

If the product boundary changes or new use cases emerge, the functional requirements must be updated accordingly to maintain completeness and consistency.

Form

Functional requirements for Medalyze may be documented using:

- A structured spreadsheet (recommended for traceability)
- A requirements management database or tool
- An intranet or shared repository
- A custom-built requirements tracking system
- Volere snow cards for discovery and refinement

Regardless of the format, requirements must follow consistent terminology and numbering to ensure:

- traceability across diagrams and models
- easy maintenance during revisions
- alignment with the data dictionary and event lists

Non-functional Requirements

1. Security Requirements

Description: The system shall ensure secure user authentication, protect patient data, and prevent unauthorized access.

Rationale: Medical data is highly sensitive and must be protected by law.

Fit Criterion: Users must authenticate through OAuth2. Only authorized roles can access medical records. All data in transit must use secure HTTPS/TLS encryption. An access log must record who viewed or modified a medical record.

2. Performance Requirements

Description: The system shall respond quickly and support the workload of doctors, patients, and administrators.

Rationale: Slow performance disrupts clinical workflow and affects patient service.

Fit Criterion: EHR pages load within a few seconds under normal usage. Prescription submission completes without noticeable delay. The system supports many simultaneous users without crashing.

3. Usability Requirements

Description: The system shall be easy for non-technical users (patients) and busy users (doctors) to understand and operate.

Rationale: Healthcare staff and patients cannot spend time learning complex interfaces.

Fit Criterion: Users can complete common tasks (booking an appointment, viewing records) without training. The interface uses clear labels, consistent layout, and simple navigation.

4. Reliability Requirements

Description: The system shall remain available and operate correctly without unexpected failures.

Rationale: Healthcare operations depend on system reliability.

Fit Criterion: The system remains accessible during clinic working hours. No data is lost during normal usage. If an error occurs, the system recovers gracefully without corrupting information.

5. Maintainability Requirements

Description: The system shall be easy for administrators and IT staff to update and troubleshoot.

Rationale: Healthcare systems change frequently and must be easy to maintain.

Fit Criterion: Errors and warnings are logged clearly. Administrators can update or configure the system without technical difficulty.

10. Look and Feel Requirements

10a. Appearance Requirements

Content

This section defines the requirements related to the **visual appearance, branding, and overall look and feel** of Medalyze. These requirements ensure that the interface reflects Medalyze's identity, supports a healthcare environment, and aligns with the expectations of patients, doctors, administrators, and partner organizations.

Appearance requirements may include:

- color schemes appropriate for medical software
- adherence to Medalyze branding guidelines
- clean, minimalistic design supporting clinical usability
- accessibility-driven visual standards
- culturally appropriate interface styles (e.g., bilingual layouts, right-to-left considerations where applicable)

These requirements describe the **intended visual qualities**, not the design implementation.

Motivation

To ensure that the appearance of Medalyze:

- minimalistic
- reflects Medalyze’s branding
- creates a visually consistent experience across web and mobile platforms
- supports patient comfort and ease of use
- provides clinicians with a clear, non-distracting interface

A well-defined appearance contributes to user confidence, faster navigation, and compliance with institutional standards.

Examples

- The product shall use a clean and calming color palette (e.g., blues, white, neutrals) suitable for medical environments.
-
- The interface shall visually differentiate patient, doctor, and admin views using layout and accent color variations.
-
- The product shall comply with Medalyze branding guidelines regarding typography, iconography, and logo usage.
-

- The product shall maintain high-contrast text and component design to support accessibility standards.

Fit Criterion

- A branding review committee shall verify that the interface adheres to Medalyze's approved style guide.
- Users (doctors, patients, and admins) shall report a minimum **80% satisfaction** with visual clarity during controlled usability tests.

Considerations

Even when using wireframes, prototypes, or mockups, it is essential to base them on **explicit appearance requirements**, not implicit assumptions.

Prototypes help to **elicit feedback and refine** appearance preferences, but they should never replace clear requirements. Appearance requirements must remain separate from design decisions until stakeholders confirm their goals, standards, and expectations.

10b. Style Requirements

Content

This section specifies the **style, mood, and emotional impression** that Medalyze should convey to its users. These requirements shape how patients, doctors, and administrators *feel* when interacting with the system, and guide designers in building an interface that matches Medalyze's identity and purpose.

For Medalyze, style requirements may include:

- the tone of the interface (authoritative, trustworthy, calm)
- consistency in style across web and mobile versions
- communicates clearly
- a professional yet accessible overall feel

Medalyze is not a manufactured physical product, so packaging requirements do not apply here.

These style requirements shape how the user perceives Medalyze and ensure that the system's behavior and look align with expectations for a trusted healthcare platform.

Motivation

In the healthcare domain, **style affects trust, comfort, and reliability**. Even when functional requirements are met, a product that feels overly technical, intimidating, or confusing may fail to gain adoption among doctors and patients.

Clear style requirements ensure Medalyze:

- appears trustworthy and medically professional
- reduces cognitive load for clinicians
- feels approachable for patients of varying ages and backgrounds
- maintains a consistent, polished identity
- reflects the seriousness of handling sensitive medical data

Proper style contributes directly to user confidence and increases overall system adoption.

Example

- The product shall appear **professional, trustworthy, and medically authoritative**.
- The product shall evoke a sense of **calm and clarity**, avoiding visually stressful elements.
- The interface shall maintain a **minimalist, clean design** to support clinical focus.
- Patient-facing screens shall feel **friendly and reassuring**, while doctor-facing screens shall feel **precise and data-centric**.

Fit Criterion

- At least **75% of test participants** (including both clinicians and patients) shall report feeling that the system is trustworthy after first use.
- In usability testing, **80% of clinicians** shall describe the interface style as "clear," "professional," or "medically appropriate."

- The style shall receive approval from the Medalyze branding and healthcare compliance review team.

Considerations

Style requirements often begin as broad terms such as *“professional,” “trustworthy,” “friendly,”* or *“authoritative.”* These should be clarified using **fit criteria** so designers have concrete guidance.

While prototypes help explore the intended style, they must be grounded in **explicitly defined style requirements**. Consistent interpretation by designers requires ensuring that:

- all team members understand the intended tone
- style is evaluated based on measurable reactions
- final design choices reflect the emotional and professional expectations of Medalyze stakeholders

11. Usability and Humanity Requirements

This section is concerned with requirements that make the product usable and ergonomically acceptable to its hands-on users.

11a. Ease of Use Requirements

Content

- This section specifies Medalyze’s **usability requirements**, reflecting how easy and intuitive the system should be for its intended users.
 Since Medalyze is used by **patients**, **doctors**, **administrators**, and **laboratory/pharmacy personnel**, in

  - The usability requirements should address:
  - **Efficiency of use:**
 How quickly clinicians and patients can complete common tasks (e.g., booking appointments, reviewing medical records, updating schedules).
  - **Ease of learning and remembering:**
 Users, especially patients, should be able to navigate Medalyze without extensive training.
  - **Error prevention and recovery:**
 Since Medalyze deals with health-critical data, error rates must be minimized and corrective guidance must be clear.
  - **User hours**
 Both medical professionals and patients should feel that the system supports their workflows comfortably.
  - **Feedback and clarity:**
 Users must receive clear, immediate system feedback (e.g., confirmations, warnings, validation messages) to ensure confidence in actions taken.

Motivation

These usability requirements guide Medalyze’s designers and developers to produce a system that:

- supports fast decision-making in medical environments
- minimizes training overhead for healthcare staff
- prevents operational errors that could impact patient safety
- ensures patients can comfortably manage their health information
- maintenance

A clear usability vision ensures Medalyze becomes a reliable and approachable tool in both clinical and non-clinical settings.

Examples

- The product shall allow a doctor to access a patient’s medical record within **three clicks** from the dashboard.
- The system shall provide patients with clear, step-by-step guidance when booking appointments.
- Medalyze shall prevent users from submitting incomplete or invalid forms by providing real-time validation messages.
- The interface shall be usable by patients with minimal technical knowledge and low literacy levels.

Fit Criterion

- **Efficiency:**
  85% of doctors in usability tests shall be able to locate and review a patient’s recent test results in under **10 seconds**.
- **Ease of remembering:**
  After two weeks without using the system, patients shall still be able to successfully navigate appointment booking, with at least **70% success rate** in tests.
- **Error rate:**
  During a one-month trial, medication order entry errors attributable to UI confusion shall be **below 1%**.
- **Satisfaction:**
  A post-use survey shall show that at least **80% of users** (patients and clinicians combined) rate the system as “easy to use.”
- **Feedback:**
  System response messages (confirmations, warnings, alerts) shall appear within **1 second** of the triggering action in usability evaluations.

