



My Health FingerPrint

Acknowledgements

Above all, we would like to express our sincere appreciation to our advisor, **Prof. Dr. Sayed Abdelgaber**, for his involvement and guidance throughout our project. His expertise and direction have been instrumental in bringing our project to fruition. Despite the challenging circumstances, we are grateful for his patience and unwavering dedication.

Our gratitude also goes to our **families** for their constant support and encouragement. Their belief in us has been a driving force in our whole journey.

Lastly, we would like to acknowledge the **faculty of Computer Science and Artificial Intelligence at Helwan University** for providing us with supporting courses and creating a conducive learning environment. We are grateful to all the professors who have had a significant impact on our lives through their teaching and guidance.

Table of contents

1 Introduction.....	6
1.1 Overview.....	7
1.2 Objectives.....	7
1.3 Purpose.....	8
1.4 Scope.....	8
1.4.1 Planning.....	8
1.4.2 Designing.....	9
1.4.3 Coding.....	9
1.4.4 Testing.....	10
1.4.5 Documentation.....	10
1.5 General Constraints.....	10
2 Project planning and analysis.....	11
2.1 Project planning.....	12
2.1.1 Feasibility study.....	12
2.1.1.1 Project description.....	12
2.1.1.2 Market analysis (SWOT analysis):.....	14
2.1.2 Estimated cost.....	15
2.1.3 Gantt Chart.....	16
2.2 Analysis and Limitation of existing system.....	19
2.3 Need for the new system.....	19
2.4 Analysis of the new system.....	20
2.4.1 System requirements.....	20
2.4.2 User requirements.....	21
2.4.3 Domain requirements.....	21
2.4.4 Functional requirements - use case scenarios.....	22
2.4.5 Non-functional requirements.....	37
2.5 Advantages of the new system.....	39
2.6 Risk and risk management.....	41
3 Software design.....	43
3.1 Class diagram.....	44
3.2 Use case diagram.....	46
3.3 Sequence diagrams.....	47
3.4 Activity diagram.....	74
4 Implementation.....	80
4.1 Software architecture.....	81
4.2 Flowchart.....	82

5 Testing.....	83
5.1 Functional testing.....	84
5.1.1 Unit Testing.....	84
5.1.2 Integration Testing.....	84
5.1.3 Acceptance Testing.....	84
5.1.4 Regression Testing.....	84
5.1.5 System Testing.....	85
5.2 Non-Functional testing.....	86
5.2.1 Performance Testing.....	86
5.2.2 Stress Testing.....	86
5.2.3 Additional test cases for suspicious behavior.....	87
6 Results and Discussion.....	88
6.1 Results.....	89
6.1.1 Expected Results.....	89
6.1.2 Actual Results.....	92
6.2 Discussion.....	96
7 Conclusion.....	97
8 Future work.....	99
Bibliography.....	101

Abstract

This paper presents the development and implementation of an electronic health record (EHR) system as a Software-as-a-Service (SaaS) solution. The system aims to replace paper-based patient records, improve data accessibility, and assist healthcare providers in decision making. It emphasizes ease of use, data security, and interoperability. The results demonstrate successful implementation and potential for enhancing care delivery. Future work includes improving data visualization and incorporating advanced decision support features.



Chapter 1

Introduction

1.1 Overview

A SaaS that serves mainly as an electronic health record system to be utilizable and adoptable in healthcare facilities of different sizes. The vital part of an EHR system is about substituting paper-based patient records for faster and easier retrieval, and helping healthcare providers make decisions.

1.2 Objectives

To ease and simplify:

- easier documentation and recording
- easy and fast data access and updating
- ease of tracking patient's history
- ease of use and intuitive user interface
- ease of on boarding process for employees

To provide organized information about patients to help decision making

To provide data visualization

To allow data exchange with other systems

To assure data safety and air tight security:

- foolproof identity validation
- data access tracking
- fine-grained authorization (project's main ideology)
- Role-Based Access Control (RBAC)

1.3 Purpose

The system can be said to finally have achieved its purpose when the healthcare process in the organization goes faster, easier, and more successful than it could have gone without adopting the system

1.4 Scope

1.4.1 Planning

Requirements gathering:

Determining the initial features and functionalities the practice requires in an EHR system.

The initial features are expected to be expended and modified anytime at the request of the project supervisors or through Request For Proposal sent from a target customer etc.

Organizing team members:

Organizing the individuals who will be involved in the project documentation and implementation according to the capabilities and hard and soft skills of each

Considering constraints:

Privacy constraints are absolutely taken into consideration (follows Health insurance accountability and portability rule), as well as any legal constraints.

Interoperability with other systems (or subsystems) is taken into consideration.

Defining project deliverables:

The final deliverable product should be an easy-to-use application to carry out the basic EHR system functionalities at the very least, and to be restricted to the constraints

1.4.2 Designing

Determining structural, functional, and high-level external diagrams through considering the user requirements, the final product in a mind image according to these requirements, and the technical requirements to turn that image into reality

1.4.3 Coding

Deciding the programming paradigm:

-Functional programming paradigms

Determining programming languages and editors to be used:

-NodeJs as a back-end environment.

-Modern client-side technologies (React.js, MongoDB, AWS as hosting provider via vercel)

-Any IDE supporting Javascript and modern used technologies

Setting common custom settings between coders for easier communication, understanding, and code integration

Considering all the features, the entities and the diagrams determined, to come up with a flowchart

Writing a pseudocode

Considering any libraries or reusable off-the-shelf codes written in a similar program codes

Actual coding

1.4.4 Testing

Functional testing:

- Unit testing.
- Integration testing.
- Regression testing.
- System testing.
- Acceptance testing.

Non-functional Testing:

- Performance testing.
- Stress testing.
- Load testing.
- Security testing.

1.4.5 Documentation

In addition to this introduction, all the points will be explained to ensure a comprehensive understanding of the system's functionalities. It is important to note that certain functionalities may be labeled as future work, to be implemented after the completion of the graduation phase.

1.5 General Constraints

As it usually goes in graduation projects, where team members are students, some delays took place due to study and exams.



Chapter 2

Project planning and analysis

2.1 Project planning

2.1.1 Feasibility study

Here are the studies and the analysis regarding the achievability and practicality of the project:

2.1.1.1 Project description

An electronic health record (EHR) is a digital version of a patient's paper chart.

EHRs should be reliable and up-to-date to help providers make decisions about a patient's care.

EHR systems are designed to store EHRs, providing the providers with a broader view of a person's health and medical history, and enabling them to make well-informed care decisions quickly to improve care and reduce safety risks.

A patient's medical and social history may include medication and allergies, immunization status, laboratory test results, radiology images, vital signs, in addition to family history

EHR systems allow EHRs to be shared with authorized providers in other organizations like healthcare specialists, pharmacies, laboratories, etc., and hence reduce replication.

EHR systems provide appointment state activation (Through which this system inhibits concurrent access to patient)

EHR systems provide data visualization about the patients

EHR systems allow secure patient data exchange with other EHR systems

This Electronic Health Record (EHR) system enables dynamic formatting of patient forms, with guarantee of data consistency. Users are not limited to specific categories such as symptoms, test results, or medications. Instead, they have the freedom to create and specify any custom category and the content to fit it.

This system will allow providers to view a patient's medical history and receive relevant statistical-based recommendations related to the patient's case. The statistical calculations will be repeated periodically over the system's patient data itself. This will inform about the likelihood of possible diagnosis, medications, etc.. to be related to a specific patient, based on the other patients' data residing in the system.

2.1.1.2 Market analysis (SWOT analysis):

Strengthens:

The coders in the project team are skilled, familiar, and experienced with such projects.

Team members are motivated to give their best as this project is their project of graduation.

The work will be under the supervision of the Information Systems manager and expert: Dr.Prof. Sayed Abdelgaber, who had various publications and projects concerning this field.

It is worth mentioning that Dr. Prof. Sayed Abdelgaber had a project called "A Roadmap to Implement (EHR) Nationwide in Egypt" included in his project list.

Weaknesses:

Team members are still under-graduated

Opportunities:

The health care facilities need to keep up with new technology that speeds up and eases documentation and reporting, and helps improve patient care and decision making.

The product will be providing unique features over the other traditional EHR systems as it will be utilizing Artificial Intelligence techniques for further better decision making.

The product will be adoptable in different types of healthcare facilities (hospitals, clinics, etc.).

The product will offer a capacity-and-feature-limited free trial.

Threats:

Customers may be afraid of security and privacy concerns, especially when they know that one hack may disclose all the documents residing in an electronic system.

Customers may find it not worthy to switch to electronic documentation.

Despite providing unique features, customers who are already adopting electronic documentation may find it difficult or not worthy to convert to any other system.

2.1.2 Estimated cost

For the project completion:

The project team members will not be paid for the software as this project is not customized for a certain customer, and it is already required for their bachelor's degree

Software implementation and testing will be performed on personal computers and free cloud database servers

For Organizations to purchase and set up the system:

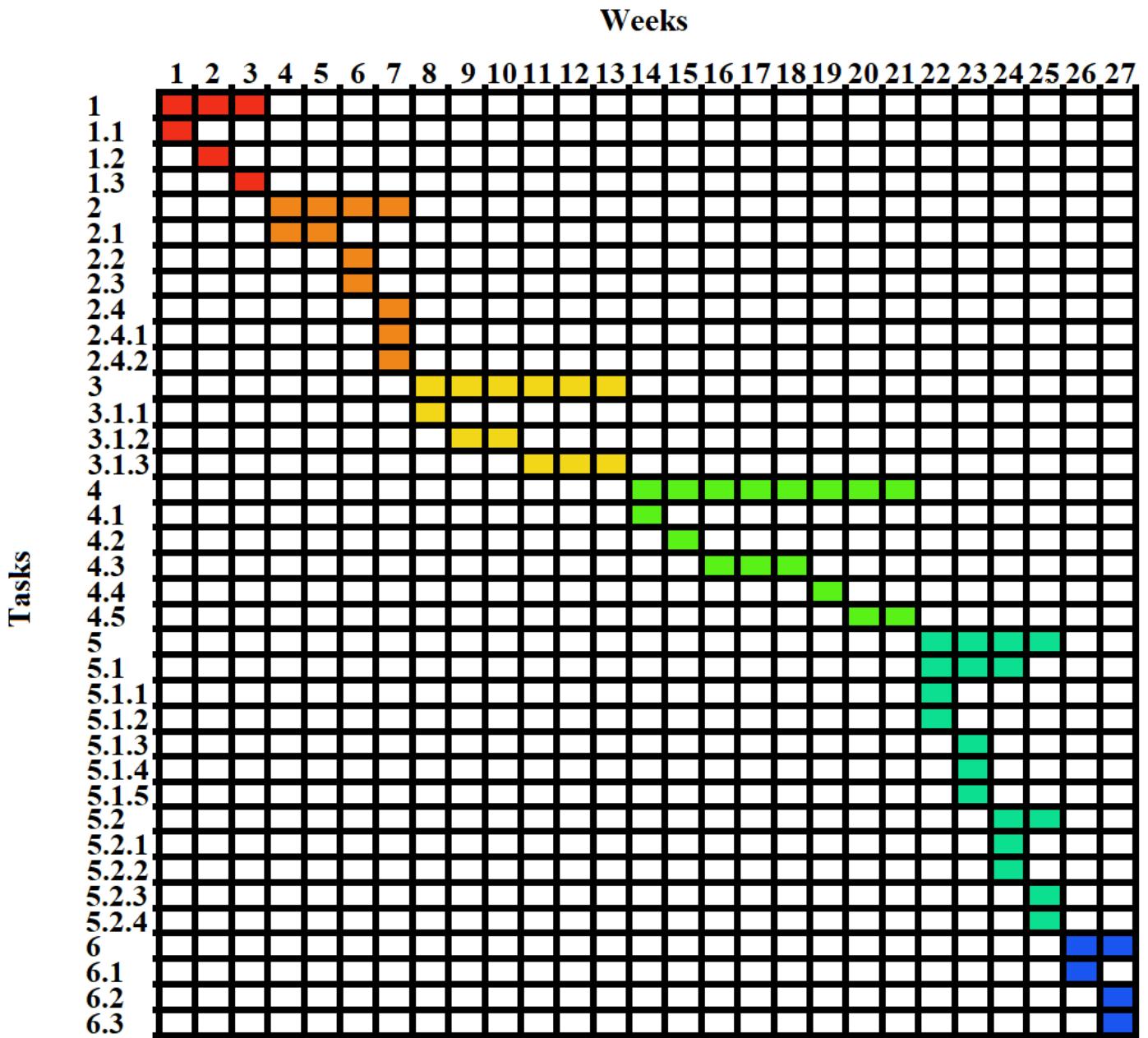
The annual license fee for the software application might be \$8,000 for the access to certain features, but free for the basic ones

For the customers adopting electronic documentation for the first time, there would be a hardware budget, including, essentially, computers and printers, and there would be internet connection budget

The number of hardware components would vary with the size of the organization, however, hardware might cost up to \$30,000 on average

Staff training might take a week and cost around \$3,000

2.1.3 Gantt Chart



1 Surveying for the requirements

- 1.1 Electronic search for EHR systems
- 1.2 Experience interviews with EHR system users
- 1.3 conducting visits to healthcare providers

2 Planning

- 2.1 Identifying the exact features required (functional and non-functional)
- 2.2 Considering technical and budgetary constraints
- 2.3 Defining project deliverables and goals
- 2.4 Team planning
 - 2.4.1 Assigning tasks to team members based on their skills and expertise
 - 2.4.2 Setting out a realistic timeline for the project

3 Design

- 3.1 Identifying UML diagrams to represent the system architecture
 - 3.1.1 Architecture diagram
 - 3.1.2 Structural diagrams
 - 3.1.3 Behavioral diagrams

4 Implementation

- 4.1 Login module
- 4.2 Manager/Sign up module
- 4.3 Receptionist module
- 4.4 Nurse module
- 4.5 Physician module

5 Testing

5.1 Functional testing

5.1.1 Unit testing

5.1.2 Integration testing

5.1.3 System testing

5.1.4 Acceptance testing

5.1.5 Regression testing

5.2 Non-Functional testing

5.2.1 Performance testing

5.2.2 Stress testing

5.2.3 Load testing

5.2.4 Security testing

6 Deployment

6.1 Hosting the system on a domain to offer a limited-capacity free trial

6.2 Creating a pitch video

6.3 Creating a demonstration video

2.2 Analysis and Limitation of existing system

Epic EHR:

Complex user interface:

Due to the inclusion of various different modules and features to cover wide range of needs, the system is said to have a degree of complexity in terms of navigation and use, so it takes longer training and getting used to

Occasional slow operation:

Due to the complexity mentioned, which hence is reflected in the database and its serving, the system might face occasional performance issues

eClinicalWorks:

Complex user interface:

The system's user interface is already criticized by users' reviews for being complex and difficult to navigate, with many options and buttons causing confusion.

Slow operation:

Users also report slow performance. One user reported by saying: "it freezes at a minimum ten times a day requiring the system to be shut-down and re-logged in". While specific factors may contribute to that user's issue, this one with the other issues reported suggest a broader problem with the system's speed and usability.

2.3 Need for the new system

This system will be designed to provide an intuitive and user-friendly interface to perform an operation as simple as it is thought to be. It will not include any redundant, rarely-used, or difficult-to-access features, ensuring that all of the included features optimally support the system's required functionalities and are easy to access and use.

2.4 Analysis of the new system

2.4.1 System requirements

In order to have the system functioning properly, the following requirements must be fulfilled:

- Network devices
- Computer devices
- Printer devices

Software requirements:

- Operating system
- Web browser
- Reliable internet connection

Performance requirements:

- High-speed internet

Security requirements:

- Authentication, authorization, and access control
- Operating system with firewall software installed
- Firewall network router devices
- Back-up database

Compliance requirements:

- Raising staff awareness
- Setting policies

2.4.2 User requirements

The user expects the system to support:

- Ease of use
- Security and privacy
- Accessibility and availability
- Patient data access and manipulation
- During-appointment status flagging
- Data visualization
- Decision support
- Interoperability

2.4.3 Domain requirements

In order to meet the constraints of the domain to be adopting the system, the system must guarantee:

- Security and privacy
 - to comply with legal requirements (ex. HIPAA)
- Ease of use and simplicity
 - to suit the clinical workflow and take less time to be understood

2.4.4 Functional requirements - use case scenarios

Name	Log in
prerequisites	Internet connection Valid national ID
Pre-conditions	User has navigated to login page User is not logged in
Basic flow	1-User enters national ID 2-System shows selectable accounts with the ID entered 3-User selects an account 4-System prompts the user to enter the password for the selected account 5-User enters the password 6-System verifies the provided password and redirects to the home/dashboard page
Alternative flow	If ID is invalid or not found: 1-System prompts a message specifying the issue 2-Step 1 in basic flow If Password is incorrect: 1-System prompts “Wrong password” message 2-Step 5 in basic flow
Post-conditions	User is logged in to his dashboard and is able to use his permissions

Name	Create account within user's organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters new account's national ID, password, and starting role (from roles created in the user's same organization) 2-System saves the account (to the same organization of the user's) and shows “account created” message
Alternative flow	If entered data is invalid (bad password or invalid ID..etc.): 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Account is created in the user's same organization

Name	Create patient profile
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to patients page
Basic flow	1-User enters new patient's ID in search field 2-System finds not the patient and suggests to add 3-User adds the ID 4-System saves and enables to manipulate patient profile 5-User adds the rest of patient data (first name, last name, phone number, phone number of first responders, address, secondary address (optional)) and saves 2- System shows "profile saved" message
Alternative flow	If entered data is invalid (incomplete fields or invalid ID..etc.) 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Patient profile is created

Name	View a patient's profile
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to patients page
Basic flow	1-User enters patient's ID 2-The system searches for the patient and brings up the data in the form of tabs 3-User opens the profile tab 4-System shows profile information
Alternative flow	If entered ID is invalid or not found 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Desired patient's Profile is viewed

Name	Add a patient's medical information
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to patients page
Basic flow	<p>1-User enters patient's ID</p> <p>2-The system searches for the patient and brings up the data in the form of tabs</p> <p>3-User opens the medical record tab</p> <p>4-The system displays the medical record in form of tabs (symptoms, medications, allergies.. etc.) each with (+) sign</p> <p>5-User opens the desired tab and clicks on (+) sign</p> <p>6-The system utilizes the fields of precreated collectios (a collection for each tab) to take the info. (in addition, it may need to retrieve data from a collection for a dropdown list if specified at collection creation)</p> <p>7-User enters information.</p> <p>The system saves the new information along with the date and organization of entry.</p>
Alternative flow	If entered data is invalid (incomplete) 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Medical information is added for the desired patient

Name	View a patient's medical record
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to patients page
Basic flow	1-User enters patient's ID 2-The system searches for the patient and brings up the data in the form of tabs 3-User opens the medical record tab 4-System shows medical record. A list of previously added symptoms, medications, test results, and others 5-User opens the tab he wants to view and can search for a target information
Alternative flow	If entered ID is invalid or not found 1-System prompts a message specifying the issue Step 1 in basic flow If there is no available information 1-System Shows "none" page
Post-conditions	Desired patient's medical record is viewed

Name	Update a patient's profile
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to patients page
Basic flow	1-User enters patient's ID 2-The system searches for the patient and brings up the data in the form of tabs 3-User opens profile tab 4-System shows profile information updateable (except ID) 5-User enters new data 6-System updates
Alternative flow	If entered value is invalid: 1-System prompts a message specifying the issue 2-Step 5 in basic flow
Post-conditions	Patient profile is updated

Name	View account in user's organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters the account's national ID 2-System finds the account in the organization and previews
Alternative flow	If entered data is invalid (bad password or invalid or not found ID..etc.): 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Desired account is viewed

Name	View account
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters the account's national ID 2-System finds the account in the whole system and previews
Alternative flow	If entered data is invalid (bad password or invalid or not found ID..etc.): 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Desired account is viewed

Name	Formulate Restricted Role
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to roles page
Basic flow	System shows a list of permissions (excluding admin privilege permissions) User picks permissions he wants to from a role by, and enters the role name and submits System saves the role to the user's organization
Alternative flow	--
Post-conditions	Role is saved in the user's same organization

Name	Assign role to account in user's organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to account roles page
Basic flow	1-User enters account's ID 2-The system searches for the account in the same organization of the user's and shows a list of its current role(s) and a list of available roles (in the user's same organization) to be assigned 3-User picks the role(s) he wants to assign and submits 4-System saves
Alternative flow	If entered ID is invalid or not found 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Role is saved for the account

Name	Delete role from account in user's organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters account's ID 2-The system searches for the account in the same organization of the user's and shows a list of its current role(s) and a list of the account role(s) 3-User picks the role(s) he wants to delete and submits 4-System deletes
Alternative flow	If entered ID is invalid or not found 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	Desired role is deleted from desired account

Name	Formulate Role
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to roles page
Basic flow	1-System shows a list of permissions (including admin privilege permissions) 2-User determines permissions he wants to from a role by, and enters the role name and submits 3-System saves the role to admin organization
Alternative flow	--
Post-conditions	Role is saved in the user's same organization

Name	Assign role to any account
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters account's ID 2-The system searches for the account in the whole system and shows a list of its current role(s) and a list of available roles (in the user's same organization (admin organization)) to be assigned 3-User picks the role(s) he wants to assign and submits 4-System saves
Alternative flow	--
Post-conditions	Desired role is saved to the desired account

Name	Delete role from any account
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters account's ID 2-The system searches for the account in the whole system and shows a list of its current role(s) and a list of available roles (in the user's same organization (admin organization)) to be deleted 3-User picks the role(s) he wants to delete and submits 4-System saves
Alternative flow	--
Post-conditions	Desired role deleted from desired account

Name	view role in organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to roles page
Basic flow	1-user clicks on the role desired 2-system views the role's permissions 3-user opts to delete 4-System deletes
Alternative flow	--
Post-conditions	Desired role deleted from desired account

Name	Update role in organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to roles page
Basic flow	1-user clicks on the role desired 2-system views the role's permissions 3-user edits permissions 4-System updates
Alternative flow	--
Post-conditions	Role's permissions are updated

Name	Create Organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to organizations page
Basic flow	User fills in the data of the new organization (ID, Name, address.. etc.) and submits System saves
Alternative flow	If entered data is invalid: 1-System prompts a message specifying the issue 2-Step 1 in basic flow
Post-conditions	New organization is saved

Name	Soft Delete organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to organizations page
Basic flow	1-User enters the organization's ID 2-System previews the organization 3-user opts to delete 4-system prompts a confirmation prompt 5-user confirms 6-system deletes
Alternative flow	If entered data is invalid (bad password or invalid or not found ID..etc.): 1-System prompts a message specifying the issue Step 2 in basic flow If user doesn't confirm: 1-step 2 in basic flow
Post-conditions	Desired organization is deleted

Name	Create account for specific organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to account page
Basic flow	System shows a list of all the organizations in the system User enters new account's national ID, password, and starting role (from roles created in the user's same organization (admin organization)) and picks the organization to create the account in and submits 3-System saves the account and shows "account created" message
Alternative flow	If entered data is invalid (bad password or invalid ID..etc.): 1-System prompts a message specifying the issue 2-Step 2 in basic flow
Post-conditions	Account created in the desired organization

Name	Update account in user's org
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to account page
Basic flow	1-User enters the account's national ID 2-System previews the account with fields updatable 3-user enters new values 4-system saves changes
Alternative flow	If entered data is invalid (bad password or invalid or not found in org ID..etc.): 1-System prompts a message specifying the issue 2-Step 2 in basic flow
Post-conditions	Desired account is updated

