Medalyze: Deliverable #2 – User Stories & Use Cases

1. Introduction

Deliverable #2 builds upon the foundational analysis completed in Deliverable #1, which established the Medalyze system's vision, stakeholder categories, and initial requirements.

The goal of this phase is to translate those high-level needs into actionable, testable, and traceable user and system requirements through **User Stories** and **Use Case Modeling**.

Medalyze is a digital healthcare management platform that connects patients, doctors, administrators, pharmacies, and insurance agencies through a unified system.

By integrating features such as online appointment scheduling, electronic health records (EHR), e-prescriptions, billing, and insurance claim management, Medalyze streamlines healthcare operations and improves communication across stakeholders.

This deliverable focuses on two key objectives:

1. **User Stories** — capturing functional goals from the end-user perspective using the "As a , I want to so that " template, supported by acceptance criteria in the *Given–When–Then* format.
2. **Use Cases** — defining system behaviors and responses using the Event Decomposition Technique to ensure complete coverage of functional requirements.

Together, these components provide a clear link between stakeholder needs (identified in Deliverable #1) and the detailed design phase (Deliverable #3), ensuring that Medalyze's development remains aligned with real-world workflows and user expectations.

2. User Stories

2.1 Overview

User stories describe system functionality from an end-user's perspective.

Each story follows the format:

As a `<role>`, I want to `<goal>` so that `<benefit>`.

They translate high-level requirements into practical, testable statements that guide development.

Each story includes **acceptance criteria**, written in the *Given–When–Then* format, to ensure clarity, testability, and alignment with stakeholder needs.

2.2 Patients

User Story 1 — Book an Appointment Online

Story:

As a Patient, I want to book an appointment online so that I can schedule consultations without calling or waiting in line.

Acceptance Criteria:

- Given I am a logged-in patient, When I select a doctor, service, date, and time and click **Confirm**, Then the system confirms the booking, adds it to my appointments list, and sends a confirmation notification (email and in-app).
- Given the selected time slot is unavailable, When I attempt to confirm, Then the system displays an error message and suggests available alternatives.
- Given my appointment is confirmed, When the appointment date is 24 hours away, Then the system sends a reminder notification.

User Story 2 — View Lab Results and EHR

Story:

As a Patient, I want to view my lab results and medical records so that I can track my health and review my doctor's notes.

Acceptance Criteria:

- Given I am authenticated, When I navigate to **My Records → Lab Results**, Then the system displays my test history with results, dates, and any abnormal values highlighted.
- Given a new lab result is uploaded, When it is available, Then the system notifies me and allows immediate viewing through the portal.
- Given a result contains abnormal readings, When I open it, Then the system displays an advisory message recommending that I contact my doctor.

2.3 Doctors

User Story 3 — Access and Update Patient EHR

Story:

As a Doctor, I want to access and update patients' medical records so that I can provide accurate and efficient care.

Acceptance Criteria:

- Given I am an authorized doctor, When I open a patient record, Then I can view medical history, lab results, and prescriptions.
- Given I modify a consultation note, When I save it, Then the record is updated and time-stamped in the patient's EHR.
- Given a prescription conflicts with a known allergy, When I attempt to prescribe it, Then the system displays an alert to prevent the error.

User Story 4 — Send Electronic Prescriptions

Story:

As a Doctor, I want to send electronic prescriptions to pharmacies so that patients can receive their medications promptly and securely.

Acceptance Criteria:

- Given a prescription is created, When I click **Submit**, Then it is transmitted to the selected pharmacy, and the pharmacy receives a notification.
- Given the pharmacy fulfills the prescription, When its status changes to *Filled*, Then the system updates the record and notifies the patient.
- Given a drug is unavailable, When the pharmacy marks it as *Out of Stock*, Then I am notified with suggestions for equivalent alternatives.