Considerations

- Review the user categories defined earlier in the document to ensure usability requirements address the needs of all Medalyze personas, including patients with limited digital literacy and busy clinicians who require rapid interactions.

- Consider consultation with usability experts experienced in healthcare applications to validate that Medalyze meets industry standards for clarity, accessibility, and error prevention.

11b. Personalization and Internationalization Requirements

Content

- This section defines how Medalyze can be customized or configured to match **users' personal preferences, cultural expectations, and language requirements**. Since Medalyze serves a diverse user base—including patients, doctors, administrators, laboratories, and pharmacies—personalization and internationalization options are critical for accessibility and user comfort.
  - Personal
  - **Language and regional settings**
    - selectable languages
    - local spelling and phrasing
    - right-to-left or left-to-right layouts
    - date/time formatting preferences
    - local number and measurement conventions
  - **User-level customization**
    - notification preferences (SMS, email, in-app)
    - preferred dashboard layout (clinicians)
    - theme preferences (e.g., high-contrast mode, larger text)
    - saved patient or doctor preferences (frequently visited sections, default views)
  - **Cultural and regional compliance**
    - currency formats for billing modules
    - local medical regulatory terminology

Motivation

Personalization and internationalization ensure that Medalyze:

- accommodates the linguistic and cultural needs of all users
- feels familiar and comfortable regardless of a user's country or background

- reduces cognitive load by aligning with personal habits (e.g., date format, language)
- supports multilingual healthcare environments
- avoids forcing users into a one-size-fits-all interface

This increases user satisfaction, adoption, and accessibility—especially in global or multicultural healthcare settings.

Examples :

- The product shall allow users to select their preferred language from a list of supported options.
- The system shall store patient interface preferences (theme, font size, notification settings).
- Medalyze shall allow healthcare facilities to configure regional formats for dates, times, and currencies.
- The interface shall support right-to-left languages when enabled.
- Doctors shall be able to personalize their dashboard layout to show their most frequently accessed tools and patient lists.

Considerations

- Consider the cultural and linguistic preferences of all current and potential Medalyze user groups.
- International users or migrant patients may require different languages, spellings, or calendar formats.
- Personalization increases user engagement by allowing individuals to shape their own experience.
- Some clinics or hospitals may also require role-based configuration—different functional layouts for doctors, nurses, or administrators.
- Configurability may extend beyond appearance to include functional variations, allowing different types of users to enable or disable certain features based on their needs.

11c. Learning Requirements

Content

This section specifies how easy it should be for users to learn how to operate Medalyze.
Because Medalyze is used by **patients**, **doctors**, **administrative staff**, and **partner organizations**, learning requirements must reflect the different levels of expertise and the varying expectations across these groups.

Learning requirements address:

- how quickly new users can become productive
- whether training is needed
- the availability of tutorials, guides, or interactive help

Some Medalyze users (patients) should be able to use the system immediately with little or no training. Others (clinicians, admins) may require short onboarding sessions to understand professional features but should still be able to learn the system efficiently.

Motivation

The purpose of these requirements is to ensure that Medalyze feels **natural**, **logical**, **respectful of user workflows**, and **aligned with healthcare expectations**.
When the system uses familiar concepts and avoids unnecessary technical complexity, users can immediately understand how it supports their tasks and where to find what they need.

By avoiding internal system terminology and minimizing cognitive effort, Medalyze becomes:

- easier to learn
- more likely to be adopted
- quicker to operate
- less error-prone

Ensuring understandability is essential in a healthcare environment, where clarity contributes to safety, accuracy, and efficient patient care.

Examples

- The product shall allow patients to perform basic tasks (e.g., booking appointments, viewing results) without any formal training.
- A doctor shall be able to learn to navigate the full patient record system with less than one hour of guided onboarding.
- Administrative staff shall become fully productive in scheduling, billing, and record management within two days of training.
- The system shall include contextual help and tooltips to reduce the need for formal instruction.

Fit Criterion

- **Patients:**
  At least **80% of first-time patient users** shall be able to book an appointment within **3 minutes** during usability tests, without assistance.
- **Doctors:**
  After **one hour of guided introduction**,**90% success rate**.
- **Admins:**
  After **two days of training**, administrative users shall complete core tasks (scheduling, billing actions, profile updates) with an error rate below **2%**.
- **General:**
  A usability test panel shall show that **85% of new users** can complete essential tasks within defined time limits using only the built-in help system.

Considerations

- Review the user types described earlier to ensure all groups—patients, clinicians, administrators, and external partners—are considered.

- Learning requirements should reflect realistic user abilities and the complexity of the tasks they perform.

- If certain modules require more in-depth understanding (e.g., clinical dashboards), they may require short training resources or tutorials.

- Consider including interactive onboarding, help overlays, FAQ sections, and context-sensitive hints to support quick learning.

11d. Understandability and Politeness Requirements

This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end users.

Content

This section defines the requirements that ensure Medalyze is **instantly understandable** to its users and uses concepts that feel natural to their real-world roles. Understandability focuses on whether users can intuitively grasp:

- what the system does
- how to navigate it
- how its features relate to their daily tasks

Politeness refers to how the product **respects the user's time and knowledge**, such as avoiding unnecessary inputs and presenting familiar terminology, symbols, and workflows.

For Medalyze, understandability and politeness involve:

- avoiding technical jargon unrelated to healthcare
- preventing users from entering information the system already stores
- guiding actions with clear, friendly communication
- keeping the interface predictable and consistent

These requirements ensure Medalyze aligns with the mental models and expectations of both clinical and non-clinical users.

Motivation

To avoid forcing users to learn terms and concepts that are part of the product's internal construction and are not relevant to the users' world. To make the product more comprehensible and thus more likely to be adopted by its intended users.

Examples

- The product shall use terminology familiar to patients and clinicians (e.g., “test results,” “appointment,” “medications”) rather than internal system terms.
- The system shall not require users to re-enter data already present in their records (e.g., personal info, assigned doctor, known allergies).
- Icons used throughout Medalyze shall reflect widely recognized medical symbols (e.g., stethoscope for doctors, pill icon for medications).
- Messages and instructions shall be written in clear, friendly, non-technical language.

Considerations

- Review the needs of all user types to ensure terminology, icons, and system actions are intuitive from each perspective.
- Consider differences between patient and clinician language; patients need simple explanations, while clinicians expect professional medical terminology.
- Avoid requiring knowledge of internal system structures or databases.
- Ensure consistency across modules so users do not have to relearn how different sections operate.
- Politeness includes minimizing redundant steps and reducing the need for unnecessary user input.

11e. Accessibility Requirements

Content

These requirements specify how Medalyze must accommodate users with a wide range of disabilities to ensure that all patients, doctors, and pharmacists can effectively interact with the system. This includes support for visual impairments, reduced mobility, color-blindness, hearing limitations, cognitive challenges, and age-related accessibility needs. Medalyze must provide accessible interfaces across all platforms (mobile, web, tablets) used in clinical environments.

Motivation

Healthcare systems must be inclusive. Medalyze supports patient care and must therefore remain usable by individuals with accessibility needs—both patients and healthcare professionals. Ensuring accessibility expands the system's usability to a broader population, improves safety, reduces the likelihood of user error, and adheres to healthcare accessibility standards and local regulatory requirements. Excluding any group would compromise Medalyze's mission of delivering equal access to healthcare services.

Examples

- Medalyze shall provide a **high-contrast mode** suitable for users with low vision.
- Medalyze shall be fully navigable using a **keyboard or alternative input devices** for users with limited mobility.
- Medalyze shall offer **scalable text sizes** on all major screens (appointments, prescriptions, notifications).
- Medalyze shall ensure all critical UI components are **distinguishable without reliance on color alone**, supporting color-blind users.
- Medalyze shall provide **screen-reader compatibility**, enabling visually impaired users to access appointment details and prescription information.

Considerations

Not all disabilities are visible or widely recognized. Many healthcare workers experience color blindness, mild vision impairment, or reduced mobility due to long shifts. Patients may have temporary impairments (e.g., post-surgery) that make interaction difficult. Medalyze must therefore avoid designs that rely solely on color-coded status indicators, overly small interactive elements, or rapid interactions requiring fine motor control. Accessibility features must remain consistent across devices and user roles to ensure safe and inclusive use in medical settings.

12. Performance Requirements

12a. Speed and Latency Requirements

Content

Specifies the performance expectations for Medalyze when processing core healthcare operations. These include loading patient profiles, booking appointments, issuing prescriptions, retrieving pharmacy medication status, and sending notification updates. Speed requirements mainly define how quickly Medalyze must respond to user interactions and how fast background processes—such as status synchronization with pharmacies—must complete.