Name	Update any account
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to account page
Basic flow	1-User enters the account's national ID 2-System previews the account with fields updatable 3-user enters new values 4-system saves changes
Alternative flow	If entered data is invalid (bad password or invalid or not found ID..etc.): 1-System prompts a message specifying the issue 2-Step 2 in basic flow
Post-conditions	Desired account is updated

Name	Remove account from user's organization
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters the account's national ID 2-System previews the account 3-user opts to remove 4-system prompts a confirmation prompt 5-user confirms 6-system deletes account from user's organization
Alternative flow	If entered data is invalid (invalid or not found ID..etc.): 1-System prompts a message specifying the issue Step 2 in basic flow If user doesn't confirm: 1-step 2 in basic flow
Post-conditions	Desired account is removed from user's org

Name	Remove account from specific org
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to accounts page
Basic flow	1-User enters the account's national ID 2-System previews the account 3-user specifies the organization and opts to remove 4-system prompts a confirmation prompt 5-user confirms 6-system deletes account from user's organization
Alternative flow	If entered data is invalid (invalid or not found ID..etc.): 1-System prompts a message specifying the issue Step 2 in basic flow If user doesn't confirm: 1-step 2 in basic flow
Post-conditions	Desired account is removed from desired org

Name	Create Database Collection (Table)
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to collections page
Basic flow	<p>1-System shows all collections created with (+) sign</p> <p>2-User clicks on (+) sign</p> <p>3-System opens creation page</p> <p>4-User determine whether the collection is patient related (medical record category to be) and whether it is patient specific or to contain records, also names the collection and submits</p> <p>5-System saves and allows to form a template of fields to add records through, in case determined to contain records (doesn't allow if determined as patient specific). And allows to form a template to add information to a patient's medical record tab through, in case determined as patient related</p> <p>6-User forms the templates by adding fields to through (+) signs in a way allowing to place the fields wherever he wants in the template, adds the number of fields he wants, specifies the name and type of each (text, number, date, select field (can import the select list content from records saved in a collection)) and submits</p> <p>5-System saves</p>
Alternative flow	If user opts to save the collection only without forming one or both of required template: 1-system saves the collection without one or both of its templates (can be added anytime)
Post-conditions	New collection is created

Name	Delete Database Collection (Table)
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to collections page
Basic flow	1-System shows all collections created by name 2-User clicks on the desired collection 3-System opens the collection page with delete option 4-User opts to delete 5-System prompts a confirmation prompts 6-User confirms 7-System deletes
Alternative flow	If user doesn't confirm: 1-step 4 in basic flow
Post-conditions	Desired collection is deleted

Name	Add template
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to collections page
Basic flow	1-System shows all collections created by name 2-User clicks on the collection he wishes to add missing template to 3-System opens the collection page 4-User determines the template to add 5-System shows a form to add fields to through (+) signs in a way allowing to place the field wherever the user wants in the form 6-User adds the number of fields he wants, specifies the name and type of each (text, number, select field) and submits 7-System saves
Alternative flow	--
Post-conditions	New template saved for desired collection

Name	Add collection document (record)
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to collections page
Basic flow	1-System shows all collections created by name 2-User clicks on the collection he wishes to add document (record) to 3-System opens the collection page showing the main form template -in case the collection is not patient specific and is to contain records- to fill (or update) 4-User fills the form template and submits 5-System saves
Alternative flow	If another template is shown: 1-User chooses the main template he wishes to add a document by from a list 2-Step 4 in basic flow
Post-conditions	New document saved in the desired collection

Name	Update a collection's template
prerequisites	Internet connection User's account has necessary permissions
Pre-conditions	User is logged in User has navigated to collections page
Basic flow	1-System shows all collections created by name 2-User clicks on the collection he wishes to add document (record) to 3-System opens the collection page showing a form template to update with new fields (or fill) in addition to a list of other created templates 4-User chooses the template he wishes to reform (if there is) or simply leaves the one already shown, enables builder mode, add fields to the form template wherever he wants through the (+) sign, names them, and submits 5-System saves
Alternative flow	--
Post-conditions	Desired template is updated

2.4.5 Non-functional requirements

Security:

The system must be secure from unauthorized access.

Account creation: Only system admin and organization managers can create an account for their workers

Passwords: each organization requires a password for accessing it the password is unique for each account

Rate limiting: a user has a certain limit of attempts to log into the organization and his account

Performance:

The system should respond to user interactions within an acceptable timeframe, such as loading patient records, retrieving data, or generating reports. For example, the system should load patient records in less than two seconds.

Scalability:

The system should be able to handle an increasing number of concurrent users and growing data volumes without a significant degradation in performance. It should be designed to scale both vertically (adding more resources to a single server) and horizontally (distributing the workload across multiple servers).

Maintenance:

The system should be designed and implemented in a way that allows for easy maintenance, upgrades, and enhancements. It should have well-documented code, modular architecture, and support for version control to facilitate future changes and enhancements.

Portability:

The system should run on various operating systems and devices like desktop and mobiles.

Capacity:

the system uses distributed databases a one database for each organization in addition to amazon s3 storage service for storing files.

Usability:

The system should be user-friendly, intuitive, and easy to navigate for healthcare professionals. It should have clear and concise user interfaces, provide appropriate feedback, and offer customization options to accommodate different user preferences and workflows.

Reliability:

The system should be highly reliable, ensuring that patient data is consistently available and accessible. It should have mechanisms in place to prevent data loss, handle system failures gracefully, and provide backup and recovery procedures.

Interoperability:

The system should support interoperability standards to exchange data with other healthcare systems, such as laboratory systems, imaging systems, or health information exchanges. It should be able to import and export data in industry-standard formats to facilitate seamless data exchange.

Compliance:

The system should comply with relevant legal, regulatory, and industry standards, such as HIPAA (Health Insurance Portability and Accountability Act) in the United States or GDPR (General Data Protection Regulation) in the European Union. It should ensure data integrity, auditability, and adherence to specific requirements related to healthcare data.

2.5 Advantages of the new system

Intuitive and user-friendly interface, improving system usability and navigation compared to complex interfaces in other systems.

Optimized performance, ensuring smooth operation and minimizing instances of slowdowns or frozen screens.

Streamlined features, eliminating redundancies and focusing on essential functionalities, enhancing system efficiency and ease of use.

Dynamic formatting of patient forms and components, offering flexibility and customization for precise data collection, with guarantee of data consistency.

Comprehensive patient data management, providing a holistic view of patient health, including medical history, medications, allergies, immunization status, and laboratory results.

Efficient data sharing and interoperability, enabling secure exchange of patient data with authorized healthcare providers, pharmacies, laboratories, and specialists.

Enhanced sessions scheduling and payment registration, simplifying administrative tasks and improving the patient experience.

Robust database collections, ensuring efficient storage, retrieval, and management of patient data, with a focus on data integrity and security.

Advanced data visualization and reporting capabilities, offering insights into healthcare facility operations and patient outcomes, supporting informed decision-making.

Decision support features, leveraging statistical calculations and patient data to provide recommendations and improve diagnostic accuracy.

Alerts for medication interactions, enhancing patient safety and reducing the risk of adverse events.

2.6 Risk and risk management

Risk	Risk avoidance	Risk management	Probability of occurrence	Level of impact
Security breaches leading to unauthorized access or data breaches	Implement robust security measures, such as encryption and user authentication	Regularly monitor system for potential vulnerabilities, conduct security audits, and promptly address any identified issues	Medium	High
System downtime or technical failures	Implement redundancy measures and backup systems	Regularly perform system maintenance, testing, and backups. Develop contingency plans to minimize downtime	Medium	High
Data loss or corruption	Implement regular data backups and secure storage solutions	Use data recovery mechanisms and redundant storage systems to minimize data loss. Regularly test data restoration processes	Low	High
Inadequate user training and adoption	Provide comprehensive user training programs and resources	Offer ongoing support and user assistance. Collect feedback to identify areas of improvement	Medium	Medium
Integration challenges with existing systems	Conduct thorough compatibility testing before integration	Collaborate with IT specialists to ensure smooth integration. Develop contingency plans for potential issues	Low	Medium

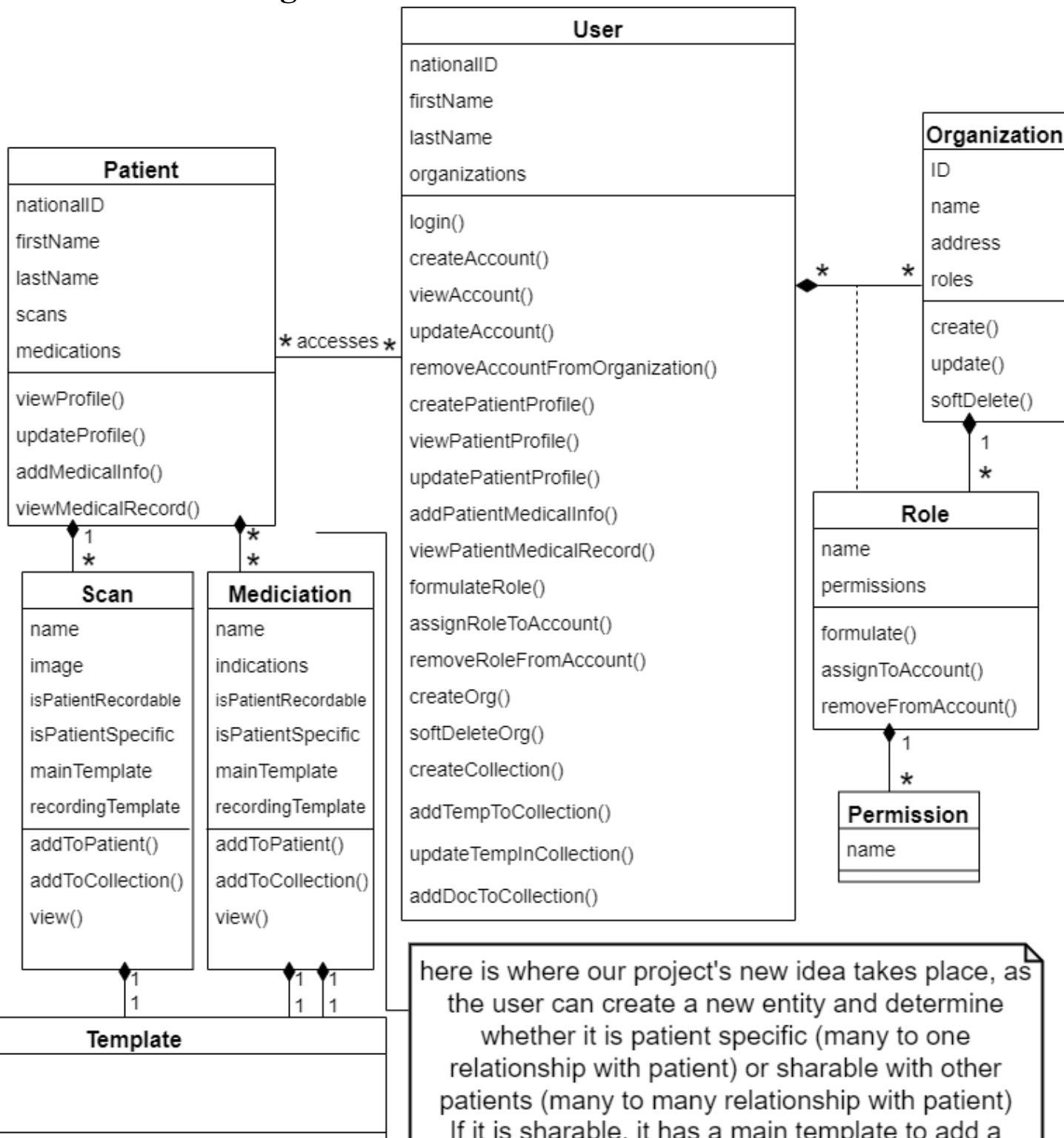
Inaccurate or incomplete data entry	Implement data validation measures and user-friendly interfaces	Develop and enforce policies and procedures to ensure compliance. Regularly audit privacy practices	Medium	Medium
Regulatory compliance and privacy concerns	Stay updated with relevant regulations and industry best practices	Develop and enforce policies and procedures to ensure compliance. Regularly audit privacy practices	Medium	Medium
Vendor dependency and system stability	Assess vendor reliability and stability before implementation	Maintain open communication with the vendor. Establish contingency plans in case of vendor-related issues	Low	Medium
Insufficient scalability and system capacity	Conduct capacity planning and scalability assessments	Regularly evaluate system performance and scalability needs. Upgrade infrastructure as necessary	Low	Low
User error and accidental data modifications	Implement user permissions and access controls	Regularly backup data and establish version control. Provide audit logs to track changes	High	Low
Inadequate backup and disaster recovery plans	Develop comprehensive backup and recovery strategies	Regularly test backup and recovery processes. Keep backups in off-site locations	Low	Low



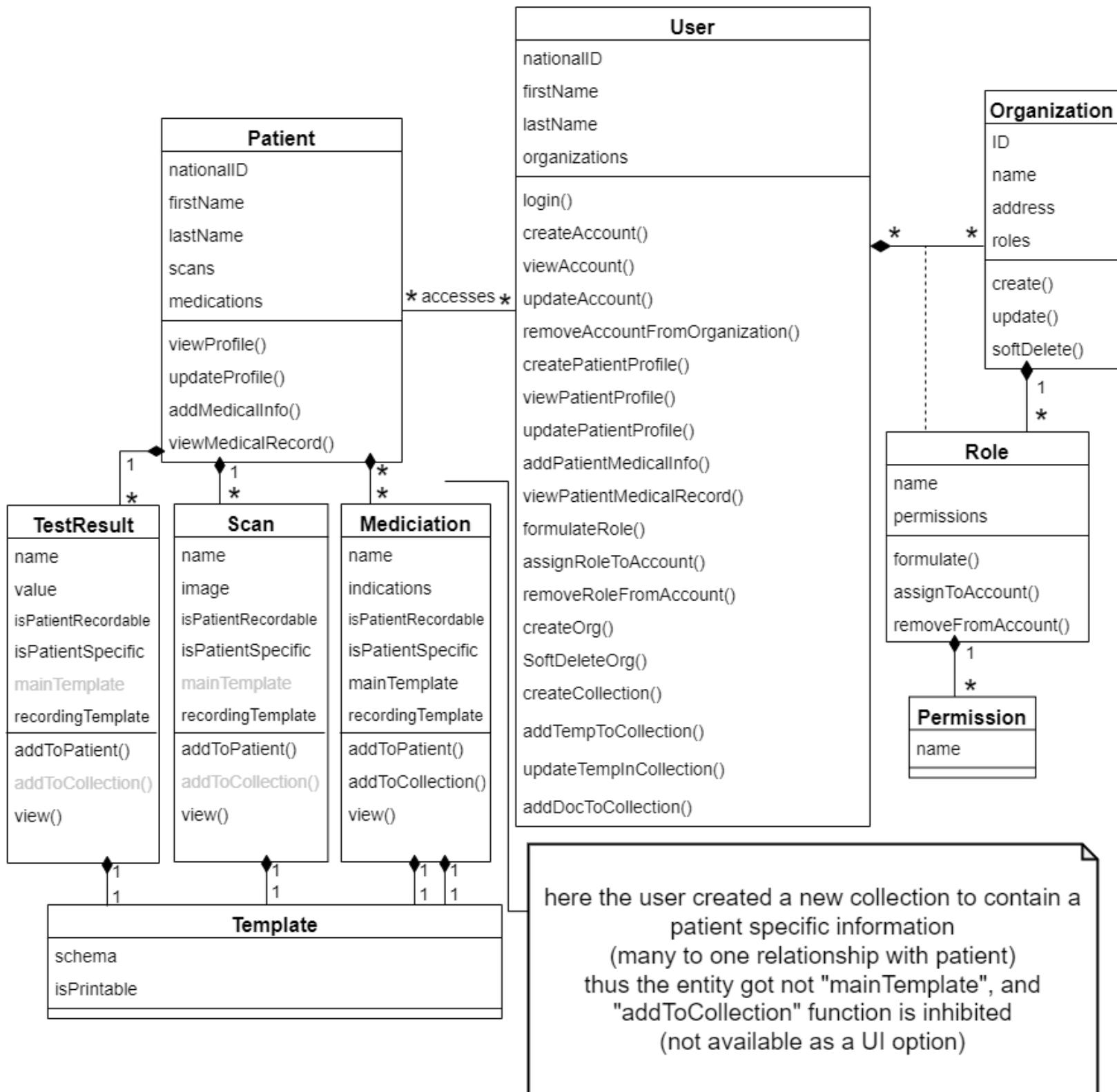
Chapter 3

Software design

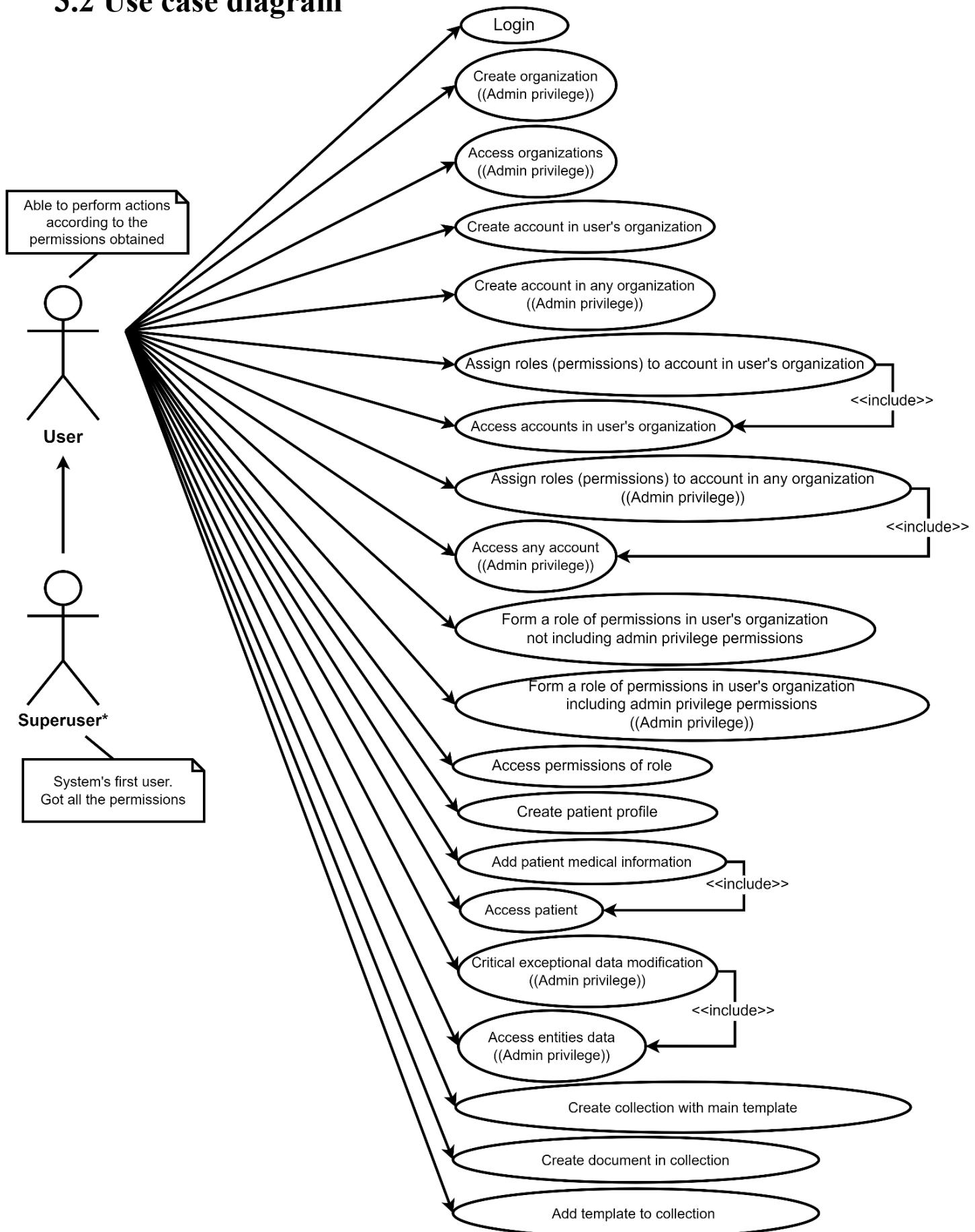
3.1 Class diagram



here is where our project's new idea takes place, as the user can create a new entity and determine whether it is patient specific (many to one relationship with patient) or sharable with other patients (many to many relationship with patient). If it is sharable, it has a main template to add a document to through, for example, "Panadol" instance to "Medication" entity