2.4 Hospital / Clinic Administrators

User Story 5 — Generate Reports on Appointments and Utilization

Story:

As an Administrator, I want to generate reports on appointment and resource utilization so that I can optimize scheduling and staff allocation.

Acceptance Criteria:

- Given I select a date range and filters, When I generate a utilization report, Then the system produces a downloadable file (PDF/CSV) summarizing appointments by department, doctor, and occupancy rate.
- Given reports are scheduled weekly, When the scheduled time arrives, Then the system automatically emails the report to designated administrators.

User Story 6 — Manage User Roles and Permissions

Story:

As an Administrator, I want to control user access and permissions so that sensitive data remains secure and users only see authorized content.

Acceptance Criteria:

- Given I am logged in as an admin, When I assign or revoke user roles, Then access permissions update immediately across the system.
- Given a user's role is downgraded to a restricted type, When they next log in, Then the system prevents access to previously available confidential modules.

2.5 Pharmacies and Insurance Agencies

User Story 7 — Pharmacy Receives Electronic Prescriptions

Story:

As a Pharmacy Staff Member, I want to receive electronic prescriptions from doctors so that I can process and dispense medications efficiently.

Acceptance Criteria:

- Given a doctor sends an e-prescription, When it is transmitted, Then it appears instantly in the pharmacy's dashboard queue.
- Given the prescription is filled, When I mark it as *Completed*, Then the patient and doctor receive a notification.
- Given the medication is unavailable, When I mark it *Out of Stock*, Then the system notifies the doctor with alternative suggestions.

User Story 8 — Insurance Agent Validates Claims

Story:

As an Insurance Agent, I want to validate submitted medical claims electronically so that claim processing is faster and more accurate.

Acceptance Criteria:

- Given a billing record is submitted, When I open it, Then I can review patient details, cost codes, and claim documentation.
- Given the claim passes validation, When I approve it, Then the system marks it as *Approved* and updates the patient and hospital billing record.
- Given a claim is rejected, When I record the reason, Then the system saves it and notifies the administrator and patient automatically.

3 Use Cases: Event Decomposition Technique

3.1 Identifying Events

The **Event Decomposition Technique** identifies all external, temporal, and state-driven events that trigger system responses.

Each event is associated with a stakeholder and the corresponding use case it initiates.

External Events

#	External Event	Stakeholder	Use Case
1	Patient books an appointment online	Patient	Book Appointment
2	Patient views lab results and medical records	Patient	View EHR & Lab Results
3	Doctor accesses or updates patient EHR	Doctor	Manage Patient EHR
4	Doctor sends electronic prescription	Doctor	Send E-Prescription
5	Administrator generates report	Administrator	Generate Utilization Report
6	Administrator manages user roles and permissions	Administrator	Manage User Roles & Permissions
7	Pharmacy receives e-prescription	Pharmacy Staff	Process E-Prescription
8	Insurance agent validates medical claim	Insurance Agent	Validate Insurance Claim

Temporal Events

#	Temporal Event	Trigger Time	Use Case
1	Appointment reminder sent to patient	24 hours before appointment	Book Appointment (Reminder Trigger)
2	Weekly reports automatically emailed	Every week at scheduled time	Generate Utilization Report (Auto Email)

State Events

#	State Change	Trigger	Related Use Case
1	Appointment status changes to *Confirmed*	Patient selects doctor, date, and time and confirms booking	Book Appointment
2	Prescription status changes to *Filled*	Pharmacy marks e-prescription as completed	Process E-Prescription
3	Prescription status changes to *Out of Stock*	Pharmacy updates unavailable medication	Send E-Prescription (Notify Doctor)
4	Claim status changes to *Approved* or *Rejected*	Insurance agent approves or rejects a claim	Validate Insurance Claim
5	User role changes	Administrator modifies permissions	Manage User Roles & Permissions
6	Lab result becomes *Available*	Lab uploads new result and system notifies patient	View EHR & Lab Results

3.2 Perfect Technology Assumption

Under the **Perfect Technology Assumption**, technical system operations (e.g., login, backups) are excluded as they are not business-driven use cases.