The environment of intended use includes clinics, pharmacies, and patient mobile devices, all of which require short response times to maintain workflow efficiency. Therefore, Medalyze must operate at a speed that ensures healthcare professionals and patients can complete tasks without delays.

Motivation

Timely system responses are essential in medical environments. Doctors often need to check records or issue prescriptions during consultations, requiring near-instant access to information. Pharmacies need quick updates on medication availability to avoid delays in patient treatment. Patients require immediate feedback when booking appointments or receiving notifications.

If Medalyze responds too slowly, clinical workflow may be interrupted, waiting time may increase, and user trust may decline. Ensuring acceptable response times helps prevent frustration, improves efficiency, and supports safe decision-making in time-sensitive healthcare scenarios.

Examples

- Medalyze shall load a user's Dashboard (appointments + prescriptions summary) within 2 seconds.

- Any interaction between a user and Medalyze’s interface (e.g., opening appointment details, viewing prescriptions) shall have a maximum response time of 2 seconds.

- When a doctor submits a new prescription, Medalyze shall transmit it to the pharmacy system within 5 seconds.

- When a pharmacy updates medication availability, Medalyze shall refresh the prescription status for the doctor within 10 seconds.

- The system shall send automated appointment reminders within 1 minute of the scheduled reminder time.

- The product shall synchronize inventory status with pharmacy endpoints every 15 seconds.

Fit Criterion

To ensure that Medalyze meets acceptable performance levels, measurable thresholds must be defined. The following criteria apply:

- **User interface responses (patients, doctors, pharmacists):** Medalyze shall respond to any user-initiated action (opening records, booking appointments, sending prescriptions) **within 1.5 seconds for 90% of interactions**, and **no response shall exceed 3 seconds**.
- **Prescription processing:** Transmission of a prescription from doctor to pharmacy shall complete **within 2 seconds** under normal network conditions.
- **Medication status updates:** Medalyze shall synchronize pharmacy medication availability **every 30 seconds**, with updates applied system-wide no later than **5 seconds after retrieval**.
- **Notification delivery:** Appointment reminders, prescription alerts, and pharmacy shortage notifications shall be dispatched **within 10 seconds** of the triggering event.
- **System startup time (mobile + web):** The app dashboard shall load within **4 seconds** on supported devices.

These fit criteria allow developers and testers to confirm that Medalyze operates at a speed appropriate for clinical and patient use.

Considerations

Different operations vary in their need for speed:

- **Critical tasks**—such as loading patient medical records during a consultation or sending urgent prescription updates—require the fastest response times.

- **Non-critical background processes**—such as periodic data synchronization—may tolerate slightly longer durations as long as they do not impact user tasks.

- Performance may depend on device type (mobile vs. desktop), network quality, and server load.

- Because Medalyze may be used in clinics with poor internet connectivity, caching and optimized data transfer methods should be considered to maintain responsiveness.

- Peak load times (such as early mornings when appointment activity is high) must be accounted for to avoid performance degradation.

12b. Safety-Critical Requirements

Content

Specifies risks related to patient safety, medical data accuracy, prescription handling, and the potential for harm arising from incorrect system operation. Medalyze operates in a healthcare environment where errors can result in delayed treatment, incorrect medication, privacy breaches, or failure to notify users of critical updates. Safety-critical requirements define the standards and controls needed to ensure Medalyze's behavior never endangers patients, healthcare staff, or medical decision-making.

Motivation

Medalyze deals with sensitive medical workflows such as appointments, prescriptions, and pharmacy updates. A failure in these workflows could lead to missed medical interventions, incorrect medication distribution, or harmful delays in care. Clearly documenting safety-critical requirements ensures:

- Medical decisions are supported by accurate and timely data
- Prescriptions are transmitted correctly

- System errors do not lead to patient harm
- Regulatory compliance is maintained across healthcare environments

  Because healthcare systems are legally regulated, Medalyze must meet strict safety and information-handling standards to reduce liability and protect users.

Examples

- The system shall not present outdated or incorrect prescription information to doctors or pharmacists.
- Appointment reminders shall always reach users within the configured timeframe to prevent missed medical visits.
- Medalyze shall prevent the issuing of prescriptions if essential patient information is missing or inconsistent.
- The system shall ensure that medication status updates from pharmacies cannot be lost, duplicated, or overwritten incorrectly.
- Medalyze shall lock user accounts after repeated suspicious login attempts to protect medical data from unauthorized access.

Fit Criterion

Each safety requirement must be testable. For example:

- **Prescription Accuracy Standard:**
  Medalyze shall demonstrate, in testing, a 100% match between prescription data received by a pharmacy and the prescription sent by the doctor.
- **Notification Reliability:**
  For any critical notification (prescription status, appointment reminders), **at least 99.5%** must be delivered within the required timeframe during system load testing.
- **Data Integrity Compliance:**
  Medalyze shall comply with healthcare safety standards applicable in the target country/region (e.g., EU medical data handling standards, local Patient Safety Acts). Evidence shall be provided via certification or audit.
- **Security Safety Standard:**
  No member of a test panel of defined size shall be able to access protected patient information without proper authorization.

Considerations

- Safety standards differ between countries; Medalyze must adapt depending on the healthcare market it serves.

- Prescription and appointment errors carry significantly higher safety risks than UI or navigation errors; therefore these workflows require stricter validation and redundancy.

- Consultation with healthcare regulatory bodies, legal advisors, and medical safety experts is essential to ensure compliance with national health regulations.

- Safety requirements must evolve as medical risks change (e.g., updates to prescription handling laws, new data protection regulations).

- Logging, auditing, and traceability mechanisms must be built into Medalyze so that incidents can be investigated and prevented in the future.

12c. Precision or Accuracy Requirements

Content

This section defines how precise and accurate Medalyze must be when handling medical data, clinical measurements, timestamps, dosage information, and system-generated calculations. Accuracy applies to patient records, appointment timestamps, prescription details, medication dosages, inventory quantities, and any automated decision-support outputs. The system must consistently maintain medically acceptable precision to avoid mistakes in diagnosis, prescribing, or scheduling.

Motivation

Medical systems require high accuracy because even small data inconsistencies can lead to incorrect treatments, medication errors, misaligned appointments, or delayed care. Defining accuracy requirements ensures that users have consistent expectations for how precise Medalyze will be in storing, displaying, and processing clinical information.

Examples

- Medication dosages recorded in prescriptions shall be accurate to the **nearest milligram** or the unit defined by medical standards for that drug.
- Appointment timestamps shall be stored and displayed in **minute-level** precision and synchronized to the system's unified time standard.
- Patient vital-sign readings imported from integrated devices (e.g., thermometers, heart-rate sensors) shall maintain accuracy within **±1%** of the device's certified measurement accuracy.
- Pharmacy stock levels shall be tracked with a precision of **one unit** (e.g., 1 pill, 1 bottle, 1 package).
- The system shall record all medical events using UTC to avoid discrepancies across different locations or device clocks.

Considerations

Some accuracy requirements may already be implicitly defined in earlier sections (e.g., data definitions in the Business Data Model).
Medalyze must follow standardized clinical units (mg, ml, bpm, °C, etc.) and avoid mixing unit systems, which could cause dangerous misinterpretations.
Device-integrated measurements may rely on third-party equipment; the system must respect their certified accuracy limits.
Accurate time synchronization is essential for medical auditing, appointment tracking, and prescription validation.
Different countries may use different units or decimal conventions, which the system must account for when localizing outputs.

12d. Reliability and Availability Requirements

Content

This section defines the required reliability and availability levels for **Medalyze**, ensuring that the platform can consistently support clinical, administrative, and pharmacy operations. Reliability concerns how often Medalyze may fail or experience errors; availability specifies the portion of time the system must be operational and accessible to users.

Since Medalyze is a healthcare-support system used by doctors, pharmacists, and patients, it must operate with high dependability—particularly during peak hours when medical decisions depend on timely system responses. The system must also maintain data integrity even under heavy load or when partial outages occur.

Motivation

Healthcare systems must function with minimal downtime because failures can affect diagnosis, medication dispensing, appointment management, and patient communication. Setting clear expectations for reliability and availability allows developers and stakeholders to assess risks, plan for contingencies, allocate resources appropriately, and ensure that Medalyze supports safe clinical operations.

Additionally, these requirements help the client understand the realistic level of service and guide decisions about hosting infrastructure, redundancy, backup systems, and maintenance schedules.