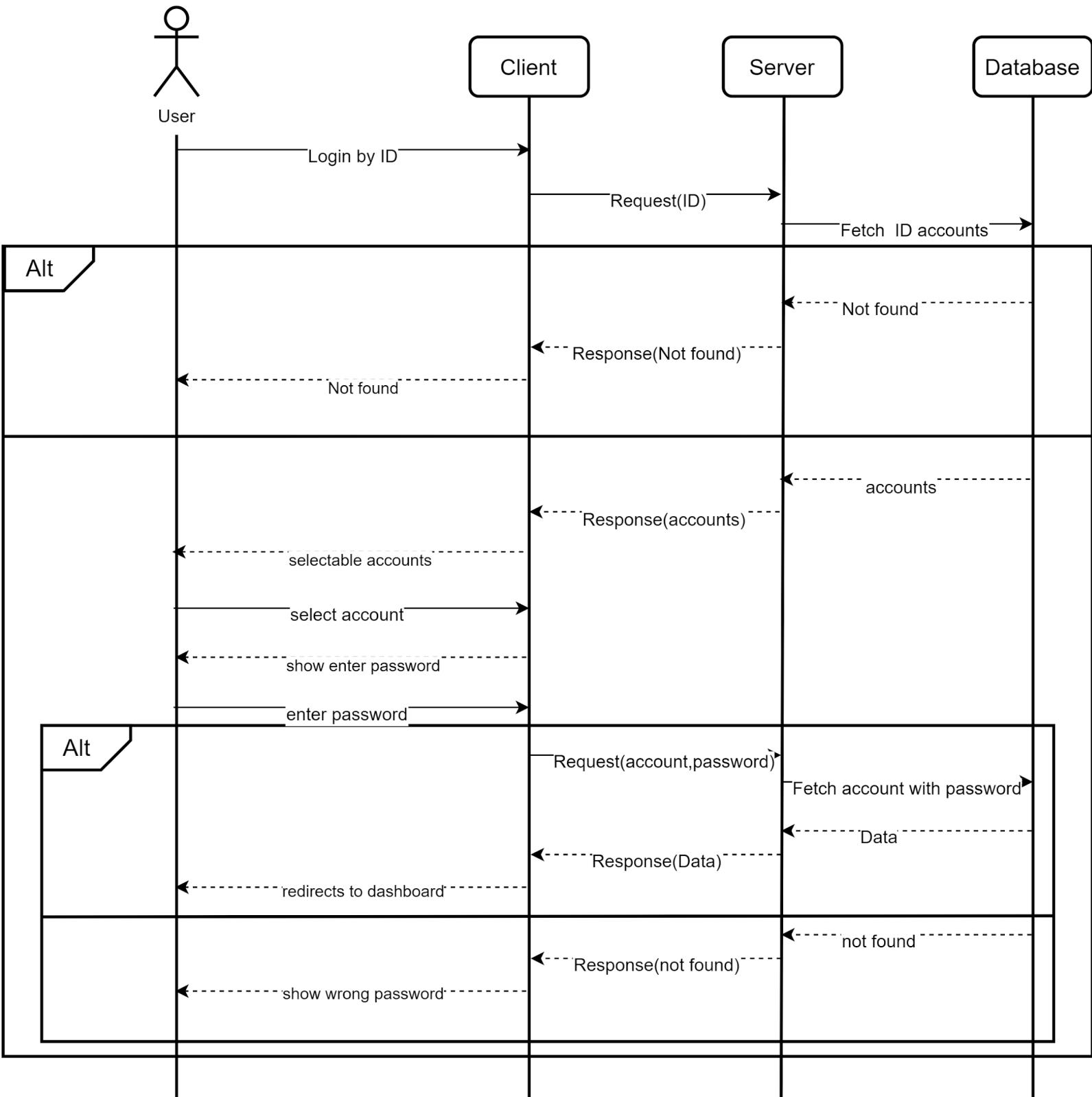


3.2 Use case diagram

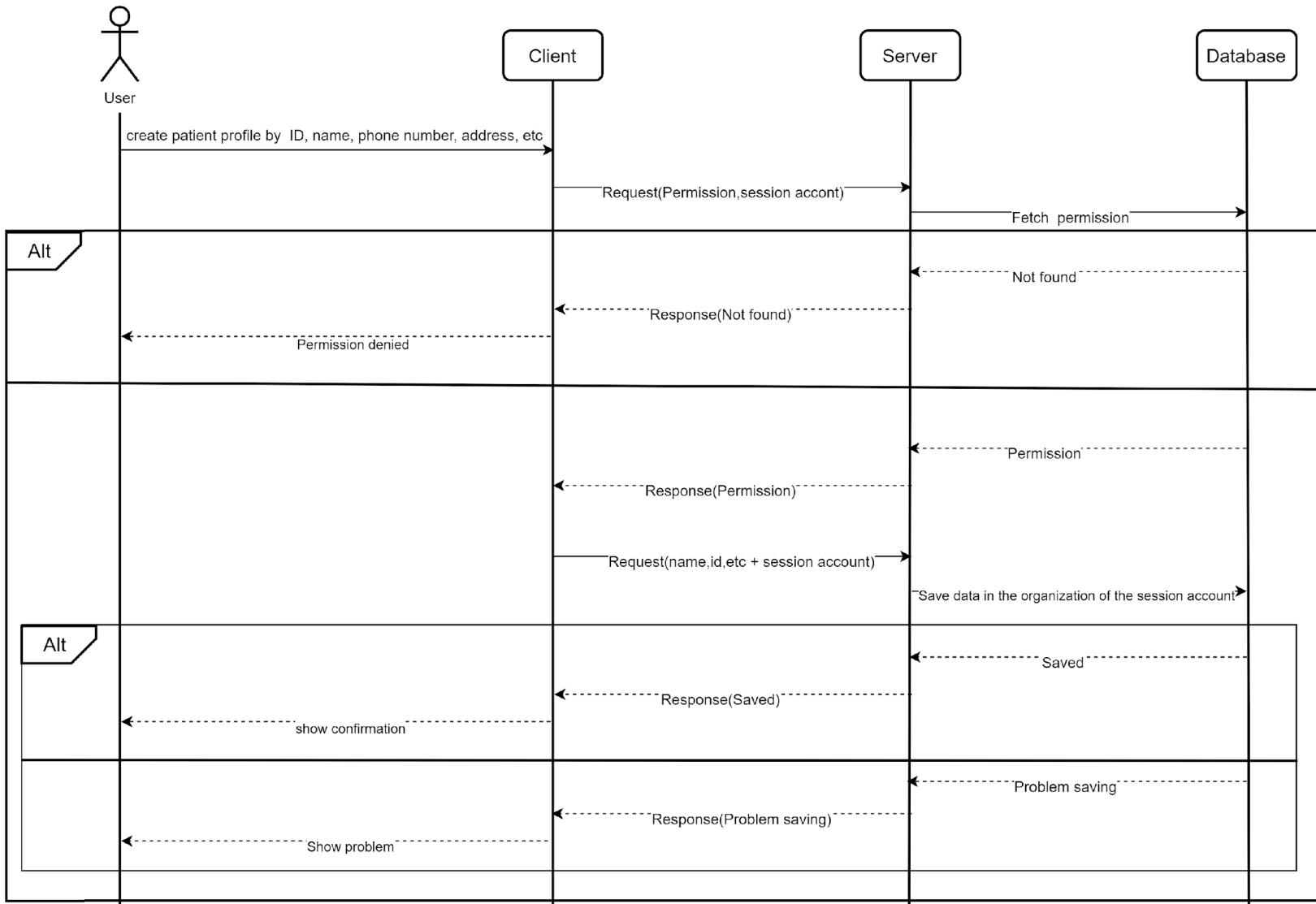


3.3 Sequence diagrams

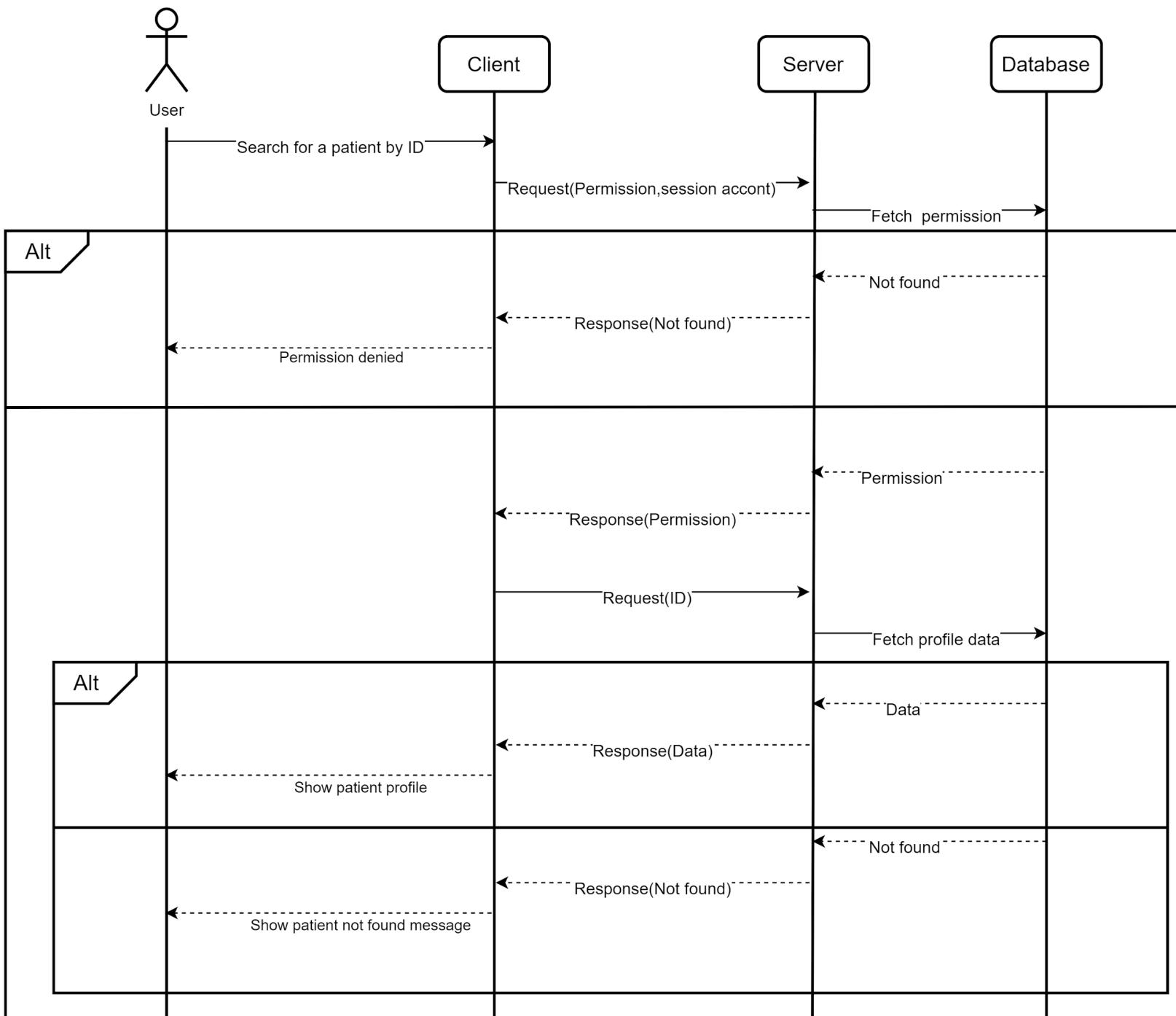
Login



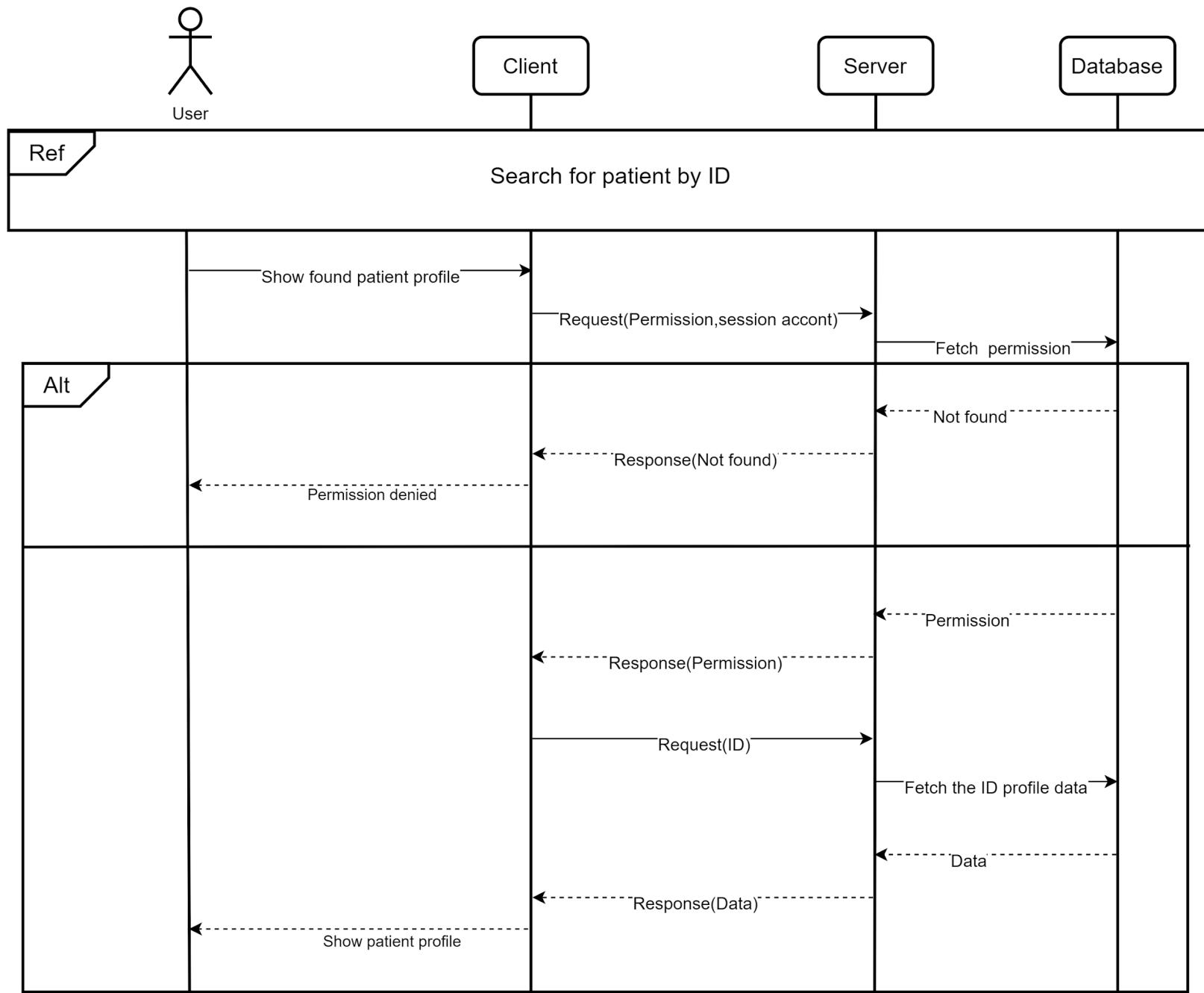
Create patient profile



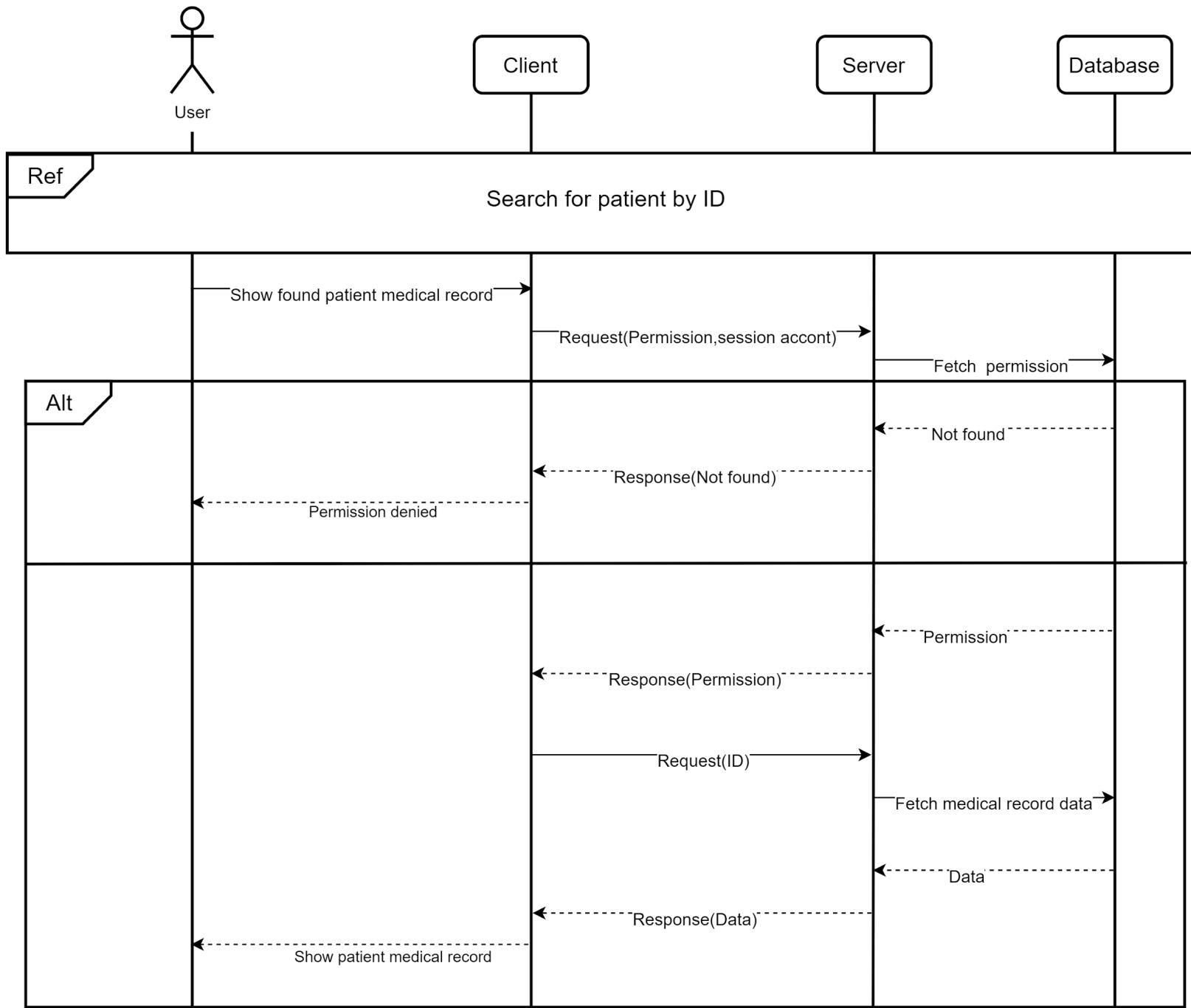
Search for patient



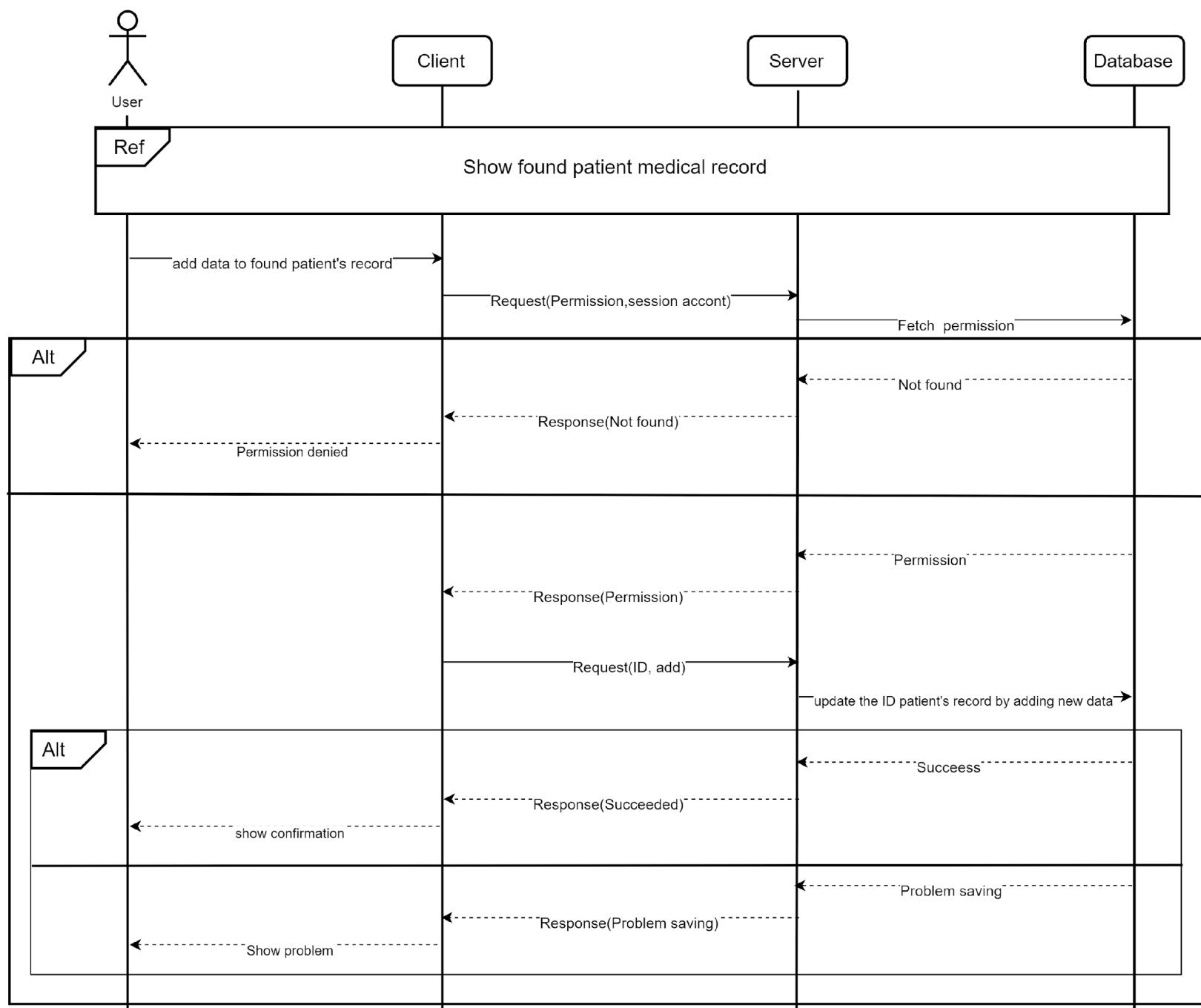
View patient profile