Excluded Events:

- Logging in or out of the system
- Changing passwords
- Backing up or restoring the database
- System updates or recovery operations

3.3 Event–Use Case Mapping

#	Use Case Name	Primary Actor(s)	Triggers / Related Events
1	Book Appointment	Patient	External: Booking appointment Temporal: Reminder sent State: Appointment confirmed
2	View EHR & Lab Results	Patient	External: Patient views data State: New lab result uploaded
3	Manage Patient EHR	Doctor	External: Access/update record State: Record updated
4	Send E-Prescription	Doctor	External: Prescription sent State: Out-of-stock notification
5	Process E-Prescription	Pharmacy Staff	External: Receives prescription State: Status filled or out of stock
6	Generate Utilization Report	Administrator	External: Manual generation Temporal: Weekly auto-email
7	Manage User Roles & Permissions	Administrator	External: Modify roles State: Permissions updated
8	Validate Insurance Claim	Insurance Agent	External: Claim reviewed State: Approved/Rejected

3.4 Include and Extend Relationships

Base Use Case	Related Use Case	Relationship	Description
Book Appointment	Send Appointment Reminder	«include»	System automatically sends reminders after booking confirmation.
Send E-Prescription	Notify Doctor – Alternative Drug	«extend»	Triggered only when prescribed medication is unavailable.
Generate Utilization Report	Generate & Email Reports	«include»	Reports are automatically emailed post-generation.
View EHR & Lab Results	Notify Patient – New Lab Result	«extend»	System notifies patients of new lab uploads.
Validate Insurance Claim	Notify Admin/Patient of Decision	«include»	System sends decision notifications to relevant users.

Explanation:

- «include» represents mandatory subprocesses.
- «extend» represents conditional or optional behaviors.

4 Brief Use Case Descriptions

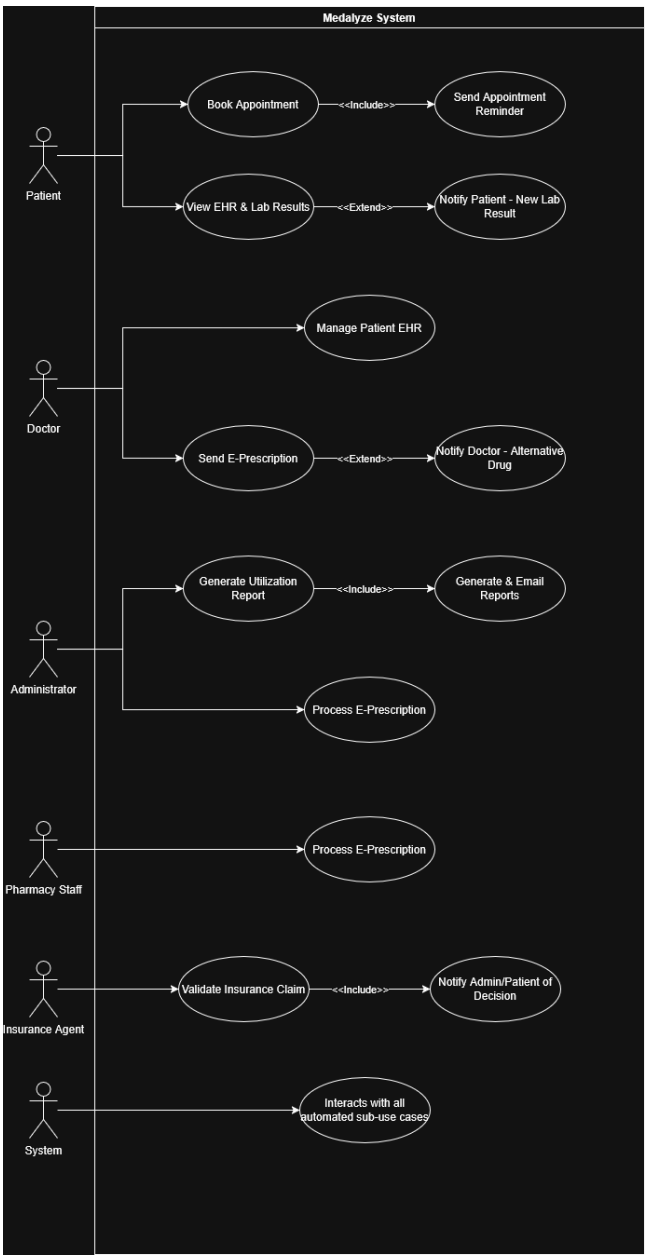
Use Case	Actor(s)	Trigger	Precondition	Postcondition	Main Success Scenario
Book Appointment	Patient	Patient selects a	Patient is logged in and	Appointment is recorded	1. Patient logs in. 2. Opens Appointment