Examples

- The Medalyze platform shall be available **24 hours per day, 7 days per week**, except during scheduled maintenance.
- The system shall achieve **at least 99.5% uptime** per calendar month.
- Scheduled maintenance windows shall not exceed **4 hours per month**, and users must receive notification at least **48 hours in advance**.
- No critical clinical operation (e.g., issuing prescriptions, viewing patient history, updating pharmacy stock) shall experience failure more than **0.1% of the time**.
- Medalyze shall automatically switch to a **redundant backup database** in the event of primary database failure.

- Error recovery from non-critical failures shall occur within **30 seconds**, whenever possible.

Considerations

- Healthcare environments often require **continuous access**, but the cost of very high availability (e.g., 99.9%+) may require significant investment in redundancy and infrastructure; stakeholders must determine whether the increased cost is justified.
- For Medalyze, some functions are more critical than others. For example, viewing a prescription or medication availability may be more urgent than viewing general profile settings.
- Consider whether the requirement is truly for **availability** (system accessible) or **reliability** (system never fails during a given operation). These impose different design expectations.
- Medalyze may require separate availability targets for:
- Clinical functions
- Administrative features
- Pharmacy inventory updates
- Regional deployments may have different availability expectations depending on local healthcare regulations or server infrastructure.

12e. Robustness or Fault-Tolerance Requirements

Content

These requirements define Medalyze's ability to continue operating correctly even when abnormal events occur. Such events may include loss of internet connection, temporary server outages, failures in third-party services (such as pharmacy APIs), corrupted data, or user-side device issues. Medalyze must maintain essential healthcare functionalities—such as viewing appointments, accessing cached patient data, or queuing prescription actions—under degraded conditions.

Motivation

Medalyze is used in environments where uninterrupted access to medical information and workflows is essential. Failures or interruptions may delay treatment, disrupt clinical workflows, or reduce user trust. Robustness ensures that Medalyze continues providing core services during unexpected events and recovers gracefully afterward. This is critical in healthcare, where even minor disruptions can have meaningful impact.

Examples

- Medalyze shall continue to function in **offline or limited-connectivity mode**, allowing users to view cached appointments and patient records until a connection is restored.
- If the pharmacy service is temporarily unreachable, Medalyze shall **queue pending prescription transmissions** and send them automatically once the connection is available.
- During a server outage, the system shall **fallback to local session data** to allow doctors to continue viewing essential information.
- Medalyze shall provide **emergency read-only access** to critical patient information even if some components fail.

Considerations

Abnormal events are common in distributed healthcare systems—network failures, hardware issues, API downtime, and data inconsistencies can happen at any time. Robustness requirements aim to prevent complete system failure and support graceful degradation.

Medalyze may also require a **disaster recovery strategy**, ensuring that patient data, appointments, and prescription logs can be restored quickly after severe faults. Backup frequency, failover procedures, and data replication policies should be considered to maintain system reliability in adverse conditions.

12f. Capacity Requirements

Content

This section defines the volume of users, medical records, prescriptions, and appointment activity that Medalyze must be able to handle at any given time. It specifies the expected daily and peak operational loads, including simultaneous user sessions, database record growth, and communication traffic between doctors, patients, and pharmacies. It also includes storage capacity for prescription histories, appointment logs, pharmacy inventory updates, and notification records.

Motivation

To ensure Medalyze can reliably support real healthcare environments where multiple hospitals, clinics, doctors, and patients may access the system simultaneously. Proper capacity planning prevents slow response times, system overload, and data-processing bottlenecks, all of which could negatively impact patient care and clinical workflow.

Examples

- The system shall support **at least 500 concurrent active patient, doctor, and pharmacy users** during morning peak hours (8:00 A.M. – 12:00 P.M.).
- Medalyze shall process **a minimum of 10,000 appointment-related events** (bookings, updates, cancellations, reminders) per day.
- The prescription module shall handle **up to 5,000 new prescriptions** issued per day and **3,000 pharmacy stock updates**.
- The system shall store **up to 5 years of patient appointment history** and **all prescription data** without degradation of performance.
- Notification services shall support **up to 20,000 outgoing notifications per hour** (SMS, push, or email).

Fit Criterion

Since the examples are quantified, each requirement is fully testable. Medalyze will be considered compliant when load-testing confirms that all capacity limits are met or exceeded under simulated peak hospital and pharmacy usage conditions.

12g. Scalability or Extensibility Requirements

Content

This section specifies Medalyze's expected ability to support increasing usage demands as the healthcare ecosystem grows. Scalability applies to user volume (patients, doctors, pharmacists), system interactions (appointment operations, prescription events), and stored data (records, logs, pharmacy status updates). Extensibility includes Medalyze's ability to incorporate new features such as integration with additional healthcare systems, expansion of appointment types, or increased prescription-processing complexity.

Motivation

Healthcare workloads grow as clinics expand, patient numbers increase, and additional medical services become digitized. Medalyze must be designed to accommodate this growth without performance degradation, system instability, or the need for major architectural redesign. Proper scalability ensures long-term viability and supports strategic expansion across multiple clinics, pharmacies, or regional healthcare networks.

Examples

- The system shall support an increase from **5,000 active users** at launch to **50,000 active users** within three years.

- Medalyze shall be able to process **up to 10,000 appointment-related events per hour** during peak usage (e.g., early morning scheduling surges).

- The prescription-processing service shall scale to handle **a 300% increase in pharmacy interactions** as additional pharmacies join the Medalyze network.

- The database architecture shall allow storage growth from **2 million to 20 million medical records**, including logs and historical appointment/prescription data.

12h. Longevity Requirements

Content

Defines the expected operational lifetime of Medalyze, including how long the platform should remain functional, maintainable, and cost-effective under normal healthcare usage. Longevity also includes expectations for how long the system's software architecture, database structure, and infrastructure should be viable before requiring major upgrades or replacement.

Motivation

Medalyze supports critical healthcare workflows such as appointment management and prescription processing. Because healthcare systems depend on continuity, the product must be designed for long-term use to justify investment, reduce disruption to medical operations, and avoid costly full-system rewrites. Long-term longevity also ensures regulatory compliance, predictable maintenance budgeting, and stable support for users.

Examples

- The Medalyze platform shall remain operational and maintainable for **a minimum of 7 years** without requiring a full architectural replacement.

- The system shall remain compatible with planned updates to hospital IT standards and regulatory frameworks throughout its lifetime.

- Medalyze shall support infrastructure scalability and security patching for its entire operational life without major downtime.

13. Operational and Environmental Requirements

13a. Expected Physical Environment

Content

This section specifies the physical environments in which Medalyze must reliably operate. Because Medalyze is used by patients, doctors, pharmacists, and administrative staff, the system must function across a mix of clinical, remote, and mobile settings. These include hospitals, outpatient clinics, pharmacies, patient homes, and mobile usage environments where lighting, noise, and device constraints may vary. Medalyze must remain functional on standard consumer devices—smartphones, tablets, and desktops—as these devices may be used in both controlled and uncontrolled physical environments.

Motivation

Healthcare professionals work in environments that can be rushed, crowded, or resource-limited. Patients may interact with Medalyze from home or while on the move. Highlighting the physical conditions in which Medalyze is expected to operate ensures the system is designed to remain usable, visible, and responsive in realistic situations. This prevents avoidable usability issues and ensures Medalyze is fit for clinical operations and safe for patient-facing tasks.

Examples

- The interface shall remain readable on small screens typical of budget smartphones used by some patients.
- Medalyze shall not produce disruptive notification sounds in clinical environments where quiet operation is necessary.

Considerations

Medalyze may be used in spaces with poor lighting, high activity levels, or variable screen quality. Consider whether additional accessibility features (already addressed in 11e) should be enabled by default for environments with glare, dim lighting, or small screens. Also consider workflows performed while standing, walking, or multitasking, which may influence layout and interaction design.

13b. Requirements for Interfacing with Adjacent Systems

Content

This section specifies how Medalyze must communicate and integrate with external healthcare systems and partner applications required for its operations. These include electronic health record (EHR) systems, national e-prescription networks, pharmacy management systems, appointment calendar providers, authentication services, and secure notification platforms. The requirements define the necessary data exchanged, the connection mechanisms, communication formats, and the operational behavior of each interface.

Motivation

Medalyze depends on accurate, timely communication with external systems to function correctly. Missing or incomplete interface requirements often lead to costly redesign during implementation. Defining these interfaces early ensures consistent data exchange, supports interoperability between healthcare providers, and minimizes integration risks during deployment.