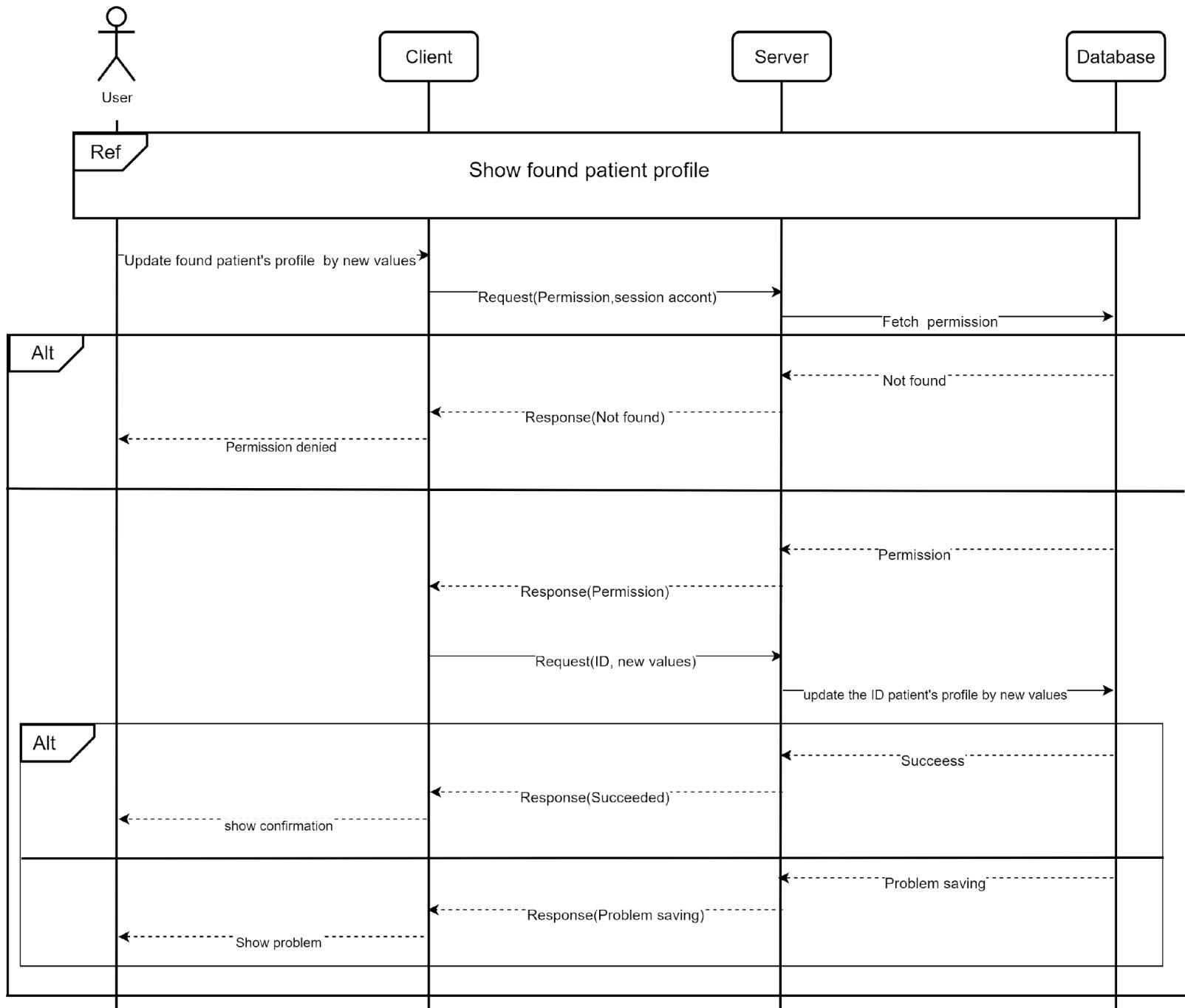
View patient medical record



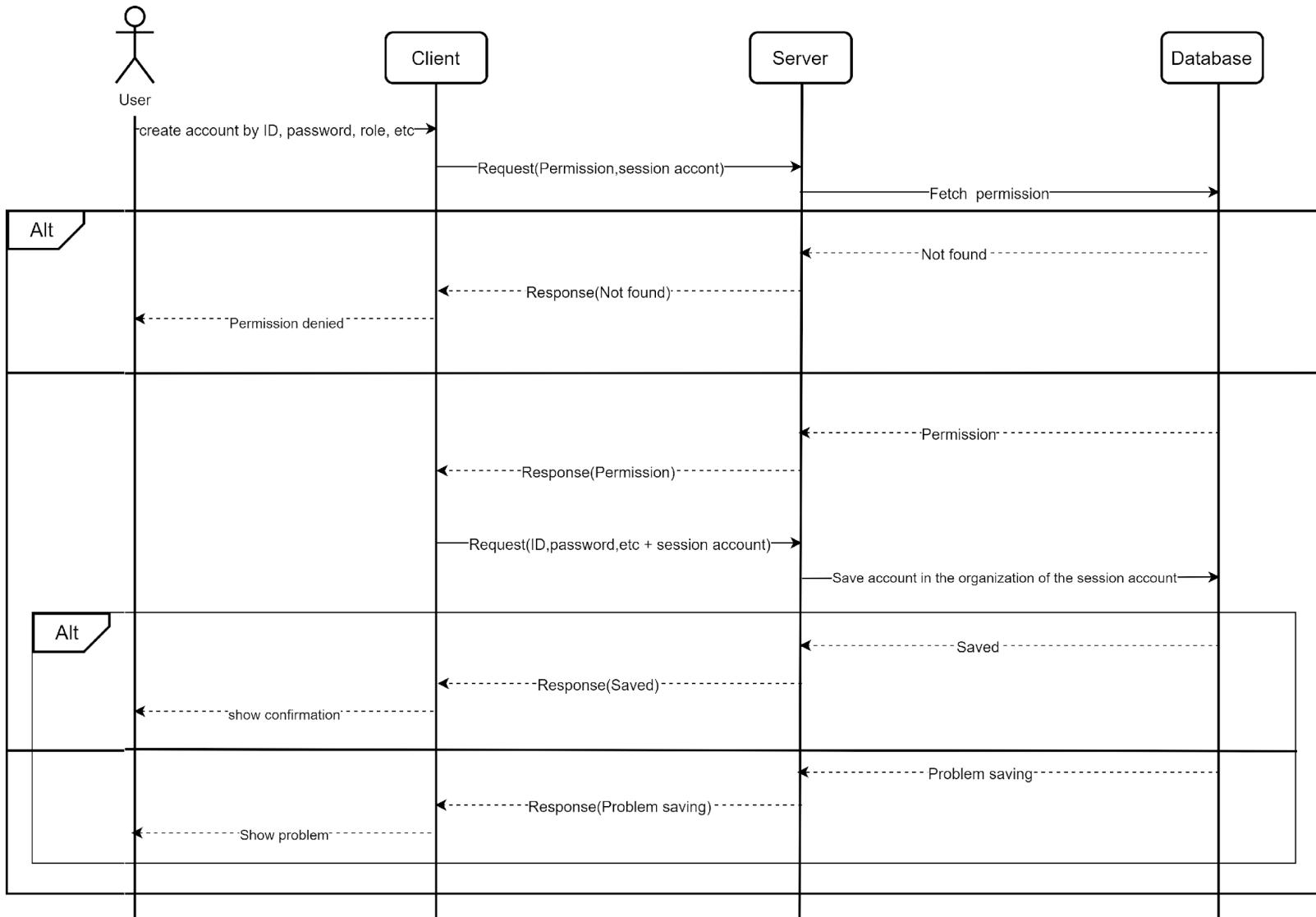
Add to medical record



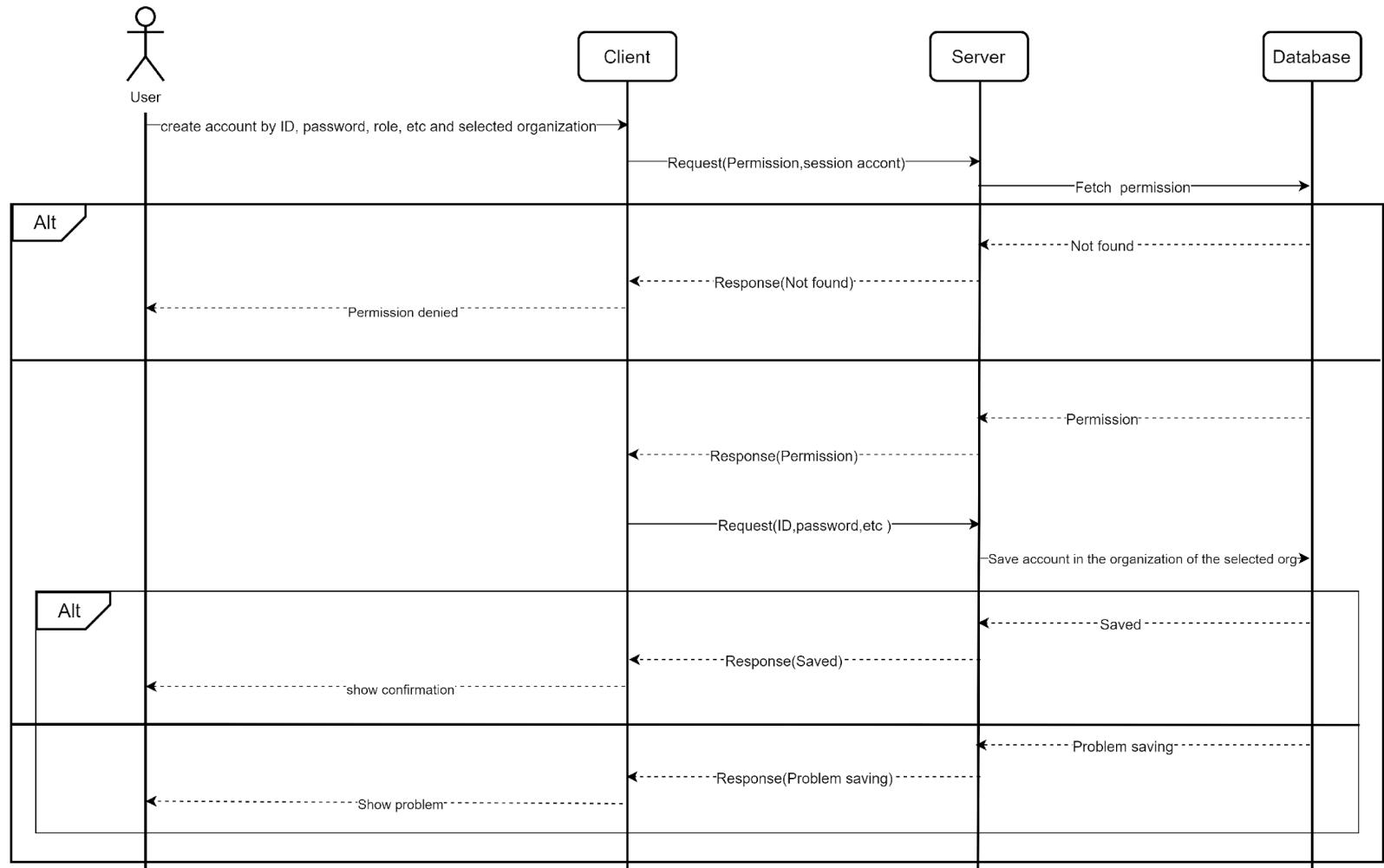
Edit patient profile



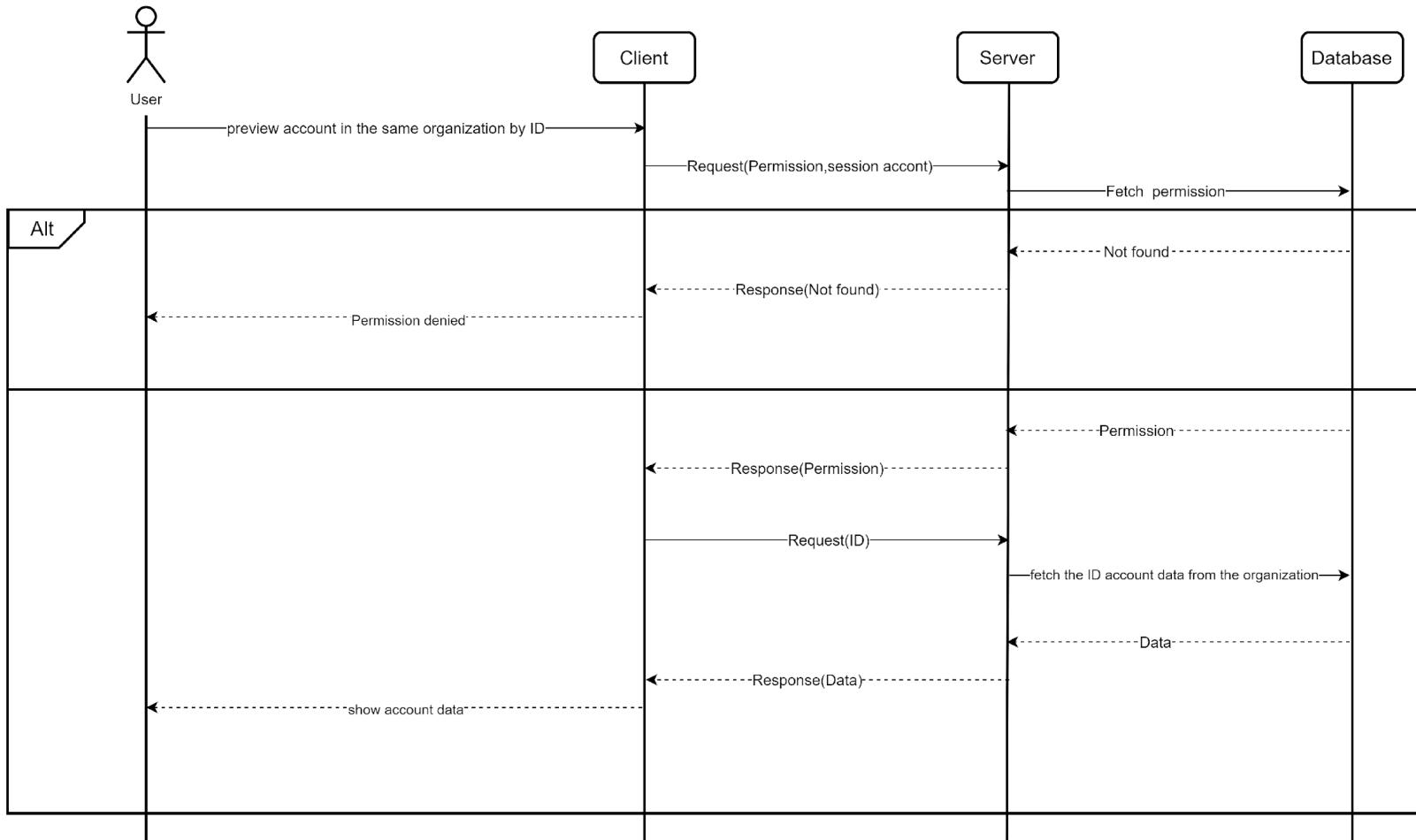
Create account in user's organization



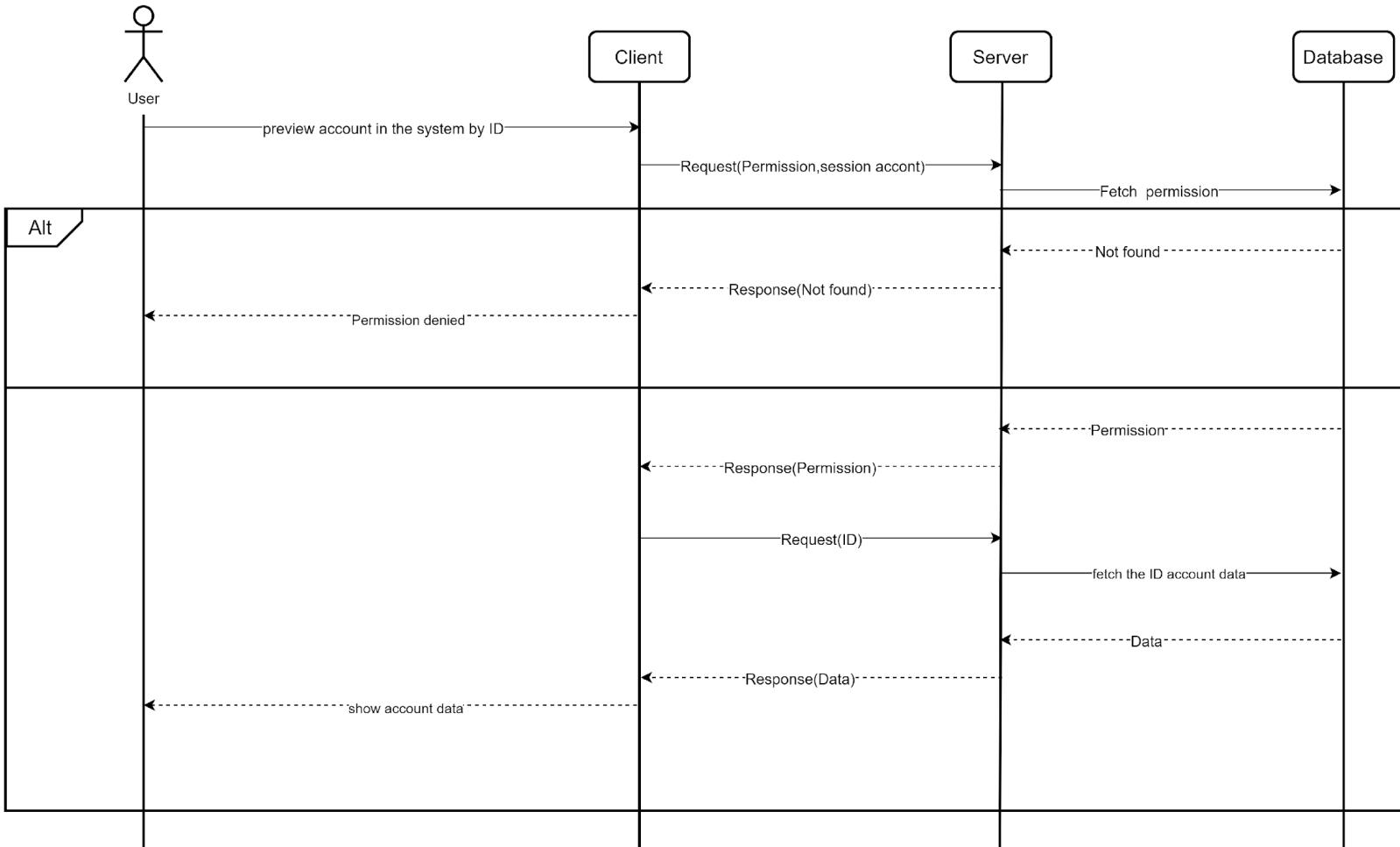
Create account in a specific organization