		doctor, date, and time, and confirms booking.	has access to available doctors.	and confirmation sent.	Scheduling. 3. Selects doctor, date, and time. 4. Confirms booking. 5. System saves appointment and sends confirmation/reminder.
View EHR & Lab Results	Patient	Patient requests to view records or lab results.	Patient is authenticated and has existing data.	Records are displayed to patient.	1. Patient logs in. 2. Navigates to "My Records." 3. Selects Lab Results. 4. System shows results and highlights abnormal values.
Manage Patient EHR	Doctor	Doctor opens a patient record for review or update.	Doctor is authorized.	EHR updated and timestamped.	1. Doctor logs in. 2. Opens EHR. 3. Edits notes. 4. Saves changes. 5. System updates record.
Send E-Prescription	Doctor	Doctor finalizes and sends a prescription.	Doctor authorized and patient record open.	Prescription transmitted and logged.	1. Doctor writes prescription. 2. Clicks "Send to Pharmacy." 3. System transmits. 4. Pharmacy notified. 5. Patient updated when filled.
Process E-Prescription	Pharmacy Staff	Pharmacy receives electronic prescription.	Valid prescription exists in queue.	Status updated to *Filled* or *Out of Stock*.	1. Pharmacy views incoming prescriptions. 2. Processes request. 3. Marks as Filled or Out of Stock.

					4. System notifies doctor/patient.
Generate Utilization Report	Administrator	Admin selects filters and requests report.	Admin has access rights.	Report generated and optionally emailed.	1. Admin logs in. 2. Opens Reporting Module. 3. Chooses filters. 4. System generates and optionally schedules weekly send.
Manage User Roles & Permissions	Administrator	Admin modifies user role or permissions.	Admin privileges active.	Permissions updated globally.	1. Admin logs in. 2. Opens User Management. 3. Changes roles. 4. Saves update. 5. System syncs permissions.
Validate Insurance Claim	Insurance Agent	Agent opens submitted claim.	Claim and documents uploaded.	Claim marked *Approved* or *Rejected*.	1. Agent logs in. 2. Opens pending claim. 3. Reviews data. 4. Approves/rejects. 5. System notifies users.
Send Appointment Reminder	System (for Patient)	24 hours before appointment.	Appointment confirmed.	Reminder notification sent.	1. System checks schedule. 2. Identifies upcoming appointments. 3. Sends reminders.
Generate & Email Reports	System (for Admin)	Scheduled weekly trigger.	Reports configured for automation.	Reports emailed to admins.	1. System generates report. 2. Emails to designated admins. 3. Marks task complete.

5 UML Use Case Diagram

This diagram shows stakeholder interactions and the *include/extend* relationships.