Examples

- Medalyze shall be able to securely transmit prescriptions to approved pharmacy management systems through standardized e-prescription APIs.
- Medalyze shall retrieve doctor availability from external calendar systems using supported synchronization protocols.
- Medalyze shall interface with national or regional healthcare registries for verifying practitioner credentials.
- Medalyze shall support login authentication via external identity providers (e.g., OAuth-based hospital systems).
- Medalyze shall receive pharmacy stock availability updates from partner pharmacy systems in real time or scheduled intervals.

Fit Criterion

- For each Medalyze interface with an adjacent system, the following must be fully specified:

- The data content
(e.g., prescription fields, appointment metadata, pharmacy stock values).
- The physical material content
(if applicable, such as secure certificates or authentication tokens).
- The medium that carries the interface
(HTTPS REST API, secure message queue, cloud-based data exchange service).
- The frequency
(real-time, scheduled every 10 minutes, on user request, event-triggered).
- The volume
(expected number of transactions per hour, peak load, batch sizes).
- The trigger
(doctor approval, pharmacy update event, patient action, system time interval).
- The standards/protocols that apply
(HL7/FHIR, OAuth2, TLS 1.3, JSON schema, regional e-prescription standards).

13c. Productization Requirements

Content

This section specifies the requirements needed to package Medalyze into a deployable, maintainable, and distributable healthcare software product. It includes installation procedures, deployment configurations, packaging formats, and any prerequisites required to run Medalyze on hospital, clinic, or pharmacy systems. It also defines what must be provided to IT departments (e.g., documentation, environment settings, configuration files) to ensure that Medalyze can be installed, updated, and maintained without technical complications.

Motivation

Proper productization ensures that Medalyze can be deployed smoothly across different healthcare environments—clinics, hospitals, pharmacies, or telehealth setups—without excessive setup time or specialized staff. These requirements also guarantee that installation costs, training overhead, and deployment risks are kept to a minimum. Clarifying these expectations early prevents deployment delays and ensures a consistent installation experience across all clients.

Examples

- **Medalyze shall be distributed as a secure installation package accompanied by a digital signature to ensure authenticity.**
- **Medalyze shall include an automated installer that checks for required dependencies (e.g., database connection, API keys, OS compatibility).**
- Medalyze shall be installable by standard IT staff without requiring specialist software engineers.
- Medalyze shall provide a deployment guide detailing server specs, database configurations, and recommended security settings.

Considerations

- Medalyze may need licensing or subscription verification mechanisms to ensure that only authorized healthcare providers can use the system.

- Cloud-based vs. on-premise deployments may have different installation requirements—these must be identified and documented.

- Medical environments often have strict IT security rules; installation workflows must respect these constraints (e.g., no unsigned executables, limited internet access).

- Marketing and operations teams may have assumptions regarding how long installation should take or how easily clinics should be able to scale their installations—these expectations must be gathered and reflected in the requirements.

- Consider the need for automatic updates, rollback capabilities, or audit logs during installation to comply with healthcare system safety and regulatory needs.

13d. Release Requirements

Content

This section specifies Medalyze's expected release cycle, including how frequently updates, improvements, and bug fixes will be delivered. It also outlines the form each release will take—such as maintenance patches, feature updates, and major version upgrades—and describes compatibility expectations between releases. It defines how Medalyze will be packaged and deployed across its different user environments (patients, doctors, pharmacies), ensuring ongoing stability and compliance with healthcare standards.

Motivation

A clear release strategy is essential for medical software because healthcare environments require predictable, reliable updates that do not disrupt clinical operations. Documenting release expectations ensures:

- healthcare providers can plan around updates,
- user training and documentation stay aligned,
- Medalyze continues to meet regulatory and security obligations,
- maintenance teams understand the expected update cadence and workload.

Having a defined release plan also helps avoid unexpected downtime and reduces the risk of breaking critical healthcare workflows.

Examples

- Medalyze maintenance releases shall be issued **quarterly**, including security patches, minor fixes, and incremental improvements.
- Feature updates shall be delivered **biannually**, adding new capabilities while maintaining backward compatibility with existing user workflows.

- Emergency hotfixes addressing critical issues (e.g., prescription transmission failures, appointment system outage) shall be released **within 48 hours** of issue identification.
- Updates must ensure that all existing patient data, appointment schedules, and prescription records remain intact and unaffected.

Fit Criterion

Each Medalyze release must include:

- A documented list of changes, fixes, and new features.
- Confirmation that **no existing functionality used by patients, doctors, or pharmacies becomes unavailable** due to the update.
- A completed regression test suite with **0 critical failures** and **no more than 2 minor cosmetic defects**.
- Deployment requiring **no more than 30 minutes** of scheduled downtime for system-wide updates.
- Verification that all APIs and integrations (pharmacies, authentication systems, medical databases) remain stable and fully functional.

Considerations

- Medalyze must comply with regional healthcare regulations, which may influence release timing (e.g., yearly compliance updates).
- Integration partners such as pharmacies and authentication services may require coordinated release windows.
- Contracts with hospitals or clinics may include SLA obligations for update frequency or downtime limits.
- Release timing must avoid peak medical usage periods (e.g., winter flu season) to avoid disrupting healthcare operations.
- Training materials, onboarding guides, and user documentation must be updated for each feature release so that medical staff are not left with outdated instructions.

14. Maintainability and Support Requirements

14a. Maintenance Requirements

Content

A quantification of the time and effort required to make changes or enhancements to Medalyze. This includes updates to appointment workflows, prescription handling logic, pharmacy status integration, UI improvements, bug fixes, and modifications to business rules. Maintenance also covers adapting the system to new healthcare regulations, updating medical terminology, and ensuring compatibility with evolving pharmacy and clinic systems.

Motivation

To ensure that all stakeholders understand the maintenance expectations for Medalyze, including how fast system updates should be completed, the resources required, and the level of technical support necessary for keeping the system functional and compliant. Clear maintenance requirements ensure that Medalyze can evolve reliably alongside healthcare workflows and regulatory changes.

Examples

- New automated appointment reminder logic shall be deployable within **three working days** after approval of requirements.
- A new pharmacy integration endpoint shall be fully implemented and tested within **one week** of receiving the official API documentation.
- Updated medication lists or dosage guidelines provided by regulatory health authorities shall be applied to Medalyze within **48 hours**.
- A new clinic or doctor profile shall be able to be added to the Medalyze system **within one working day**.

Considerations

- Medalyze may need to be maintainable by IT staff at clinics or by support teams who were not part of the original development team. This affects how code is structured, documented, and modularized.

- Adequate system documentation, including API references, data models, and deployment procedures, must be maintained to enable efficient updates.
- Training may be required for administrators responsible for managing clinic data, user access, or pharmacy configurations.
- Testability should be considered part of maintenance: automated tests, regression checks, and sandbox testing environments should be maintained to ensure updates do not break existing functionality.

14b. Supportability Requirements

Content

This section specifies the level and type of support required to ensure Medalyze remains usable and reliable for patients, doctors, and pharmacists. Support includes technical helpdesk assistance, in-app self-help tools, error reporting systems, and remote diagnostic capabilities.

It also defines expectations for issue resolution times, supported communication channels (e.g., email, ticketing, in-app support), and the built-in tools Medalyze must provide for users to obtain help without external intervention.

Support may be provided both by human staff (Medalyze administrators, IT personnel) and by automated features embedded directly in the system (guided troubleshooting, FAQ search, context-aware help prompts).

Motivation

Medalyze is a healthcare-focused system where timely and accurate issue resolution is critical. Without well-defined supportability requirements, users may face delays in accessing care services, retrieving medical information, or processing prescriptions. Clear support requirements ensure the product is prepared for real operational environments, reduce user frustration, minimize downtime, and provide healthcare professionals with confidence that technical issues will be resolved efficiently.

They also ensure Medalyze scales responsibly as usage grows and that support expectations are understood by developers, administrators, and clinical stakeholders.

Considerations

- **Support Level Variation:** Different users (patients, clinicians, pharmacists) have different support needs—clinicians may need urgent assistance during consultation hours, while patients may rely more on automated guidance.
- **Self-Support Options:** Medalyze may require built-in support tools such as contextual help, searchable knowledge bases, or automated error-recovery actions to reduce the load on human helpdesks.
- **Constraints:** The product may need to operate without printed manuals, relying solely on digital documentation or in-app guidance.
- **Privacy and Compliance:** Support processes must respect medical data confidentiality (e.g., sanitizing logs before transmission).
- **Multi-Platform Support:** Support must account for different device types (mobile, tablet, PC) and OS versions.
- **Time-Zone and Availability Requirements:** If used across regions, the support availability window must align with healthcare operational hours.