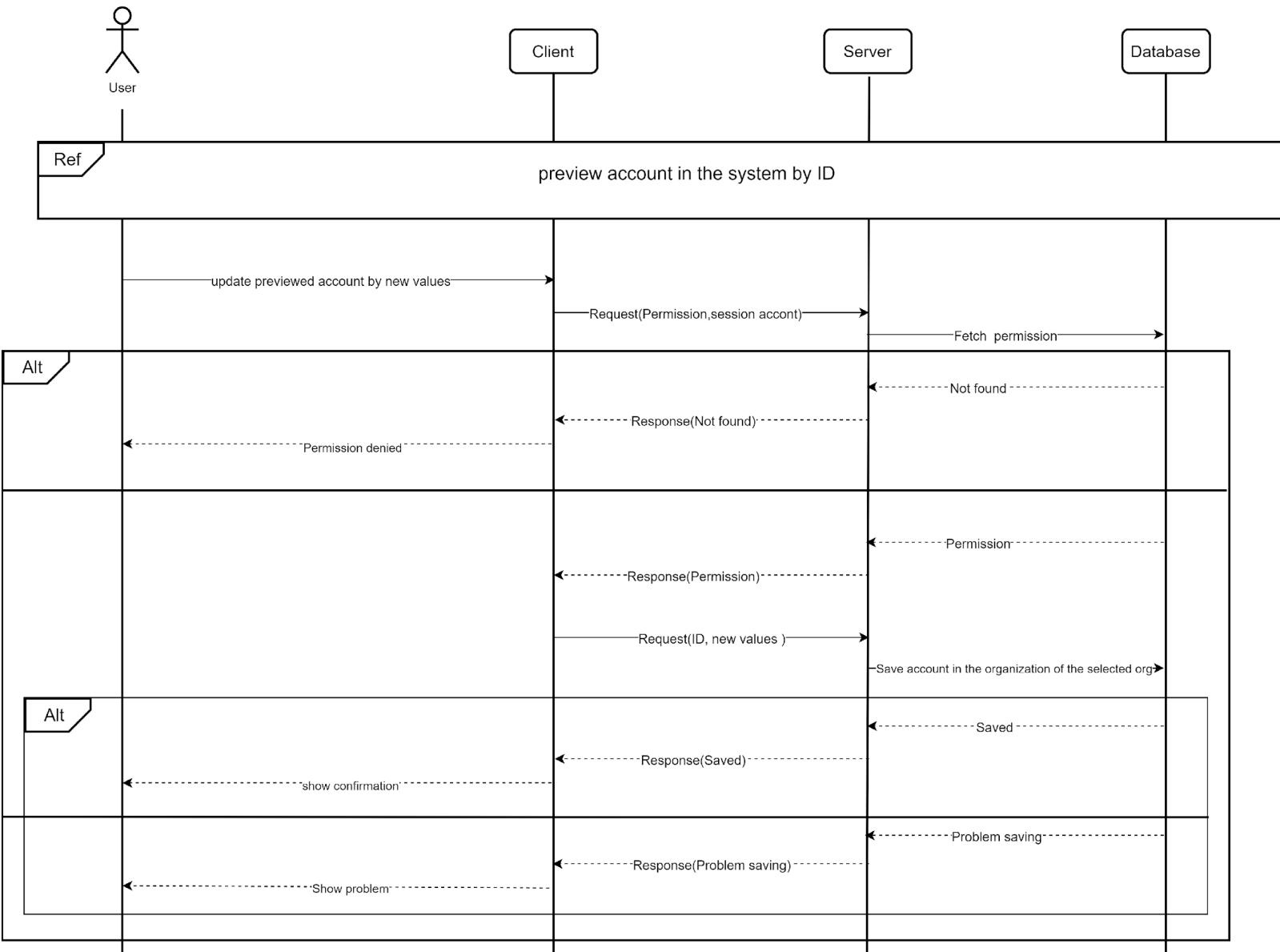
View account in the user's organization



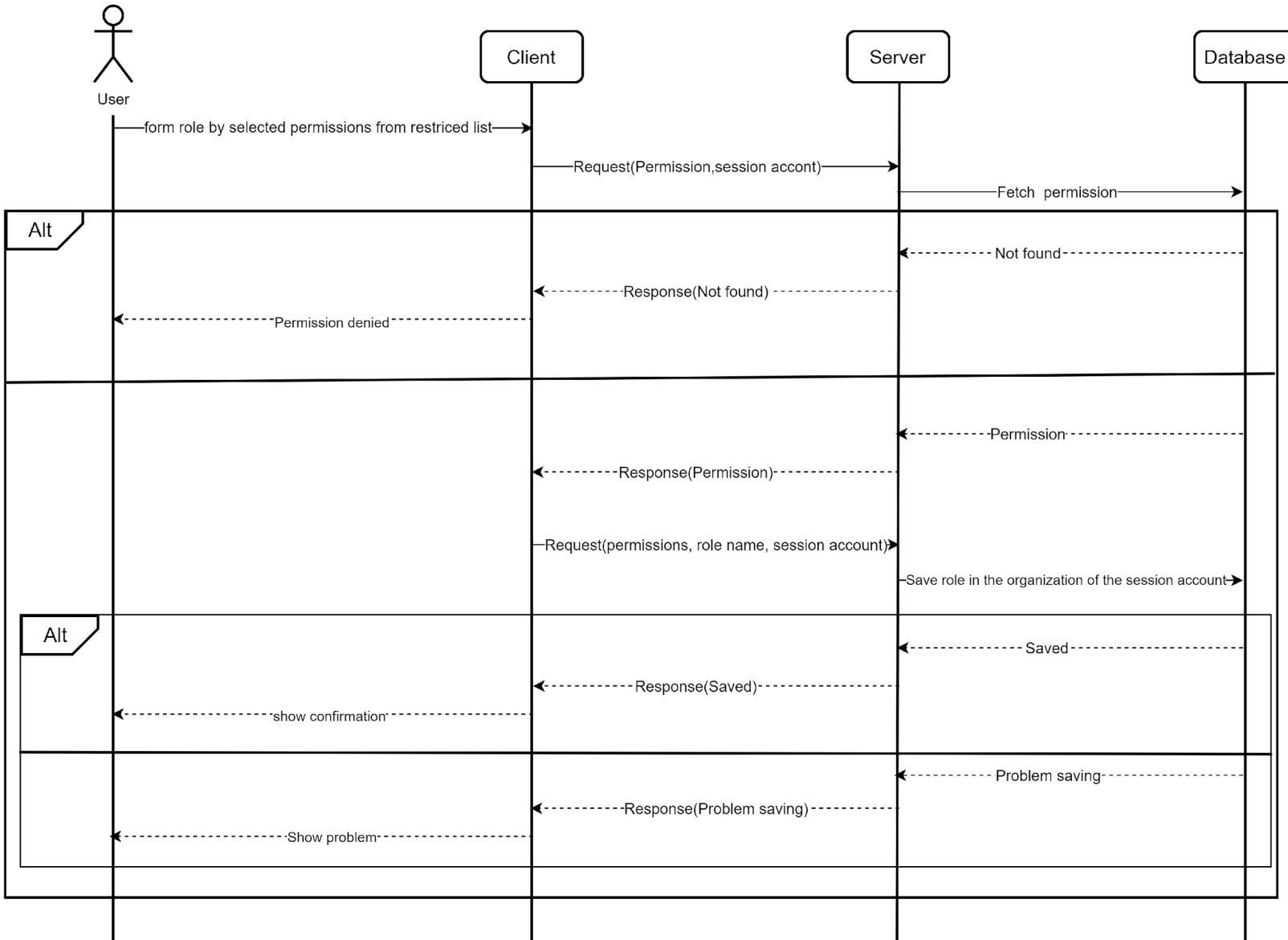
View account in the system



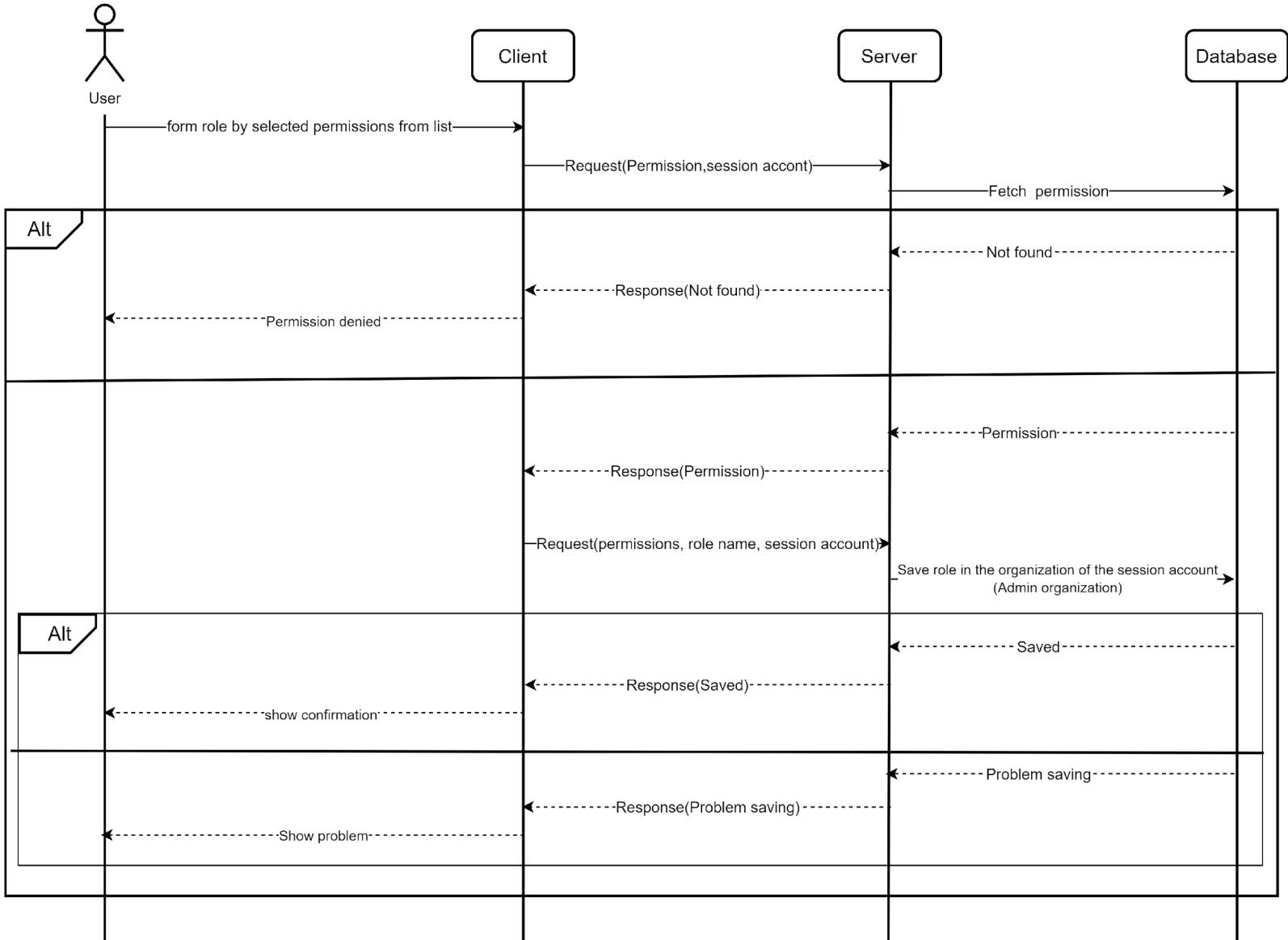
Update account



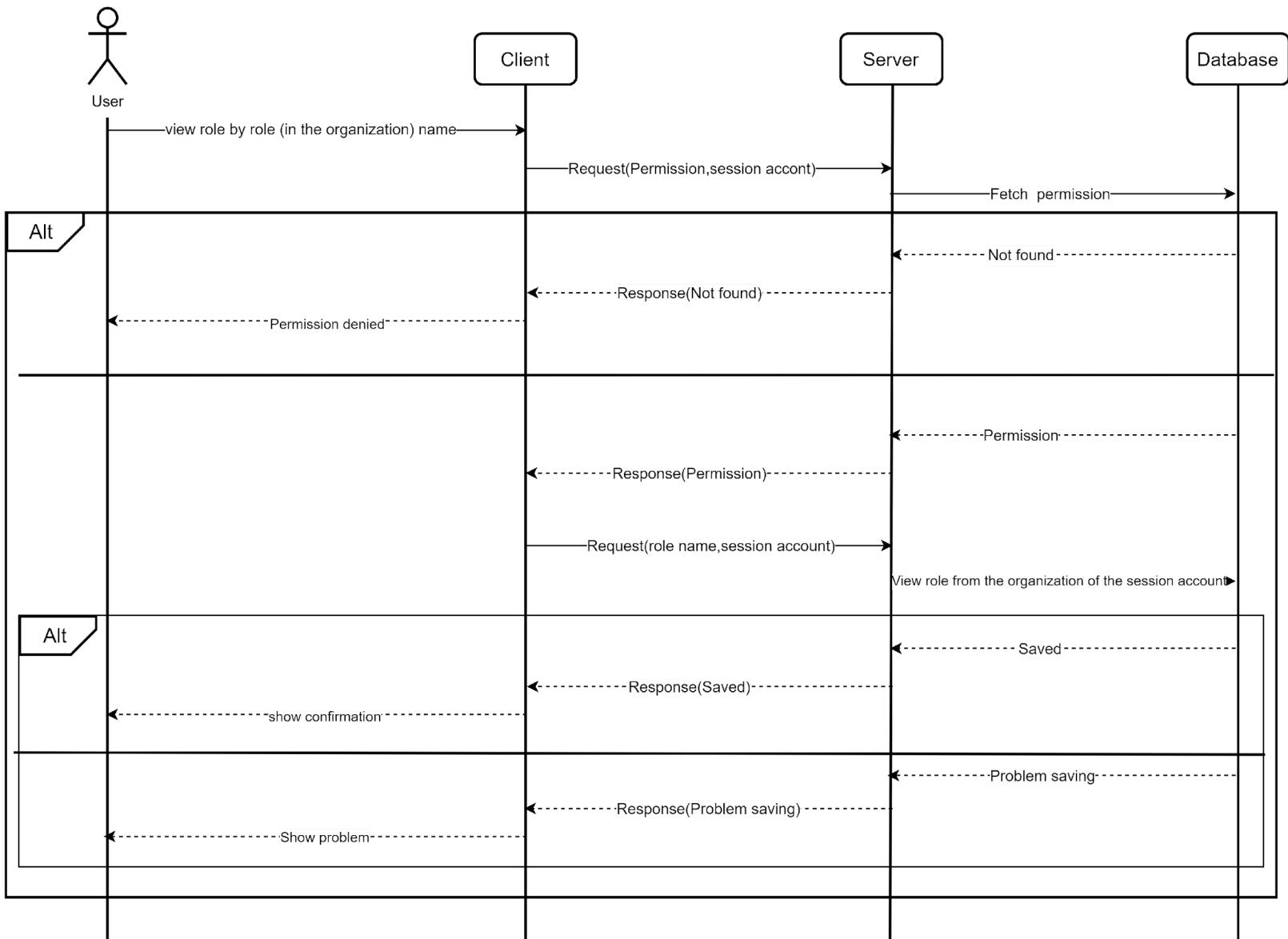
Formulate role in user's organization



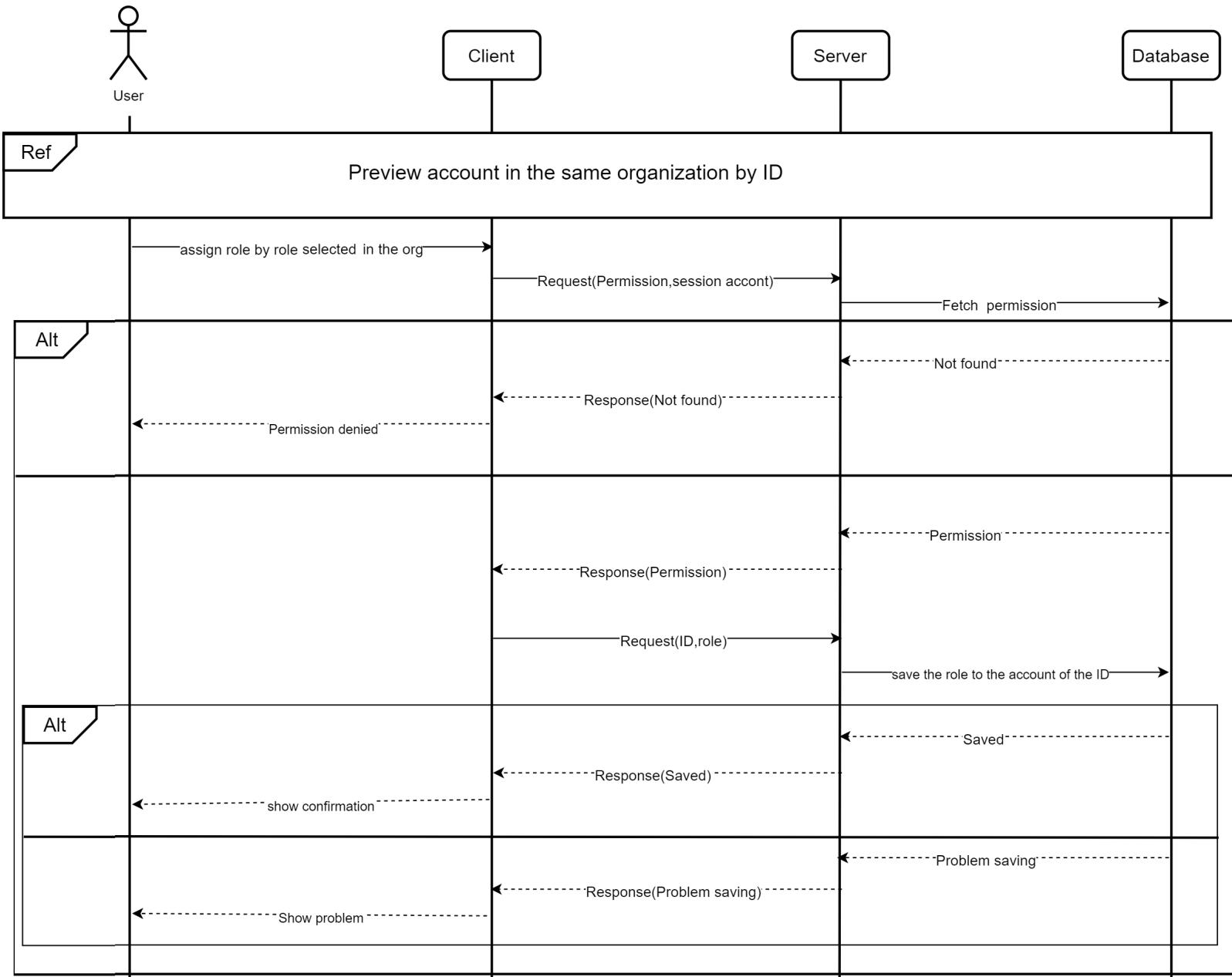
Formulate role in the system (system's org.)



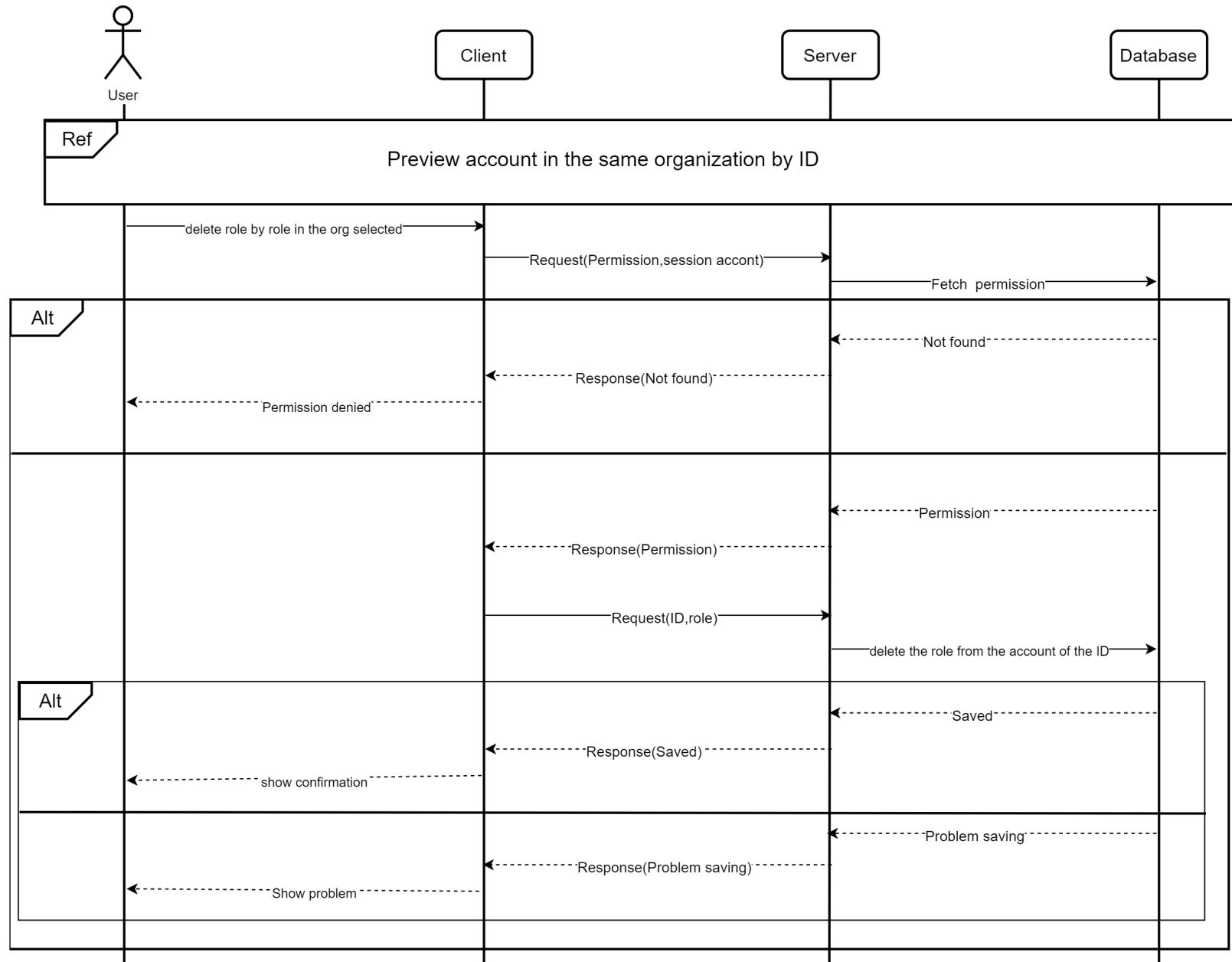
View roles



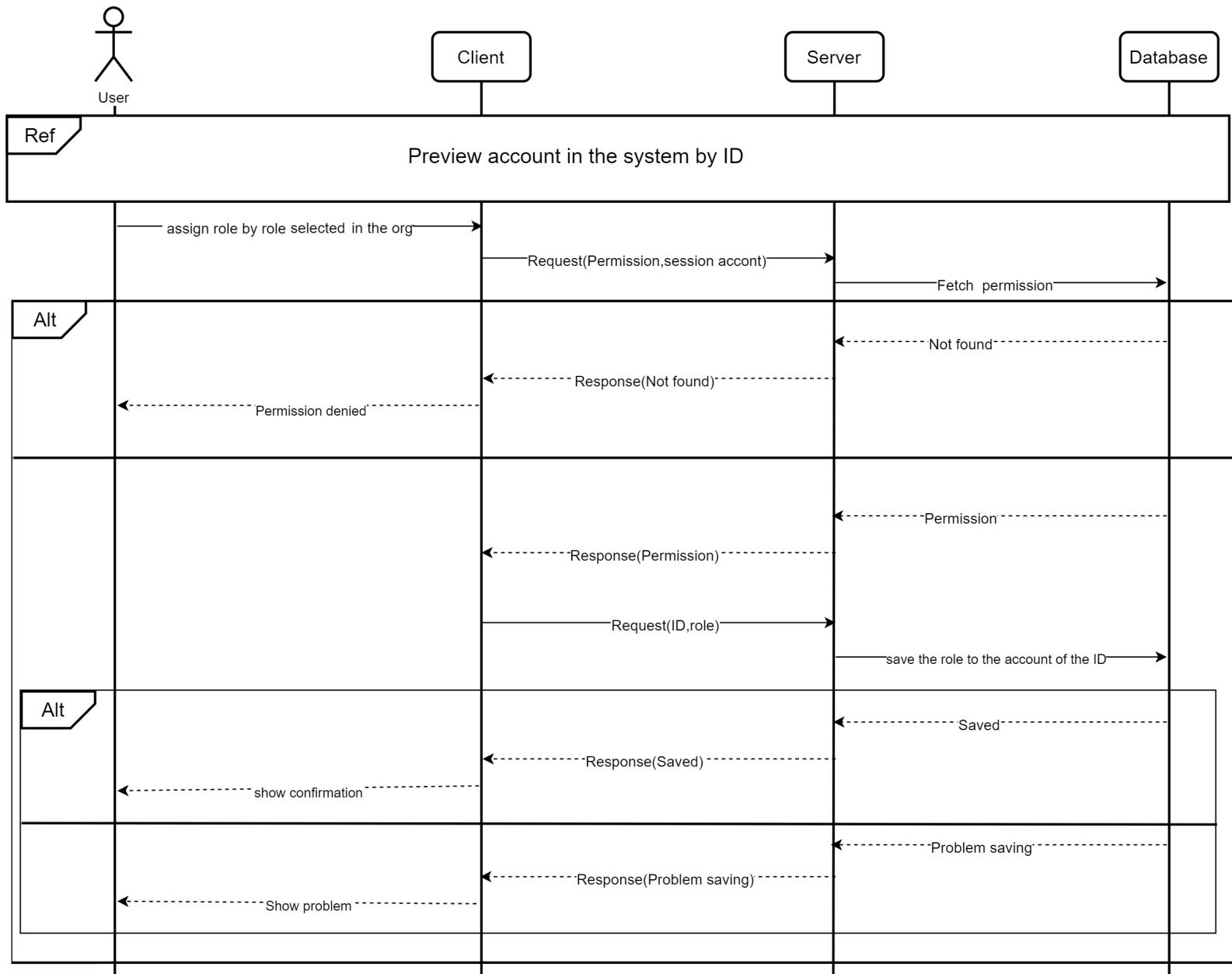
Assign role to account in user's organization



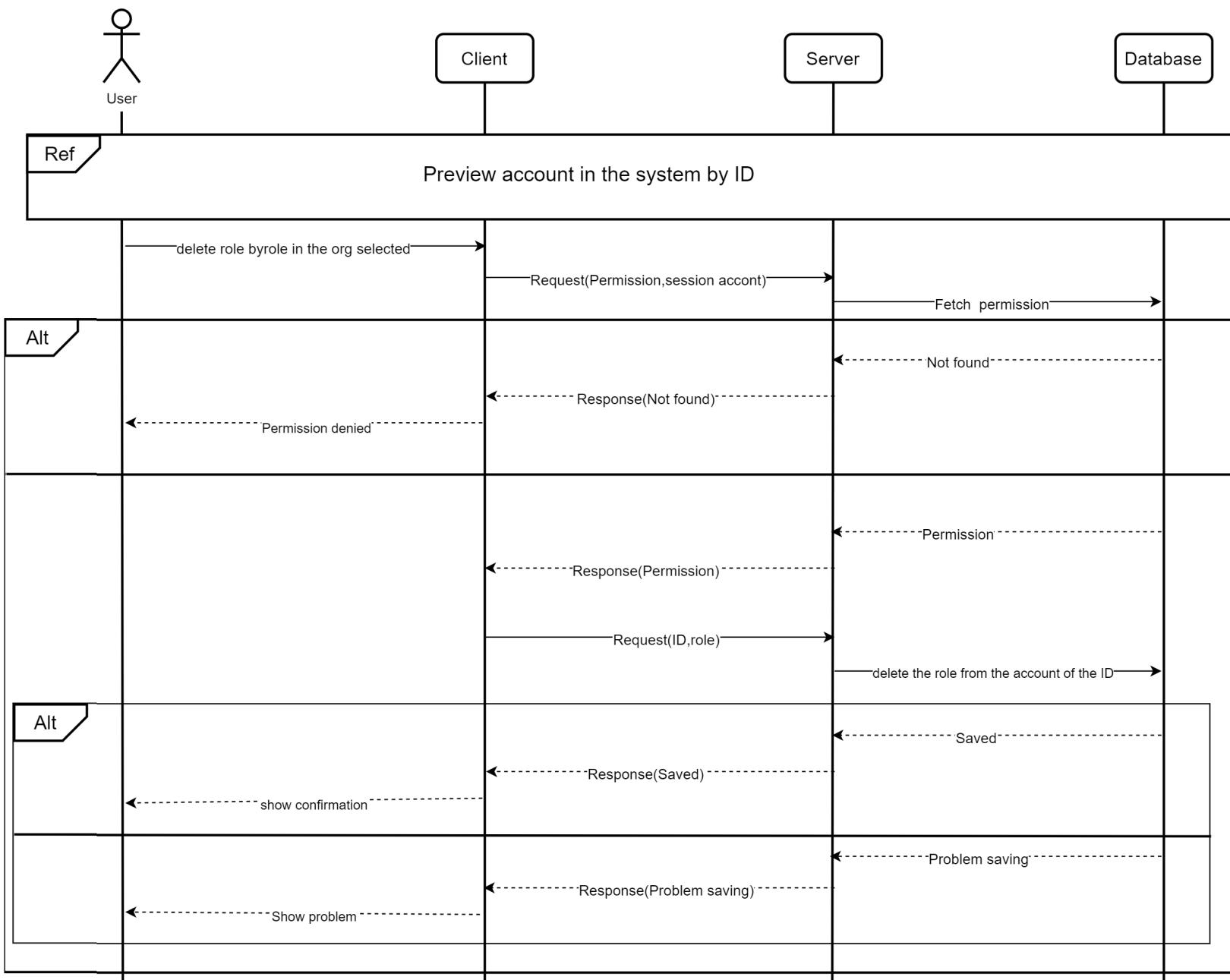
Remove role from account in user's organization