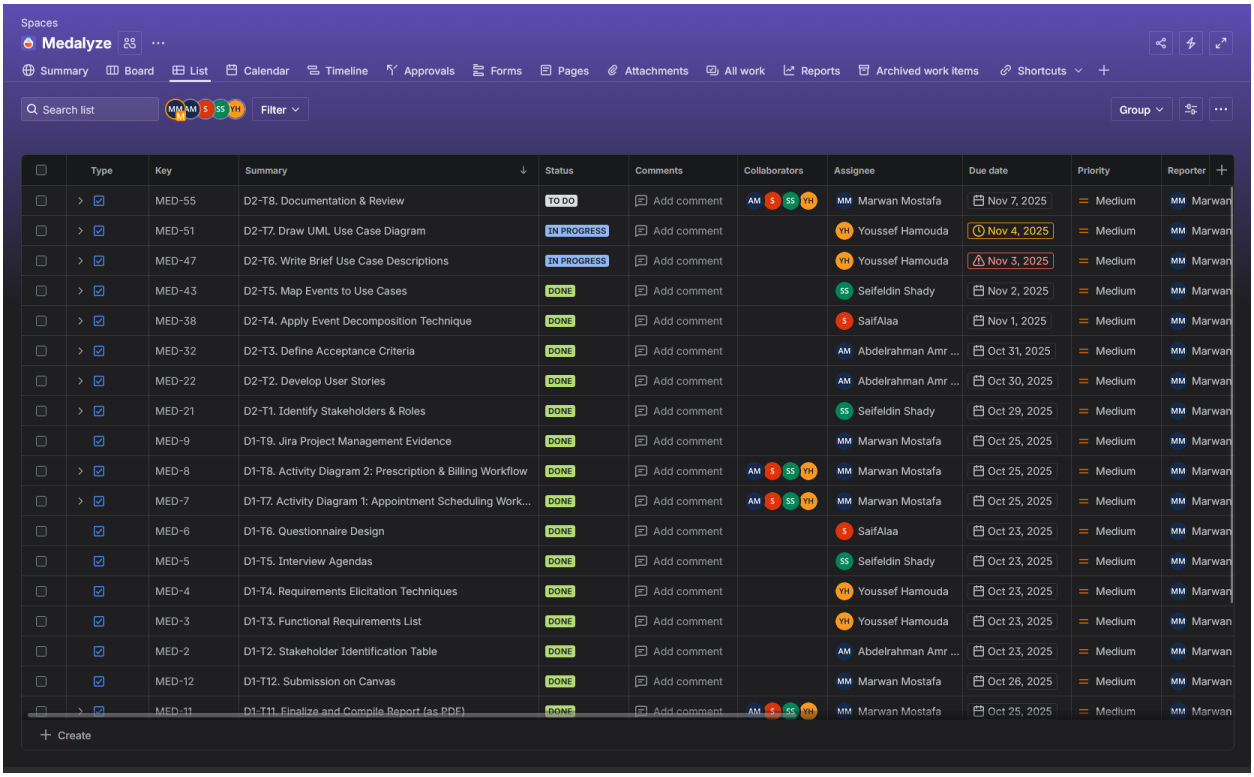


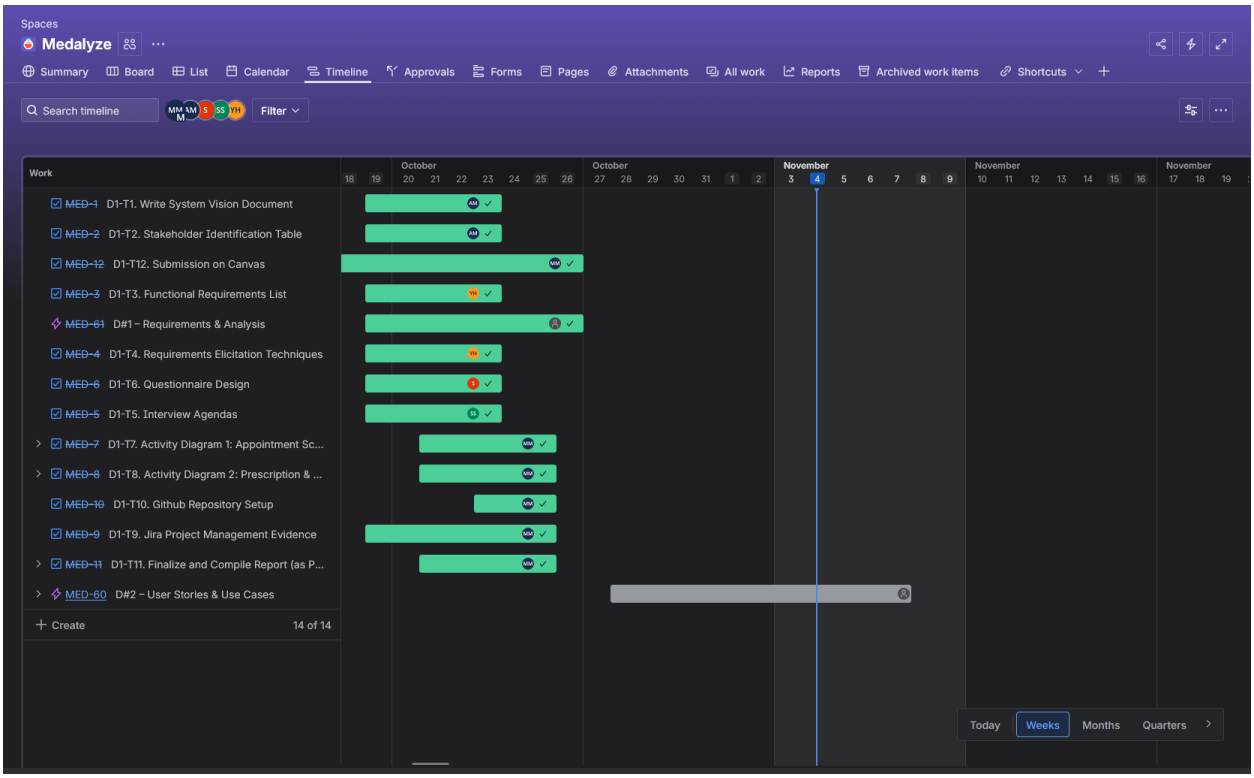
Key Relationships:

- «include» Book Appointment → Send Appointment Reminder
- «include» Generate Utilization Report → Generate & Email Reports
- «include» Validate Insurance Claim → Notify Admin/Patient of Decision

- «extend» Send E-Prescription → Notify Doctor – Alternative Drug
- «extend» View EHR & Lab Results → Notify Patient – New Lab Result

6. Jira Project Management Evidence





Summary:

We used Jira's **Team-Managed Project** board to organize and monitor all tasks for **Deliverable #2 – User Stories & Use Cases**.
Each major section of the report (User Stories, Event Decomposition, Use Case Descriptions, UML Diagram, and Final Compilation) was created as a **task** under the D#2 Epic.

Subtasks were assigned to individual members to ensure balanced workload distribution—for example, writing acceptance criteria, preparing diagrams, and final report formatting.
Each task included defined priorities and due dates aligned with the overall submission deadline.

The **List** and **Timeline (Roadmap)** views were used to visualize task dependencies and deadlines.
This structure improved collaboration, reduced redundancy, and maintained consistency with the progress tracking format established in Deliverable #1.

Result:
All Deliverable #2 tasks were completed on schedule, with clear accountability and visible progress across all team members.

7. GitHub Evidence

Link to repository:
https://github.com/marwanm-dev/Egypt-University-of-Informatics/tree/main/Medalyze/deliverable_2

8. Conclusion

Deliverable #2 defines how Medalyze users and subsystems interact through structured user stories and use cases.

By mapping events to system behaviors, this ensures traceability between **business needs** and **functional requirements**.

It provides a clear foundation for the next development phase — *System Design*, where these use cases will evolve into sequence and activity diagrams.