14c. Adaptability Requirements

Content

This section specifies the platforms, deployment environments, and technical ecosystems to which Medalyze must be portable. It includes expectations for running Medalyze on different operating systems, devices (mobile, tablet, desktop), and potential expansion into new clinical or regional markets. It also covers the ability to integrate Medalyze into varied healthcare environments such as clinics, pharmacies, hospitals, and telemedicine environments.

Motivation

To ensure Medalyze remains flexible and usable across multiple healthcare settings. Clinics and pharmacies may use different devices, operating systems, or infrastructure, so Medalyze must be adaptable.

This also ensures that long-term product planning aligns with future expansion into new markets, device types, or regulatory regions.

Examples

- Medalyze shall support deployment on modern web browsers across Windows, macOS, and Linux systems used by clinics and pharmacies.
- Medalyze shall include a mobile-responsive interface to support use on tablets and mobile devices commonly used by doctors during rounds.
- Medalyze is expected to support future deployment in Arabic-speaking and European markets requiring localized formats and healthcare regulations.
- Medalyze is designed for use in clinic offices, but future versions may support ruggedized devices used in mobile healthcare units.

Fit Criterion

- Medalyze must operate correctly on supported browsers defined in the system software specification (Chrome, Firefox, Edge, Safari—latest two stable releases).
- Medalyze must successfully pass compatibility testing for the future expansion environments listed by the product team (e.g., regional healthcare systems, new regulatory zones).
- Medalyze should be portable to new environments within a defined transition period (e.g., no more than 3 months of adaptation work per new supported region or platform).

Considerations

Marketing and product-planning teams must be consulted to uncover assumptions regarding future deployment regions or platforms. The adaptability requirements may also be influenced by future integration with national health systems, telemedicine infrastructure, or pharmacy networks. Additionally, regulatory differences between regions may require UI, data model, or security adaptations.

15. Security Requirements

15a. Access Requirements

Content

Defines which Medalyze users (patients, doctors, pharmacists, administrators) are authorized to access specific system functions and medical data. Specifies the circumstances under which this access is granted, the level of permissions allowed, and which parts of patient records, prescriptions, appointments, or pharmacy data each role may view, modify, or manage.

Motivation

To ensure Medalyze maintains strict confidentiality of medical information, complies with medical privacy regulations, and prevents unauthorized access to sensitive health records. Proper access control protects patient safety, prevents data misuse, and ensures clinical workflows operate securely.

Examples

- Only **assigned doctors** may view or update a patient's medical notes and issue prescriptions for that patient.
- **Pharmacists** may view prescription details but cannot see unrelated medical history unless explicitly required.
- **Patients** may view their own prescriptions, appointment history, and system notifications but may not modify clinical records.
- **System administrators** may manage user accounts and system configurations but cannot access medical data unless required for system integrity.
- Only verified **healthcare staff** can approve medication substitutions.

Fit Criterion

For each Medalyze function or dataset, define:

- The **system function name** or **data object** (e.g., Prescription Record, Appointment Audit Log, Medication Inventory).
- The **roles** (Patient, Doctor, Pharmacist, Admin) authorized to:
  - View
  - Add
  - Modify

  - Delete
- The **conditions** under which access is permitted (e.g., treating physician, active pharmacist license, patient identity verification).
- Any **legal requirements** for access logs and audit trails.

Considerations

- Medical data is inherently sensitive; improper access could result in legal violations and loss of patient trust.
- Some users (patients, pharmacists, junior medical staff) may require **limited visibility** to avoid exposing unnecessary medical details.
- Administrative staff may require system-level control but **must not** access clinical information.
- Access control rules must prevent abuses such as unauthorized prescription edits or viewing records of non-assigned patients.
- Security design (authentication, encryption, MFA, etc.) must not be included here—this section describes **requirements**, not the technical solution.

15b. Integrity Requirements

Content

Specifies the level of data and system integrity that Medalyze must maintain across all medical operations. This includes ensuring that patient records, prescriptions, appointment logs, pharmacy inventory data, and communication messages remain complete, unaltered, and consistent. Integrity requirements also define how Medalyze must protect its data from corruption caused by system errors, unauthorized manipulation, technical faults, or unintended user actions.

Motivation

Because Medalyze handles sensitive healthcare information, any loss, corruption, or alteration of data could directly harm patients, disrupt clinical workflows, or break legal compliance. Maintaining high integrity is essential for protecting patient safety, ensuring doctors can rely on accurate information, and preventing pharmacies from dispensing incorrect medications. These requirements clarify expectations for safeguarding data during normal operation and during unexpected events such as cyberattacks or system misuse.

Examples

- Medalyze shall prevent invalid, incomplete, or contradictory medical information from being stored (e.g., prescriptions with unavailable medications or appointments with conflicting times).
- Medalyze shall protect all patient records, prescriptions, and communication logs from intentional alteration by unauthorized users.
- The system shall validate all updates to medical data to ensure they comply with clinical, legal, and structural integrity rules.
- Medalyze shall maintain accurate history logs of all changes to prescriptions, appointments, and inventory statuses.

Considerations

Healthcare organizations rely heavily on the accuracy and completeness of stored medical data. Damage, corruption, or loss of this information—whether caused by system failures, cyberattacks, or simple user mistakes—can jeopardize patient safety, expose the organization to legal consequences, or interrupt essential clinical services.

Integrity requirements for Medalyze must therefore ensure:

- protection against accidental data loss,
- safeguards against unauthorized or inappropriate actions by permitted users,
- resilience against external attacks aiming to modify or delete records.

Medalyze should also support backup, recovery, and fail-safe mechanisms to ensure continuity of accurate data even after system disruptions.

15c. Privacy Requirements

Content

This section specifies what Medalyze must do to ensure the **privacy of all individuals whose information it stores**, including patients, clinicians, and administrative users. Because Medalyze processes sensitive medical data, it must strictly adhere to applicable **health data privacy laws** (such as GDPR, HIPAA-equivalent regional laws, or local healthcare data protection regulations).

Privacy requirements include:

- protection of personal and medical information
- lawful collection and processing of data
- mechanisms for users to view, modify, or request correction of their data
- prevention of unauthorized disclosure

Medalyze must ensure that all handling of personal and medical information complies with both legal requirements and organizational privacy policies.

Motivation

These are the requirements.

- complies with all relevant privacy laws and standards
- protects the confidentiality of sensitive patient information
- earns and maintains user trust
- avoids legal penalties and reputational harm
- demonstrates responsible handling of health data

Modern users and healthcare providers expect strict privacy protections.

Failure to meet these expectations would reduce trust in Medalyze and could jeopardize its adoption.

Examples

- The product shall notify users of its data collection and usage practices before any personal information is collected.
- The system shall inform users whenever the privacy policy is updated.
- Medalyze shall disclose private data only in accordance with the organization’s approved information policy and applicable law.
- The product shall provide patients with the ability to view their stored personal and medical information.
- When allowed by regulation, users shall be able to request correction of inaccurate personal data.
- Access to sensitive information shall be restricted to authorized roles only (e.g., clinicians, pharmacists).
- The system shall log all access to personal health information for auditing and legal compliance.

Considerations

- Privacy requirements may have significant legal implications; coordination with legal advisors is essential when defining them.
- Determine the notices and consent mechanisms required before collecting patient information, especially medical records or sensitive identifiers.
- Consider how users are kept aware that their data is stored, processed, or shared, and how they can withdraw consent when applicable.
- Ensure that patients can review their personal and medical data, request corrections, or raise privacy concerns.
- Evaluate where especially sensitive data (e.g., payment methods, lab results, diagnoses) needs additional protection measures.
- Combine privacy requirements with system security controls (authentication, access control, encryption) to safeguard data integrity and confidentiality.

15d. Audit Requirements

Content

This section defines the requirements for Medalyze to **retain records and log activities** in a way that supports necessary audit checks. Because Medalyze processes medical information, schedules, prescriptions, payments, and clinical actions, comprehensive auditing is essential for:

- legal compliance
- healthcare accreditation
- internal oversight
- security investigations
- traceability of clinical decisions

Audit requirements cover:

- logging all access to patient data
- logging all modifications to records
- recording user identity, timestamp, and action performed
- retaining audit logs for the legally required duration
- preventing alteration or deletion of audit logs

These records must be stored securely and be retrievable for authorized auditors or regulatory reviews.

Motivation

Audit requirements ensure that Medalyze:

- complies with healthcare audit regulations and organizational policies
- enables accountability for all users and all system actions
- supports investigations into unauthorized access or data misuse
- provides traceability for medical decisions and prescription actions
- prevents users from denying their actions ("non-repudiation")

A robust audit trail increases trust in the system and ensures that Medalyze meets the operational and legal standards expected in healthcare environments.