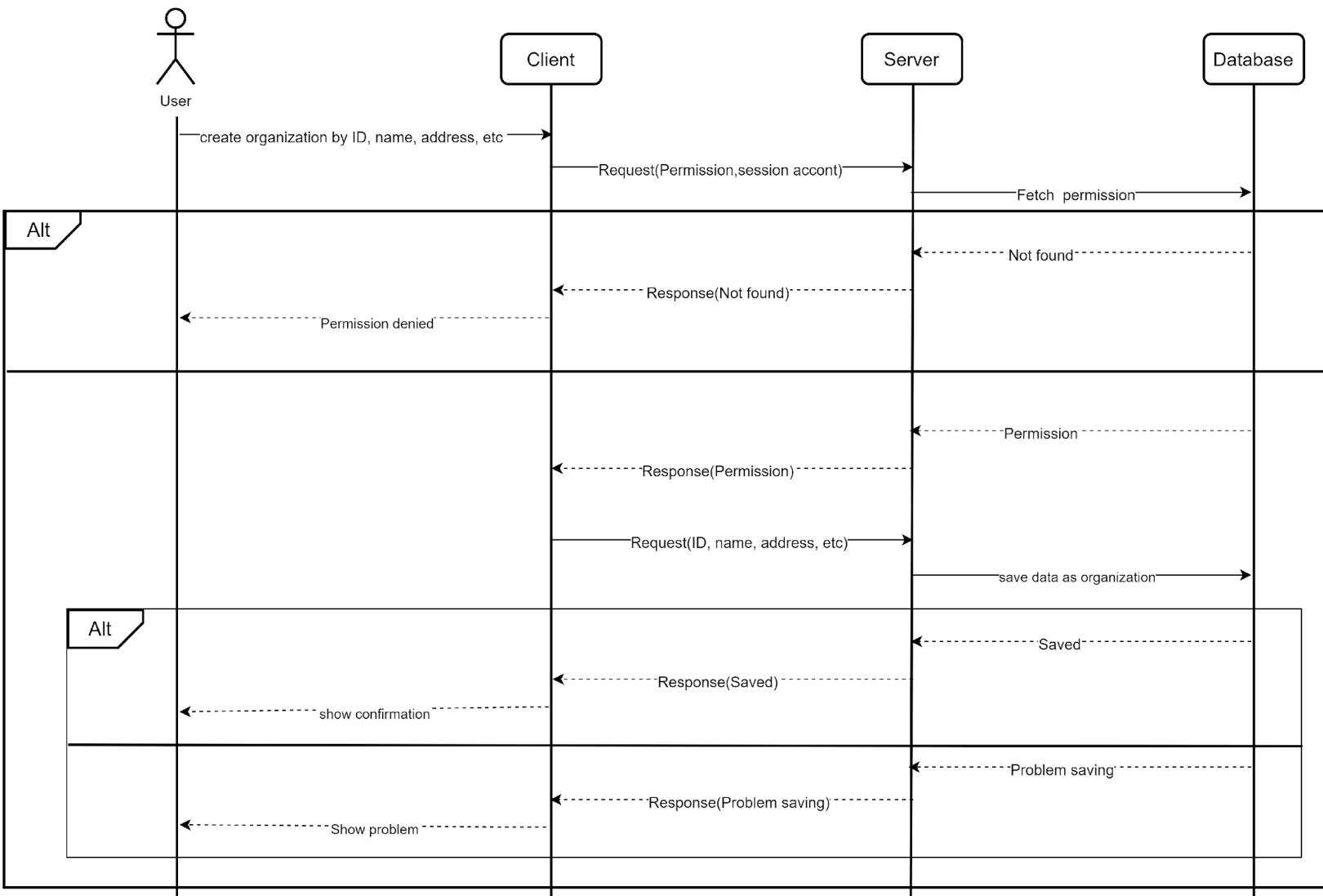
Assign role to account in the system



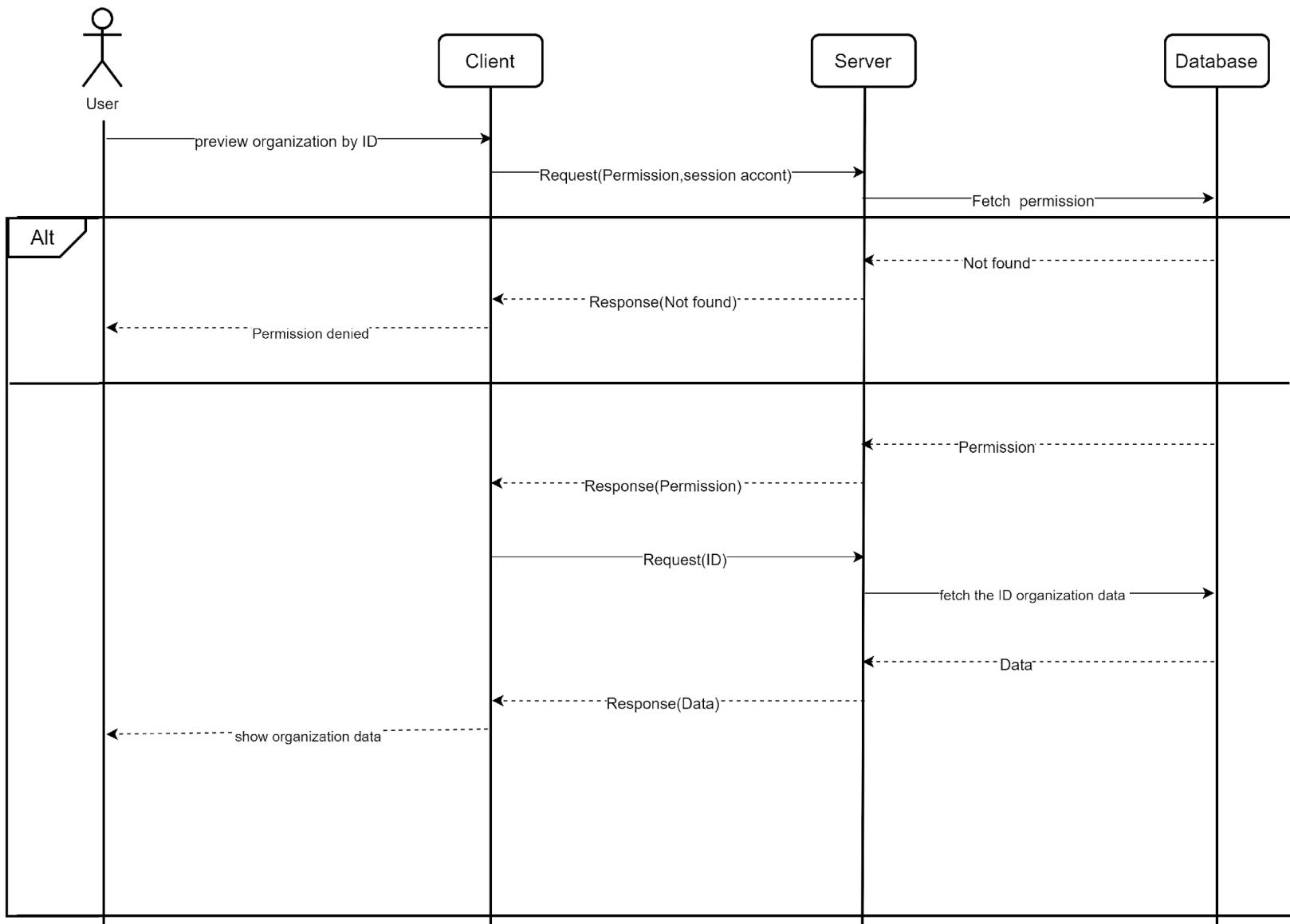
Remove role from account in the system



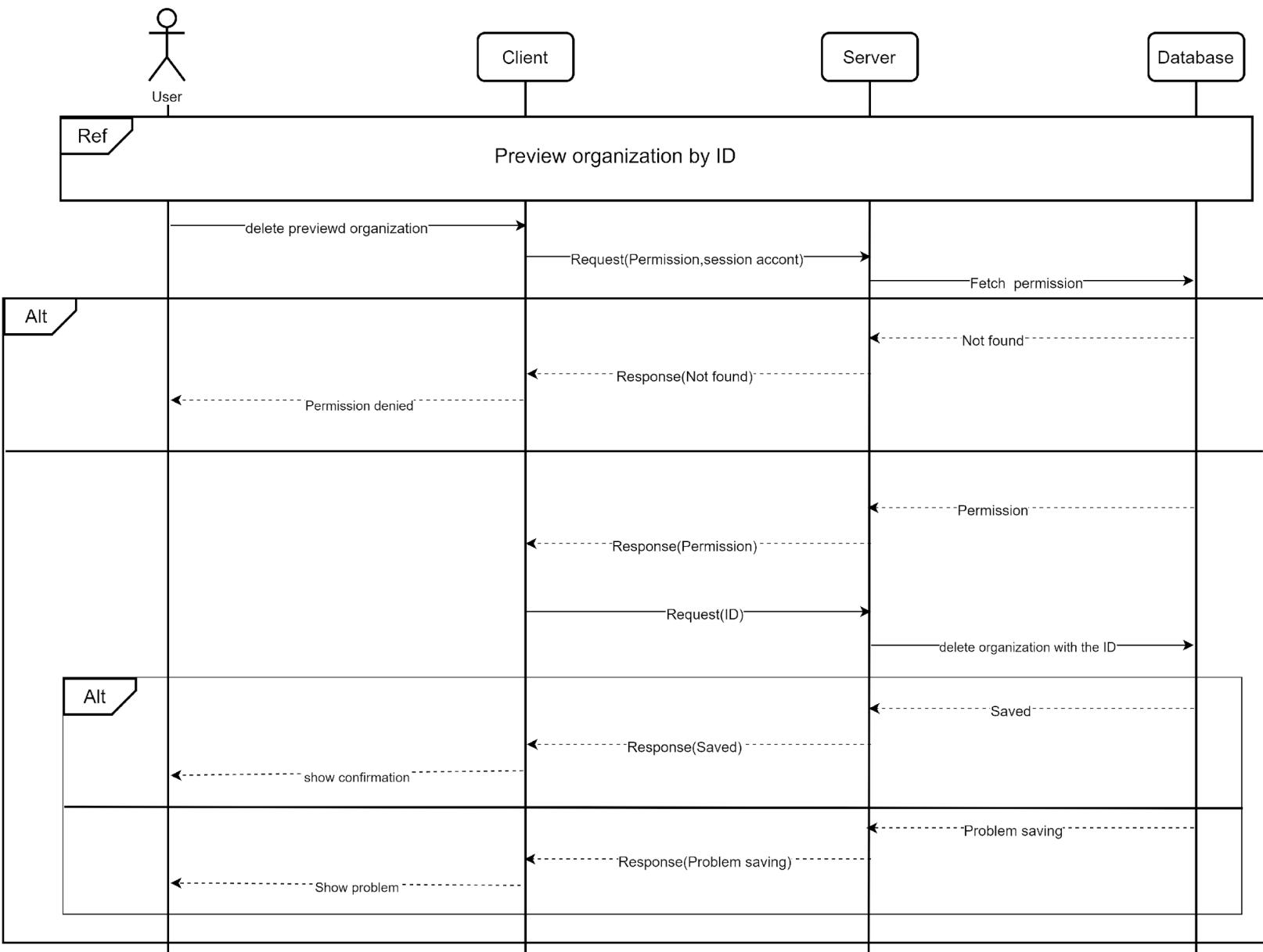
Create organization



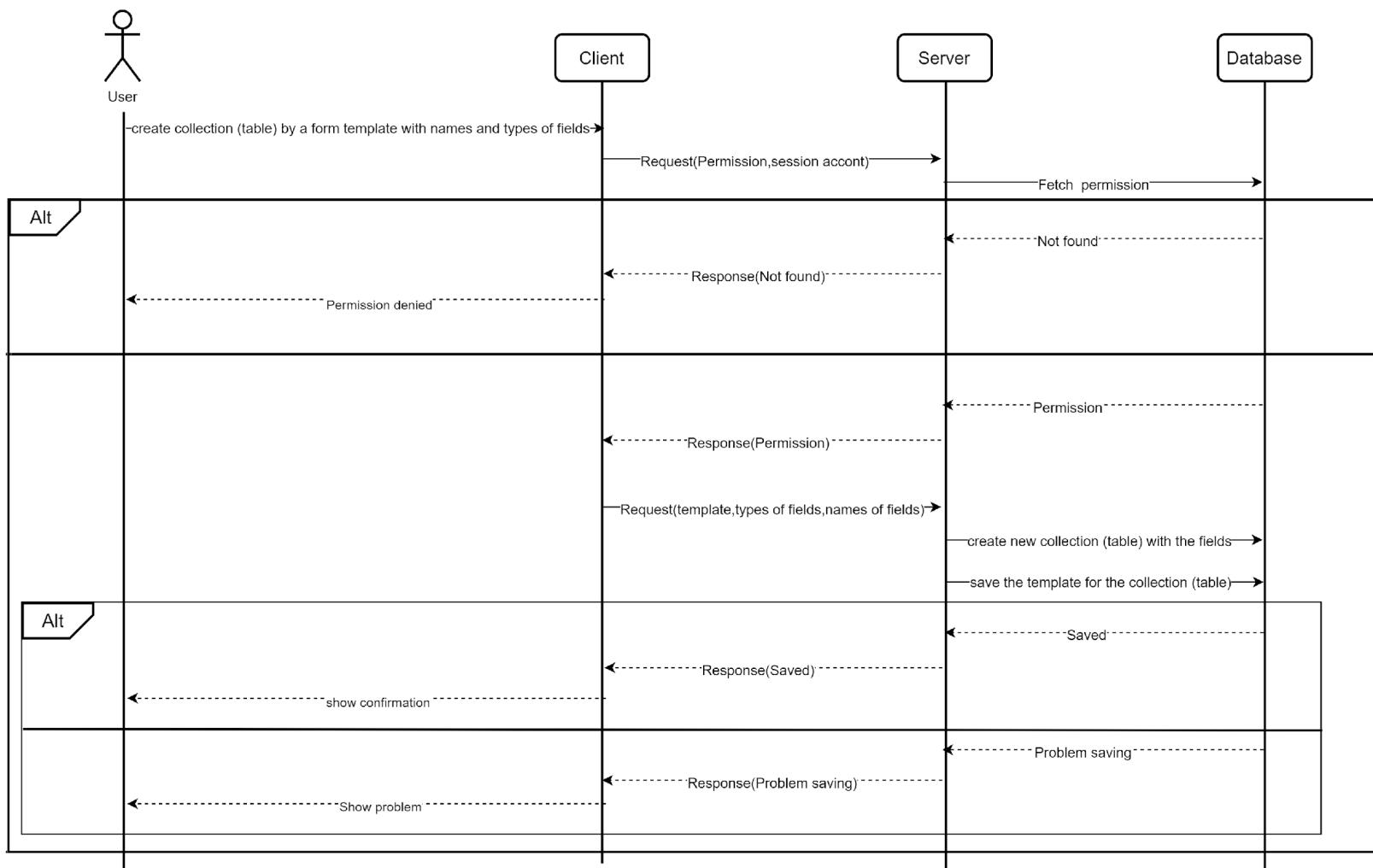
View organization



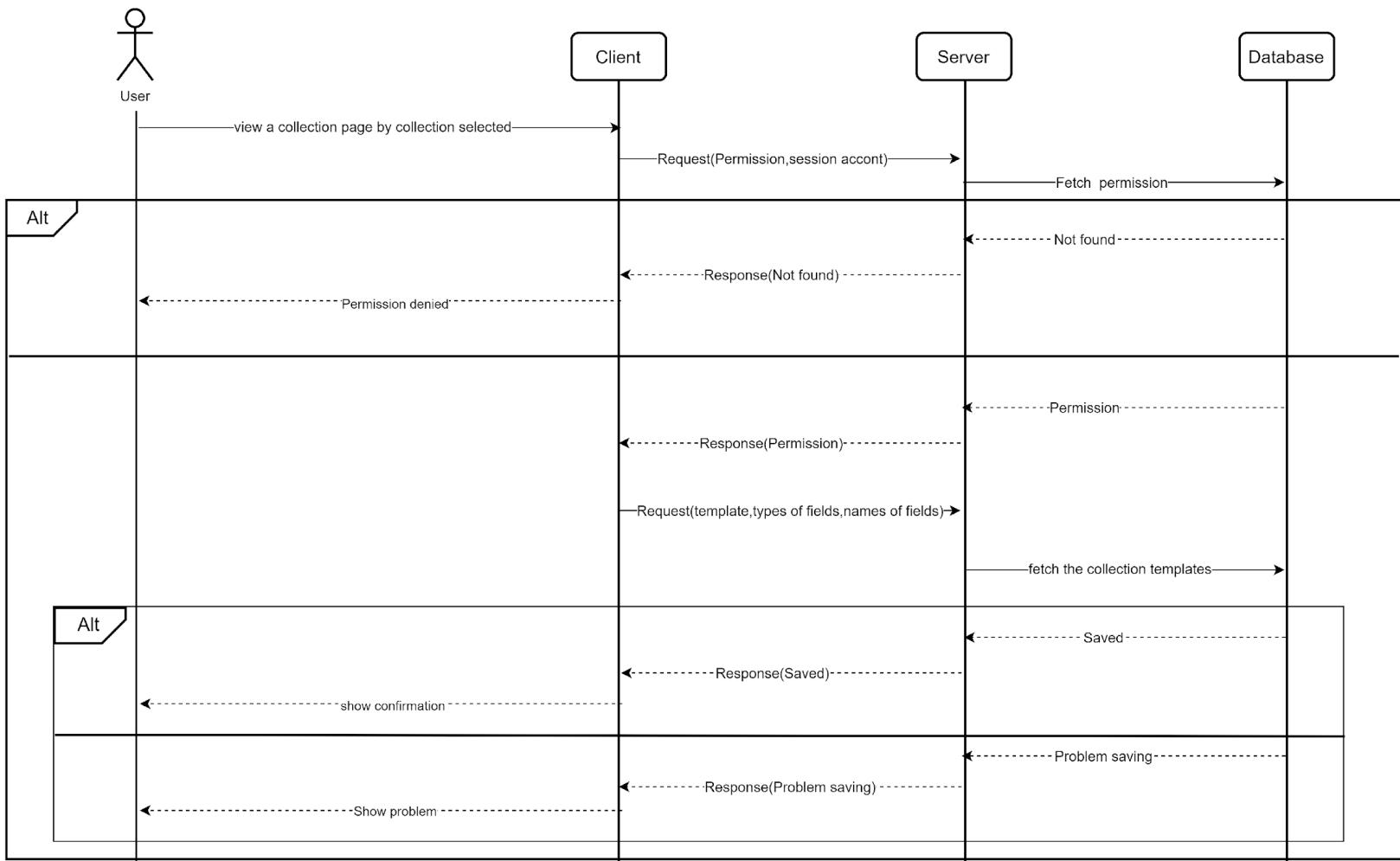
Soft delete organization