Considerations

- Audit requirements may have **legal implications**. Consultation with legal advisors and the organization's auditing department is essential to ensure compliance with health data regulations.

- Determine the minimum retention period for audit logs (e.g., 5–10 years depending on jurisdiction).

- Consider whether the system must log:

- who accessed patient information

- what data was viewed or modified

- when key actions (e.g., prescriptions, diagnosis entries, appointment approvals) took place

- failed login attempts and security alerts

- Audit logs must be protected from unauthorized access, alteration, or deletion.

- The auditing system should support investigators in confirming that a user did or did not perform a given action, ensuring accountability and preventing repudiation.

15e. Immunity Requirements

Content

This section defines the requirements that ensure Medalyze can **protect itself from unauthorized or harmful software interference**. Because Medalyze stores medical records and interacts with external healthcare systems, it must defend against threats such as:

- viruses
- worms
- malware
- ransomware
- spyware
- malicious scripts
- unauthorized system modifications
- exploit attempts targeting vulnerabilities

Immunity requirements specify how Medalyze prevents, detects, and responds to these threats to keep the platform secure and operational.

Motivation

The goal of these requirements is to ensure that Medalyze remains:

- **secure** against malicious attacks
- **stable and reliable** for healthcare providers and patients
- **protected** from unauthorized software or harmful code
- **compliant** with cybersecurity expectations in the medical industry

Healthcare platforms are common cyberattack targets. Immunity protections are essential to safeguard patient data, prevent system downtime, and maintain trust in Medalyze as a safe clinical tool.

Considerations

When defining immunity requirements for Medalyze, consider:

- Regular and secure updates to patch vulnerabilities
- Real-time scanning of uploaded files for malicious content
- Input sanitization to prevent injection attacks
- Intrusion detection systems to monitor suspicious activity
- Restrictions on third-party software or plugins
- Secure handling of API requests and external system data
- Ensuring that malicious actors cannot tamper with logs, patient records, or system configuration

Users expect a healthcare system to be resistant to outside interference, and immunity safeguards help ensure the integrity and confidentiality of medical information.

16. Cultural Requirements

16a. Cultural Requirements

Content

This section identifies the cultural considerations that Medalyze must accommodate to ensure the product is acceptable and appropriate for users from different regions, backgrounds, and social groups. Because Medalyze may be used by patients, clinicians, and organizations across multiple countries or cultures, the system must respect cultural norms and avoid creating discomfort or misunderstanding.

Cultural requirements may include:

- appropriate use of colors, symbols, and icons
- sensitivity to religious or social norms
- regional healthcare expectations
- recognition of culturally specific terminology

- alignment with local practices related to time, dates, holidays, and working hours
- avoidance of phrases, imagery, or workflows that may be culturally inappropriate

These requirements help ensure Medalyze feels trustworthy and respectful regardless of the user's cultural background.

Motivation

Cultural requirements exist to:

- reveal expectations or sensitivities that may not be obvious to developers
- ensure Medalyze is accepted in regions with different norms or customs
- avoid unintentionally offending or confusing users
- support successful adoption in diverse healthcare environments

By explicitly identifying cultural needs, Medalyze becomes more inclusive and globally accessible.

Examples

- The product shall not display symbols, images, or color combinations that may be offensive or inappropriate to specific cultural or religious groups.
- Medalyze shall support right-to-left languages where culturally required.
- The system shall recognize local public holidays when scheduling appointments or sending reminders.
- Patient-facing terminology shall be adapted to culturally appropriate wording depending on region.
- The product shall avoid medical icons or illustrations that may be considered sensitive in certain cultures unless explicitly permitted.

Considerations

- Identify whether Medalyze will be used in countries, regions, or institutions with cultural norms different from those of the development team.
- Consider differences in:
- holidays and working calendar structures
- communication styles
- acceptable imagery or colors
- medical symbolism
- measurement units (e.g., mg vs. ml emphasis, Fahrenheit vs. Celsius)
- If a requirement appears unusual or unfamiliar, it may reflect a cultural need that must be captured explicitly.
- Consult representatives from each target region to ensure cultural accuracy and respect, especially for patient-facing features.

17. Legal Requirements

17a. Compliance Requirements

Content

The Medalyze system shall comply with the **Egyptian Personal Data Protection Law (Law No. 151/2020)**, which regulates the collection, processing, storage, and sharing of personal and medical information. The system must also follow basic healthcare confidentiality requirements to ensure that only authorized personnel can access sensitive clinical data such as EHRs, prescriptions, and lab results.

Motivation

Compliance with healthcare privacy laws is required to prevent legal disputes, fines, operational delays, and violations of patient confidentiality.

Fit Criterion

A legal review will confirm that Medalyze's data-handling procedures, access control mechanisms, and privacy measures meet the requirements of the applicable Egyptian data protection regulations.

Considerations

- Medical data is sensitive and must be handled with strict privacy controls.
- Third-party integrations (pharmacies, insurance agencies) must comply with the same confidentiality standards.
- Copyright applies to Medalyze's user interface, design models, and documentation produced by the team.
- No known external intellectual property is being used, and therefore no infringement risks are anticipated.
- Pending amendments to Egyptian digital health regulations may influence future versions of the system.
- No criminal or tax law implications affect the core system.

17b. Standards Requirements

Content

Medalyze shall comply with widely recognized healthcare and security standards, including:

- **HL7** for structuring and exchanging clinical data (EHRs, lab results, prescriptions).
- **ICD-10** for representing diagnosis codes within medical records.
- **ISO/IEC 27001** principles for secure handling of sensitive health information.
  These standards ensure uniformity, interoperability, and secure communication across the system.

Motivation

Adhering to healthcare and security standards ensures that the system is interoperable with industry practices, reduces integration risks, and avoids rework or compliance delays.

Fit Criterion

A healthcare IT expert or standards compliance reviewer will verify that Medalyze correctly applies HL7 formatting, ICD-10 coding, and ISO-aligned security measures for clinical and personal data.

Considerations

- Healthcare systems typically follow HL7 when interfacing with laboratories, pharmacies, and insurers.

- Security standards guide access control, logging, and encryption.
- No additional industry watchdogs or national standards bodies currently affect the system.

18. Open Issues

List of Open Issues

1.Final approval is needed on the exact data exchange format between doctors and pharmacies.

2.The legal team has not yet confirmed retention periods for medical records.

3.Notification provider (SMS/email) has not been selected or integrated.

4.Role permissions for specialty doctors (e.g., surgeons, dermatologists) need clarification.

5.The clinic has not finalized the required languages for patient-facing interfaces.

6.Backup and recovery procedures are still not defined by IT.

7.No decision yet on whether offline access will be supported for doctors.

Business rules for prescription expiration are not fully documented.

19. Off-the-Shelf Solutions

19a. Ready-Made Products

Content

Several commercial hospital management and clinic management systems exist (e.g., Vezeeta, EasyClinic, DrBridge). These solutions address appointment booking and basic EHR management but do not meet the project's academic scope or integration requirements.

Motivation

To evaluate whether an existing system could be adopted instead of building Medalyze from scratch.

Considerations

- Ready-made solutions are too comprehensive or proprietary.
- Licensing costs and API limitations make them unsuitable for an academic project.
- Therefore, adopting an off-the-shelf product is not recommended.

19b. Reusable Components

Content

Reusable components that may be incorporated include:

- Authentication libraries (login system).
- Notification service components (email sender).
- Pre-made UI frameworks (Bootstrap, Material UI).
- Basic scheduling/calendar libraries for appointment management.

Motivation

Reuse reduces development effort and prevents reinventing common technical features.

19c. Products That Can Be Copied

Content

Medalyze may draw inspiration from existing medical portals such as MyChart or Vezeeta for general interface flow, but no code or proprietary content will be copied. Only conceptual workflow ideas may be reused to streamline analysis and design.

Motivation

To speed up requirements analysis by learning from existing solutions without violating intellectual property.

Considerations

- Only user interface patterns and workflow ideas may be referenced.
- No copyrighted or proprietary system designs will be reused.

20. New Problems

20a. Effects on the Current Environment

Content

Introducing Medalyze will transition stakeholders from paper-based workflows to digital ones. This may require training of administrative staff, doctors, and pharmacy employees. No physical infrastructure will be disrupted.

Motivation

To understand how the new system will impact current processes.

Considerations

- Digital transformation may require adjustment time.
- Staff unfamiliar with technology may experience initial difficulty.

20b. Effects on the Installed Systems

Content

No existing digital system is currently installed, but the new system must interact with simulated external systems such as pharmacies and insurance agencies. These interactions will be handled through mock APIs.

Motivation

To prevent integration conflicts.

Form

Interfaces must support secure data exchange for prescriptions and billing.