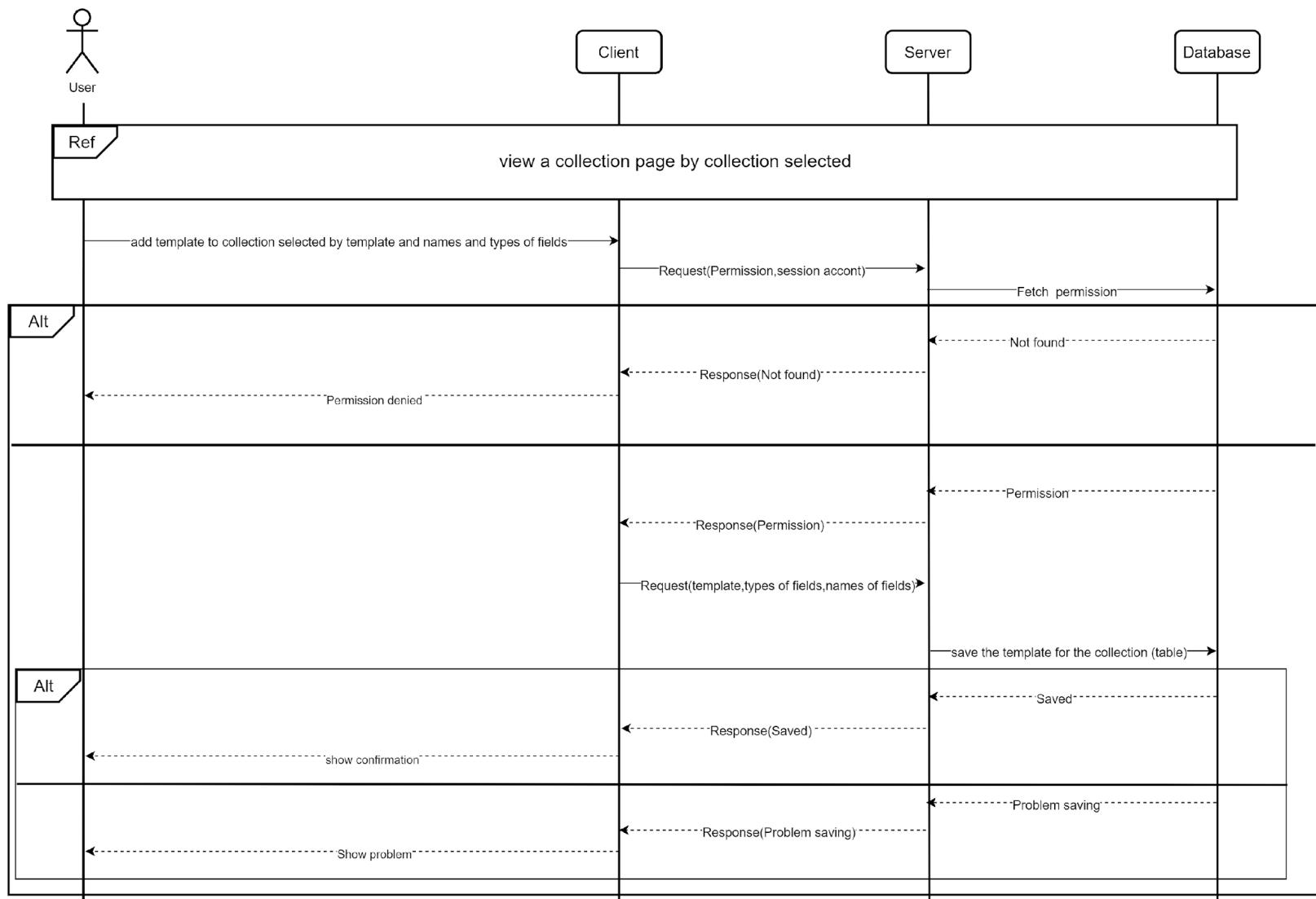
Create collection



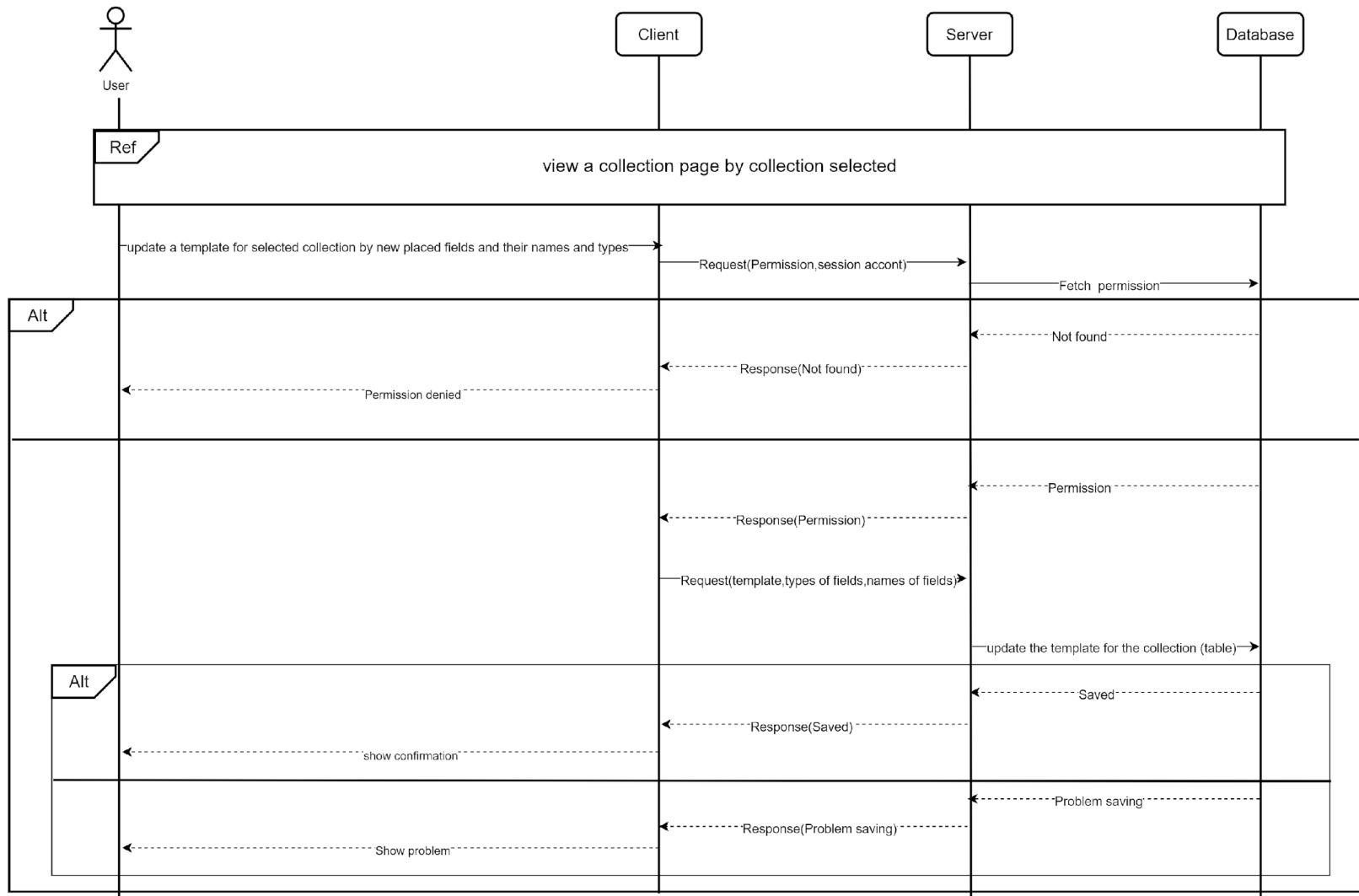
View collection page



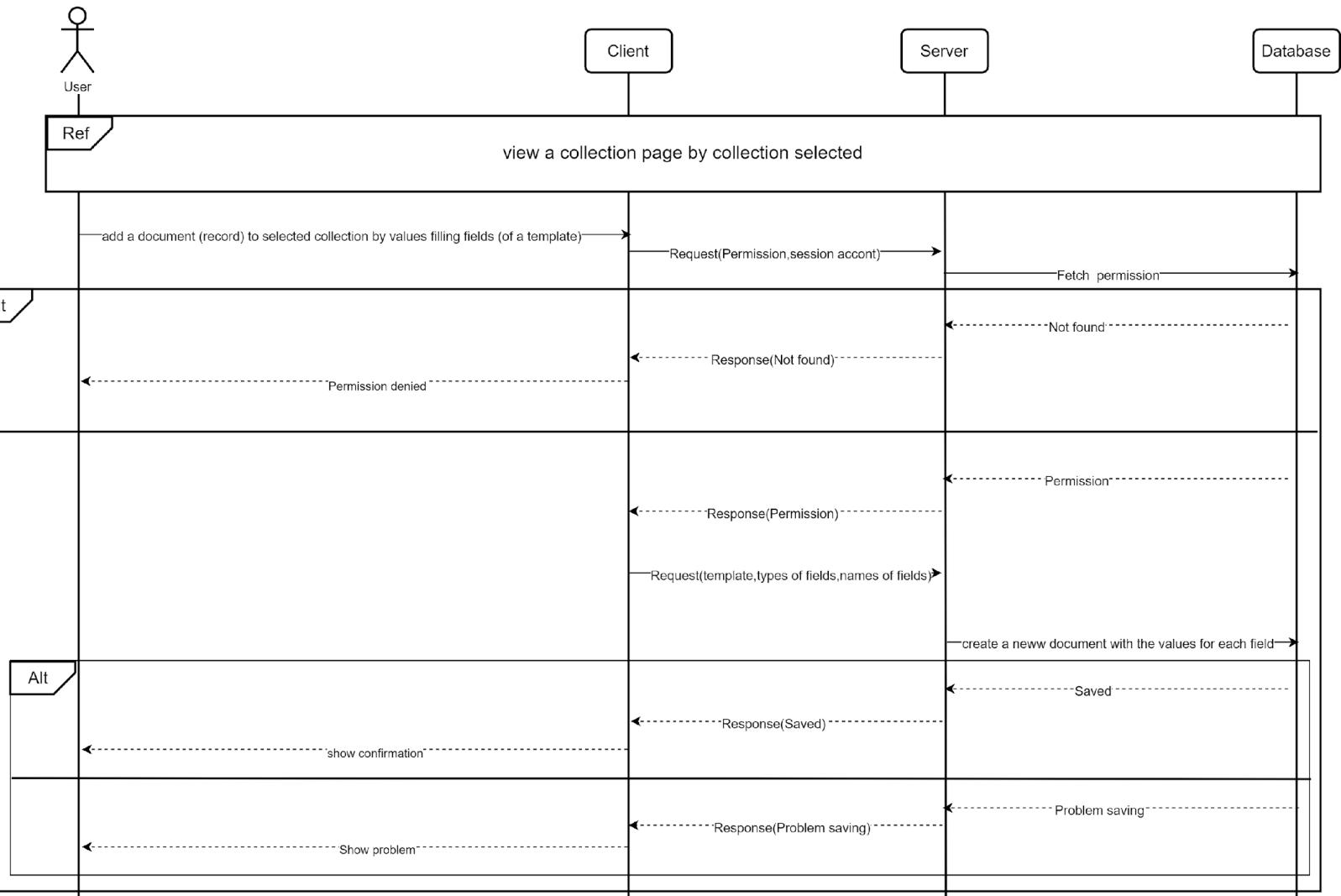
Add template to collection



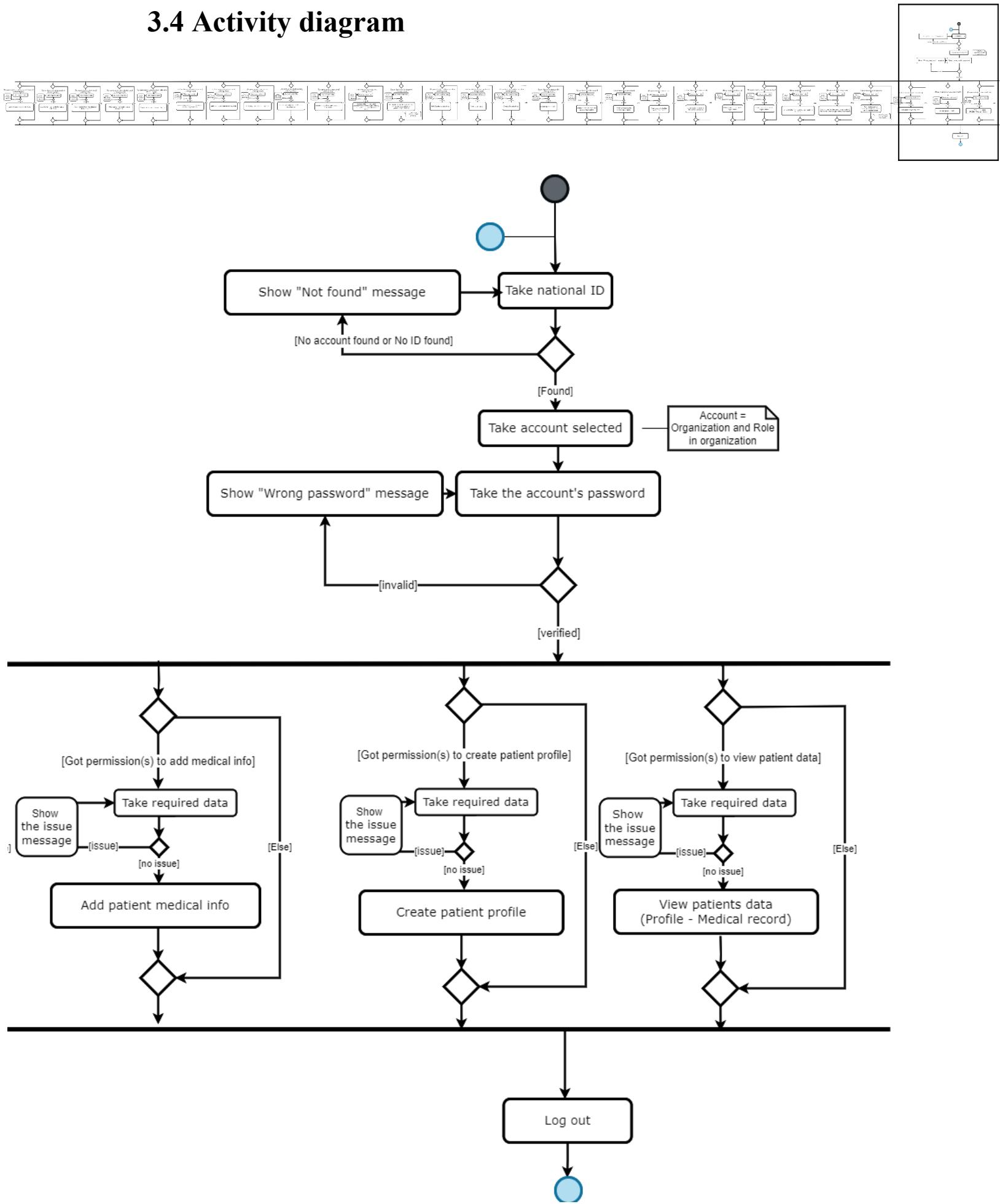
Edit template

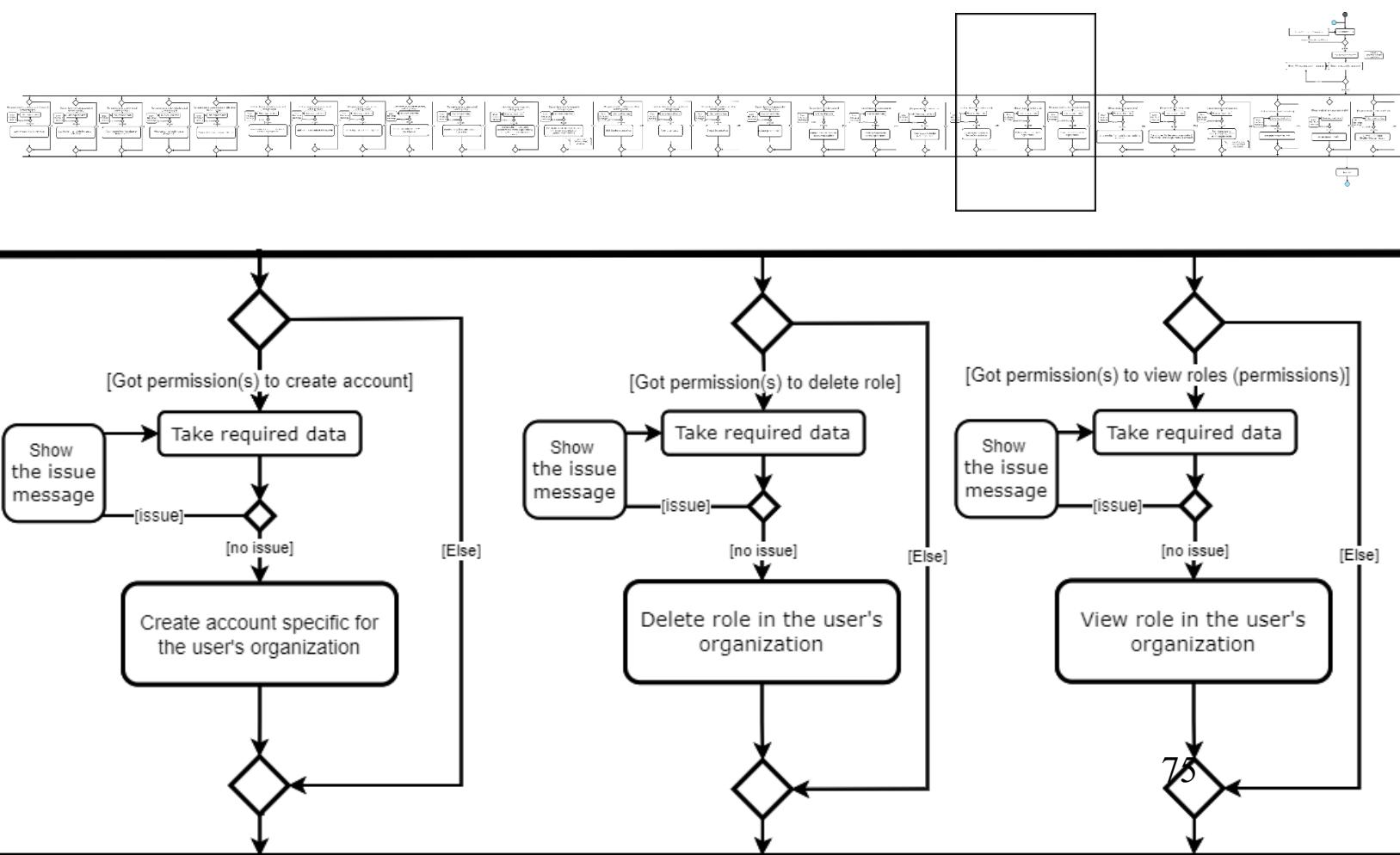
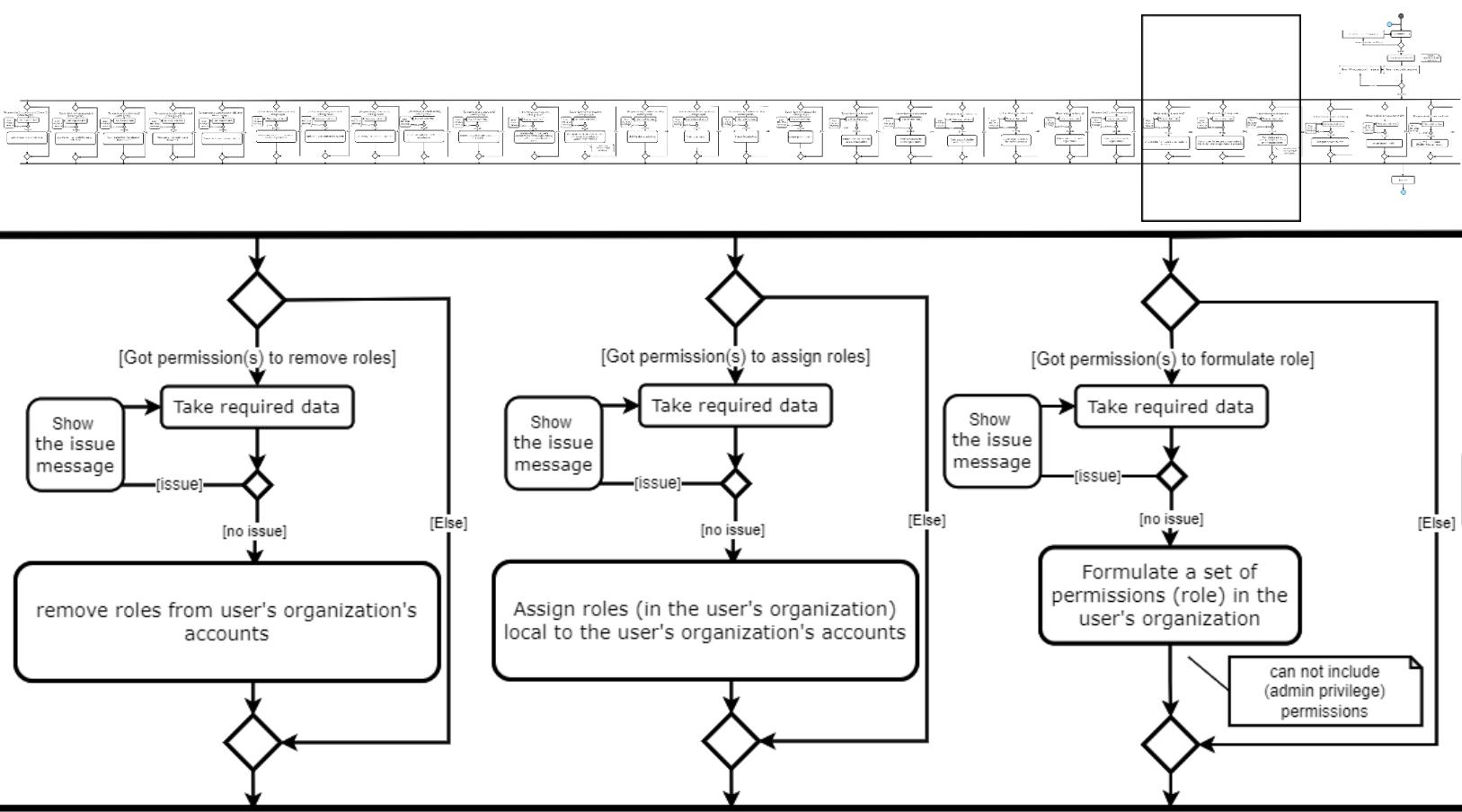