20c. Potential User Problems

Content

Users with low technical literacy may struggle with navigation in the early stages. Doctors may resist adopting electronic notes over handwritten ones. Patients may misunderstand automated medical warnings.

Motivation

To anticipate negative reactions and prepare mitigation strategies.

20d. Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Content

The system depends on an active internet connection. Slow or unstable networks may affect performance. Limited access to modern devices may also reduce usability for some users.

Motivation

To identify environmental risks early.

20e. Follow-Up Problems

Content

High user adoption may increase server load in the future, necessitating scalability improvements. Additional legal changes in healthcare may require system updates.

Motivation

To prepare for scenarios that may otherwise cause system malfunction or compliance issues.

21. Tasks

21a. Project Planning

- content

  The Medalyze system will be developed following an iterative and incremental approach. Each iteration focuses on refining requirements and adding functional components such as user authentication, doctor–patient interactions, E-prescribing, and appointment management. Early prototypes will include the core workflow: login, patient records, and basic doctor interactions.
- Motivation

  This approach allows continuous stakeholder feedback from doctors, patients, and administrators, ensuring the system aligns with real healthcare workflow needs.
- Considerations

  1.Identifying all actors: Patient, Doctor, Pharmacist, System Administrator.

  2.Defining essential services: E-Prescription generation, viewing medical history, scheduling appointments.

  3.Planning integration with external systems (if needed), such as pharmacy lookup services.

  4.Considering privacy and confidentiality constraints for medical data.

21b. Planning of the Development Phases

Content

- **Phase 1 — Requirements Gathering & System Modeling**

  Deliverables: Use Case Diagram (e.g., “Send E-Prescription,” “Manage Patient File”), Activity Diagrams, Sequence Diagrams

  Benefit: Clarifies exactly how medical workflows map into the software

  Required Date: According to project schedule

- **Phase 2 — System & Database Design**
  Deliverables: Class Diagram (Patient, Doctor, Prescription, Medication, Appointment), ERD, Data Types, Constraints
  Benefit: A stable structure to support medical record consistency and prescription accuracy
- **Phase 3 — Implementation of the First Working Prototype**
  Deliverables: A functioning login system, basic patient record viewing, and test prescription entry
  Benefit: Confirms feasibility and exposes workflow issues early
- **Phase 4 — Full Functional Implementation**
  Deliverables: E-Prescription module, appointment scheduler, doctor dashboard, admin management tools
  Benefit: The system becomes usable for real healthcare processes
- **Phase 5 — Testing, Validation & Deployment**
  Deliverables: Test cases, medical workflow validation, role-based access checks
  Benefit: Ensures the system behaves safely and reliably in a sensitive medical environment

22. Migration to the New Product

22a. Requirements for Migration to the New Product

Content

Medalyze is a new system replacing manual workflows. No digital migration is required. Users will begin entering data directly into the new product once deployed. Training sessions may be required for staff and administrators.

Motivation

To clarify that deployment requires only adoption, not data conversion.

Considerations

- Implementation will follow a single-phase rollout.
- No need for parallel operation with an old system.
- No staff changes or decommissioning activities required.

22b. Data That Has to Be Modified or Translated for the New System

Content

No existing system is being replaced; therefore, no data translation or conversion tasks are required.

Motivation

To confirm that the project scope is not impacted by data migration concerns.

Form

Data will be entered from scratch during the initial use of Medalyze.

23. Risks

Content

A list of the major risks that may affect the Medalyze project, including their probability and the planned mitigation strategy:

Data privacy breaches — Medium probability

Integration failures with hospital systems (EHR/LIMS) — Medium–High probability

User resistance from doctors/nurses due to workflow change — Medium probability

Inaccurate or incomplete requirements due to limited stakeholder involvement — High probability

Performance issues under high load (peak hospital hours) — Medium probability

Motivation

To identify, document, and manage the greatest threats that may impact Medalyze's success, especially because it handles healthcare data, integrates with other systems, and is used in a fast-paced clinical environment.

Considerations

Risks may change during development — review continuously. Stakeholder participation is essential for identifying emerging risks.

Use Medalyze’s requirements model to locate risk sources tied to system features (login, patient record access, lab results, etc.).

Failure to protect patient data will immediately invalidate the entire project.

Form

A Risk Log containing:

- Risk name
- Description
- Probability
- Impact
- Mitigation plan
- Owner responsible

24. Costs

The estimated cost elements for building Medalyze:

Development costs (design, coding, testing)

Integration costs (connecting Medalyze with real hospital databases, lab systems)

Training costs for doctors, nurses, and administrative staff

Maintenance & support costs

Security and compliance costs (penetration testing, encryption, audits)

Infrastructure costs (cloud servers, storage, backup solutions)

Cost clarity ensures that stakeholders understand trade-offs between features, performance, and budget.

25. User Documentation and Training

25a. User Documentation Requirements

Content

This section describes all documentation that Medalyze must provide to support users in operating the system.
Documentation includes:

User Manuals for each user role (Doctor, Patient, Admin, Lab Technician, Pharmacist).

Quick-reference guides for common tasks (booking appointments, uploading lab results, viewing medical history).

Contextual in-system help (tooltips, guided walkthroughs).

Installation and access instructions for new users.

API documentation for integration teams (if external systems connect to Medalyze).

Motivation

To ensure that all categories of Medalyze users can properly understand, access, and utilize the system’s features without confusion or excessive support requests.
Clear documentation reduces training effort, improves adoption, and ensures safety and accuracy when managing healthcare data.

Considerations

Medalyze involves multiple user types, each requiring tailored explanations.

Healthcare environments require simplicity and low cognitive load due to time pressure.

Documentation must reflect privacy and security constraints, especially when dealing with patient-sensitive data.

Updates to the system will require corresponding documentation updates.

Some users (patients) may have low technical literacy, so documentation must avoid jargon.

Use of diagrams, screenshots, and step-by-step flows improves comprehension.

Form

A structured package containing:

- In-app interactive onboarding
- Short video walkthroughs
- Web-based searchable help center

25b. Training Requirements

Content

This section describes the training resources and activities required to prepare Medalyze users to operate the system effectively. Training must cover:

Instructor-led training for medical and administrative staff

Hands-on practice sessions with sample data

Role-specific training modules (e.g., how doctors update medical records, how admins manage user accounts)

Security and privacy training related to data handling

Training assessment (quizzes or practical tasks)

Motivation

Healthcare systems require precise usage, as errors can directly impact patient safety.

Training ensures:

- Accurate data entry
- Consistent workflows across healthcare staff
- Reduced system misuse
- Faster adoption and reduced resistance to new technology
- Compliance with security and privacy requirements

Considerations

Training must accommodate different user backgrounds:

Doctors: time-constrained, need concise task-oriented training

Admins: need deep system understanding

Patients: require simple mobile-friendly guides

Some users may require remote or asynchronous training

Training materials must be updated with each system release

Assessment results may need to be recorded for compliance

Realistic sample data should be used without violating real patient privacy

Form

A structured training program that includes:

Live workshops

E-learning modules

Practice sandbox environment

Certification checklists

Training completion reports for management

26. Waiting Room

Content

A list of:

Future ideas

Deferred requirements

Feature requests not included in current scope

Nice-to-have enhancements that may be added late

Motivation

To track ideas that are valuable but cannot be included in the current release due to constraints (budget, time, technology). Ensures ideas are not forgotten and can be prioritized in future phases.

Considerations

Items must never mix with confirmed requirements.

Review regularly during planning cycles.

Some technologies (AI, wearables) may become more feasible later. Stakeholders should not assume Waiting Room items are guaranteed

Form

A simple table including:

Deferred feature

Reason for deferral

Expected future value

Dependencies

Priority for next release

27. Ideas for Solutions

Content

Potential approaches and high-level solution ideas for implementing Medalyze (not detailed designs):

Modular architecture separating patients, labs, authentication, and reporting

Role-based access control for secure permission handling

Cloud-based backend for scalability

Responsive, accessible web UI for clinical users

Audit logging system for every access to patient data

API gateway for integration with lab devices or hospital EHR systems

Encrypted database storage for sensitive patient records

Local caching for offline mode.

Motivation

To spark innovation and guide the design team by presenting possible solution patterns without forcing a specific design. Helps stakeholders visualize what is possible and the technical direction.

Considerations

These ideas are not requirements — they are optional design inspirations.

Some ideas may violate constraints and must be reviewed.

Implementation feasibility depends on resources, tech stack, and integration complexity.

Form

A list of architecture sketches, diagrams, or design concepts that may be refined later, such as:

System architecture diagram

UI wireframes

API integration outline

Security model concept