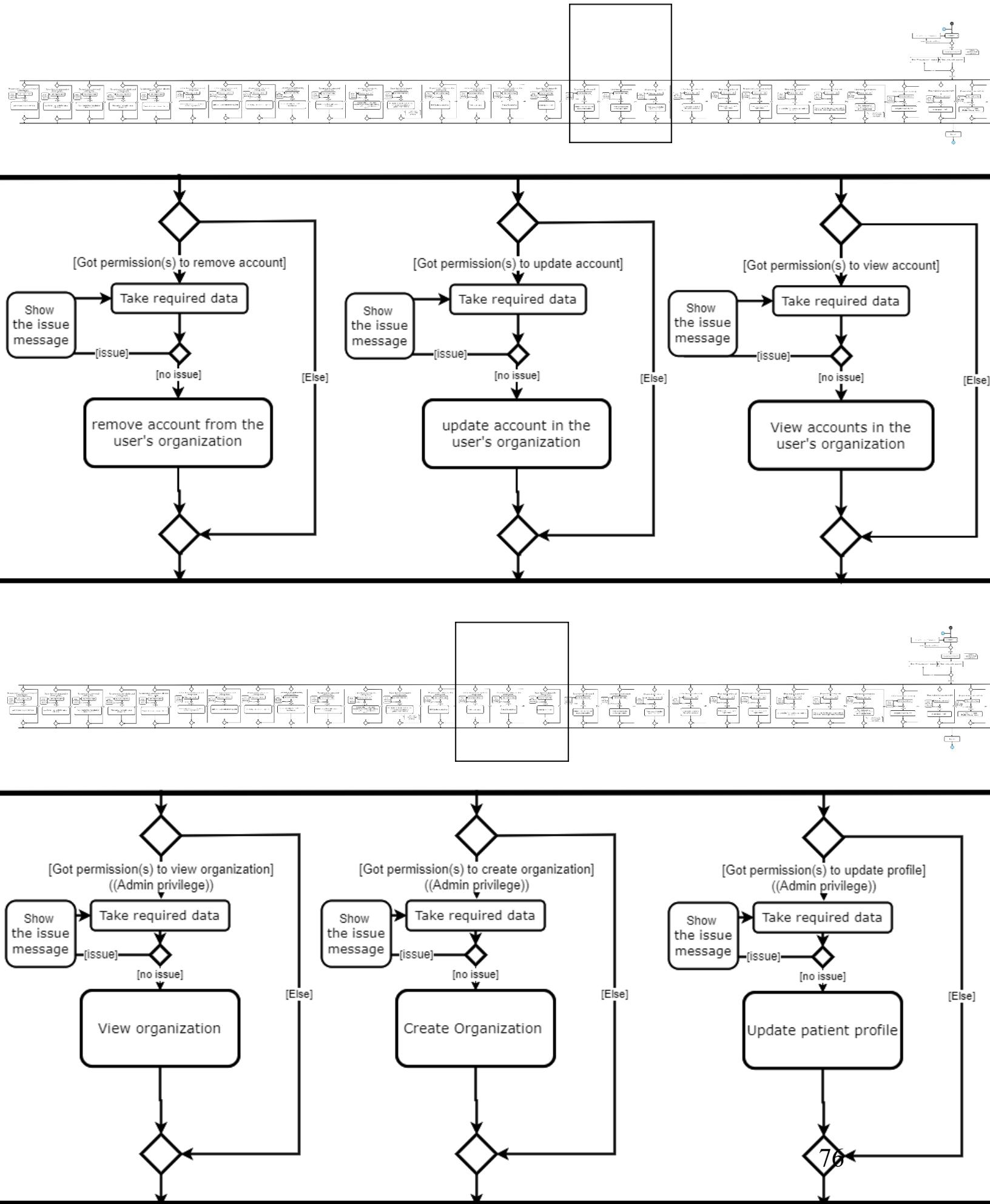
Add document (record) to collection

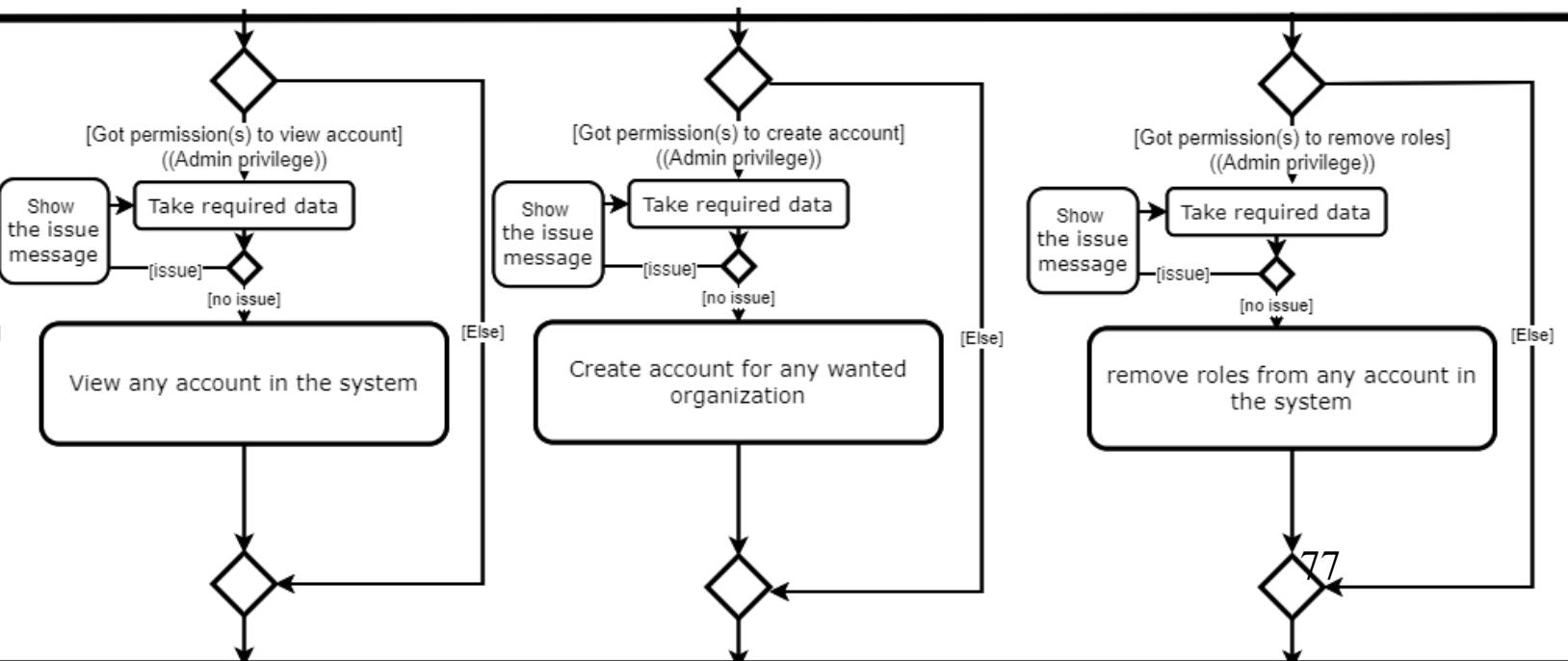
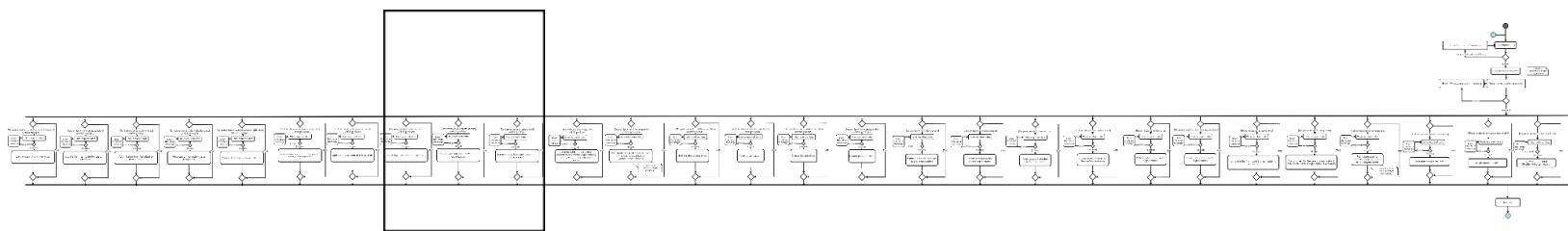
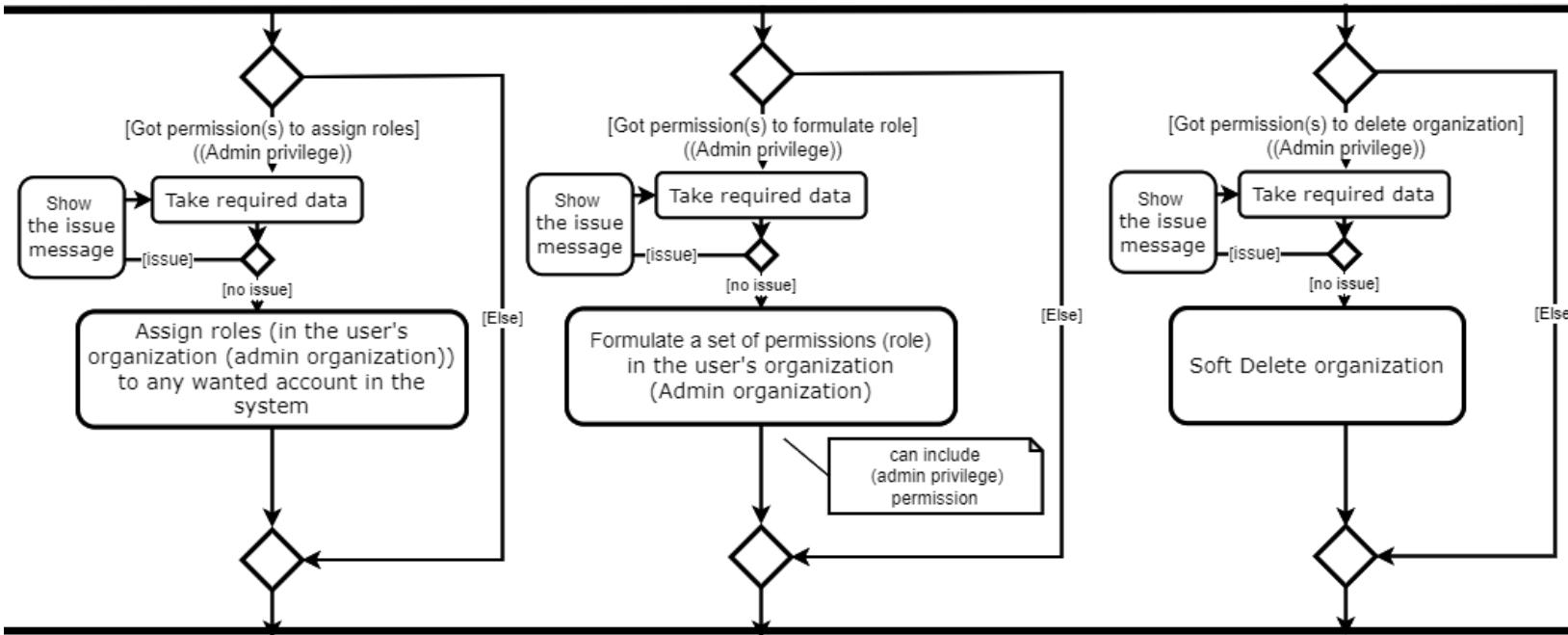
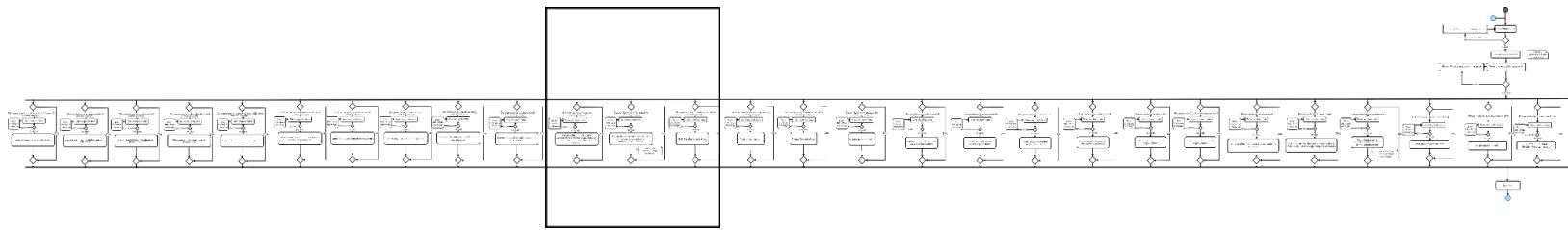


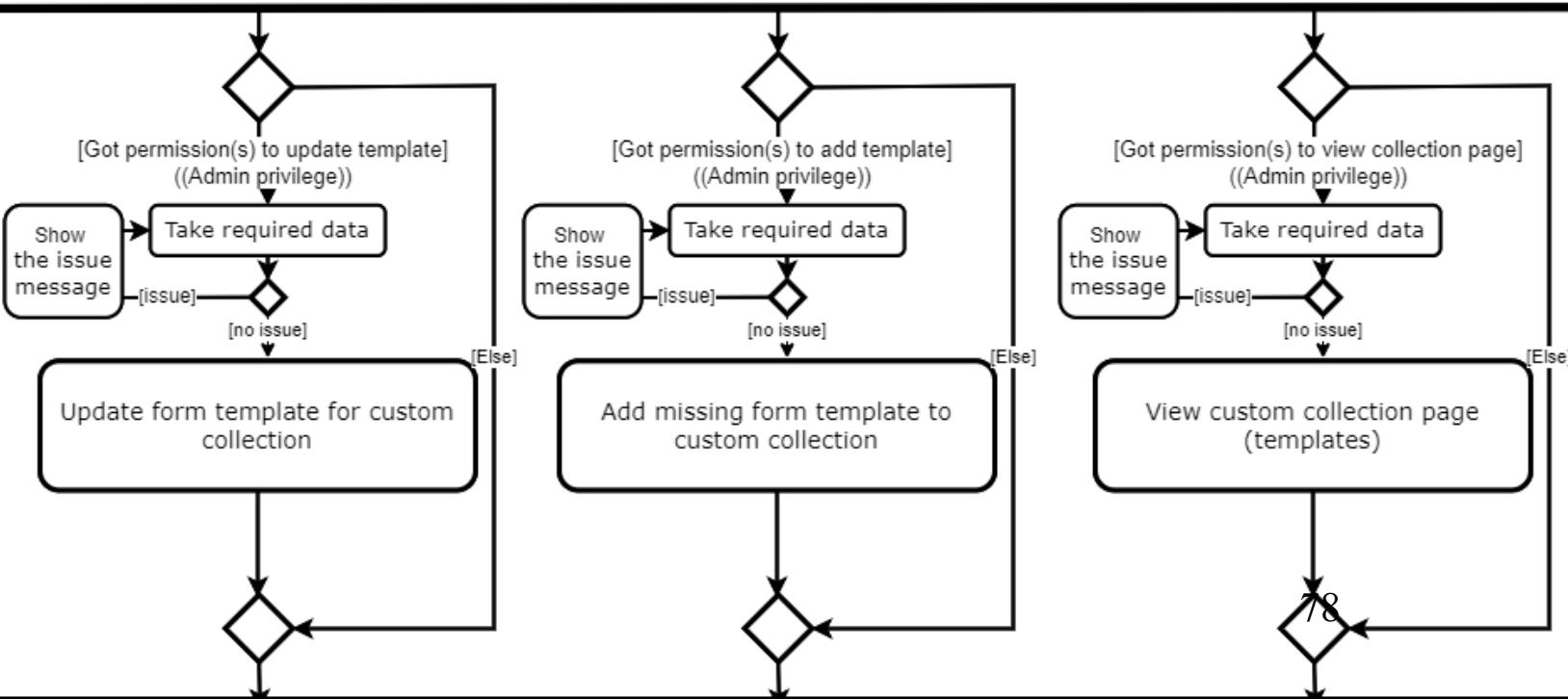
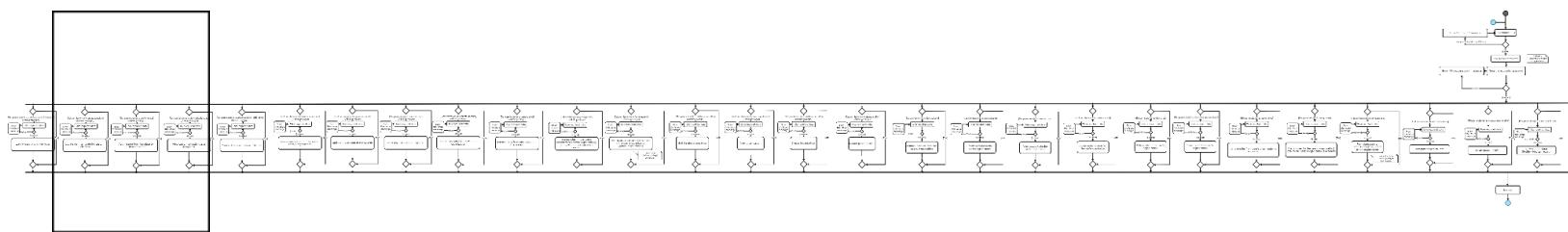
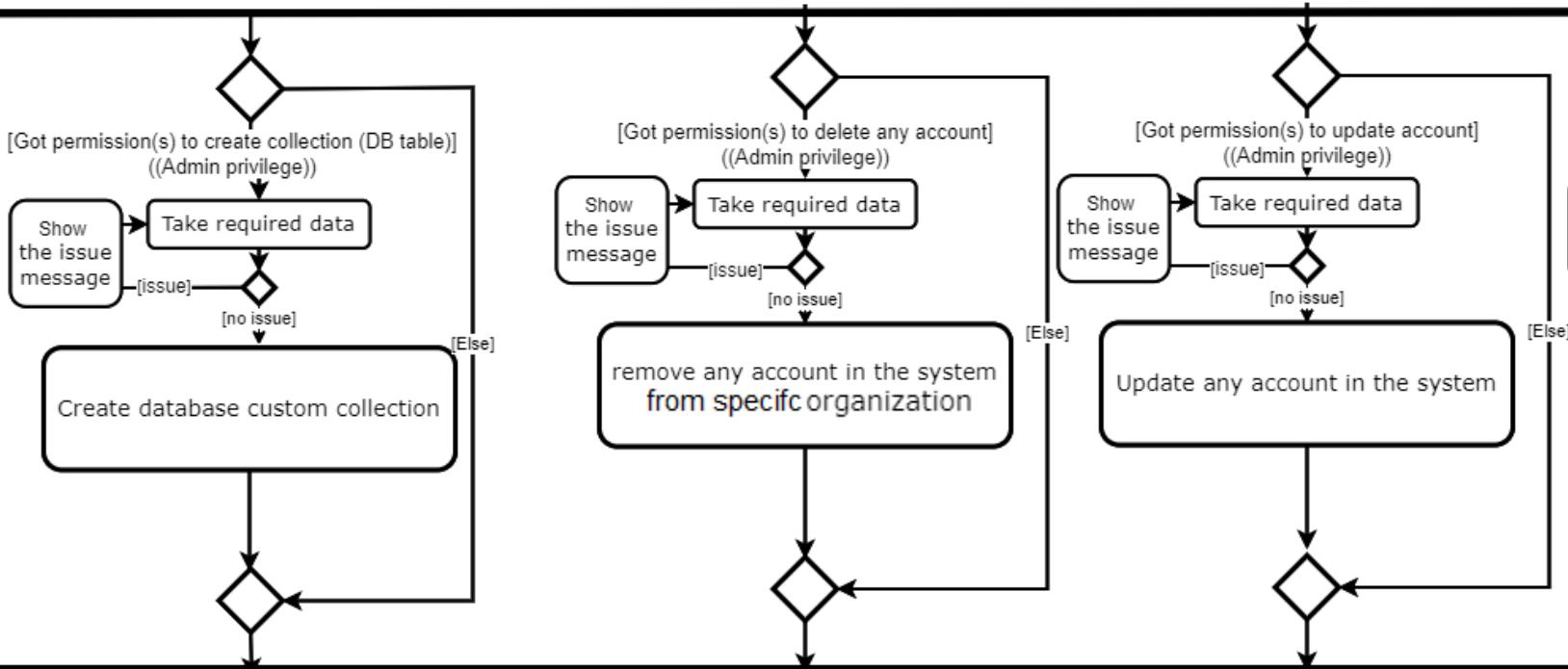
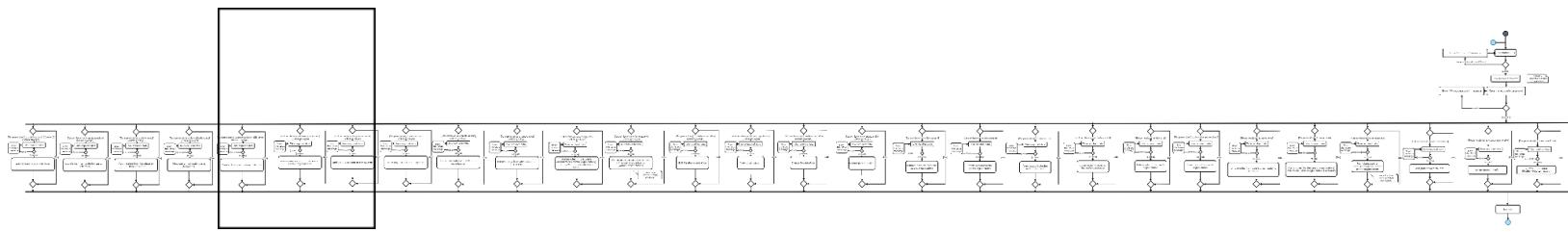
3.4 Activity diagram











[Got permission(s) to add document (DB record)
((Admin privilege))

Show
the issue
message

Take required data

[issue] [no issue]

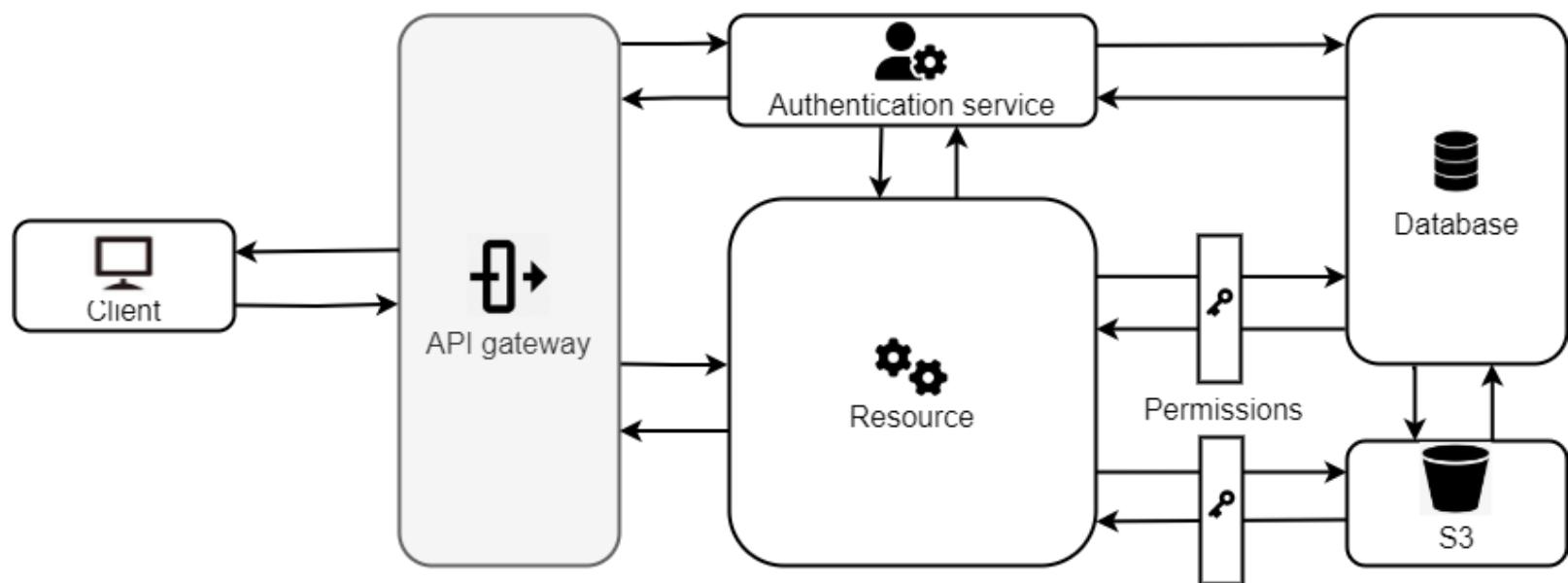
Add document in a collection



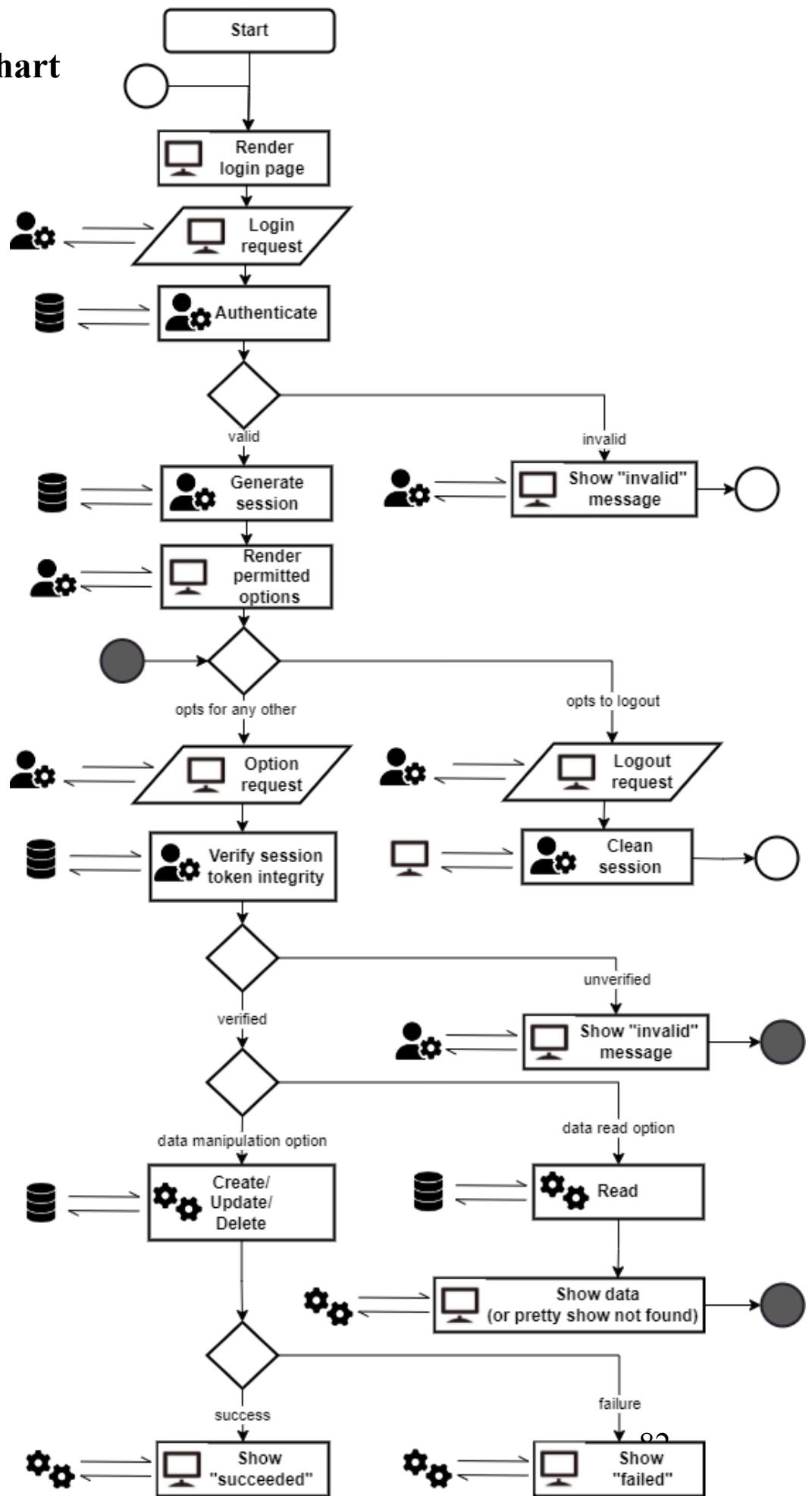
Chapter 4

Implementation

4.1 Software architecture



4.2 Flowchart





MHFP
Chapter 5
Testing

5.1 Functional testing

5.1.1 Unit Testing

During this phase, individual components of the Electronic Health Record (EHR) system were tested in isolation to identify and rectify defects. Comprehensive test cases cover various scenarios, verifying correctness and adherence to requirements. Unit testing guarantees reliability and performance.

5.1.2 Integration Testing

Testing the collaboration and integration between EHR system components was the focus here. Test cases encompass data flow, interfaces, and system behavior, detecting issues such as communication failures and data corruption. It ensures seamless integration, verifies adherence to functionalities (whether in the basic or alternative flows) and guarantees effective functionality.

5.1.3 Acceptance Testing

Representative end-users actively participate in evaluating system suitability. Real-world scenarios are simulated to assess functionality, user interface, performance, security, and regulatory compliance. Acceptance testing validates user needs, system behavior, and enhances user satisfaction.

5.1.4 Regression Testing

This phase verifies that modifications don't introduce new defects or impact existing functionality. Core functionality is tested using predefined test cases, ensuring no unintended side effects or regressions. Regression testing maintains system reliability, stability, and saves time through automation.

5.1.5 System Testing

The entire EHR system is evaluated to ensure it meets specified requirements and functions correctly in real-world scenarios. Thorough testing uncovers defects, errors, and inconsistencies, ensuring system reliability and meeting the needs

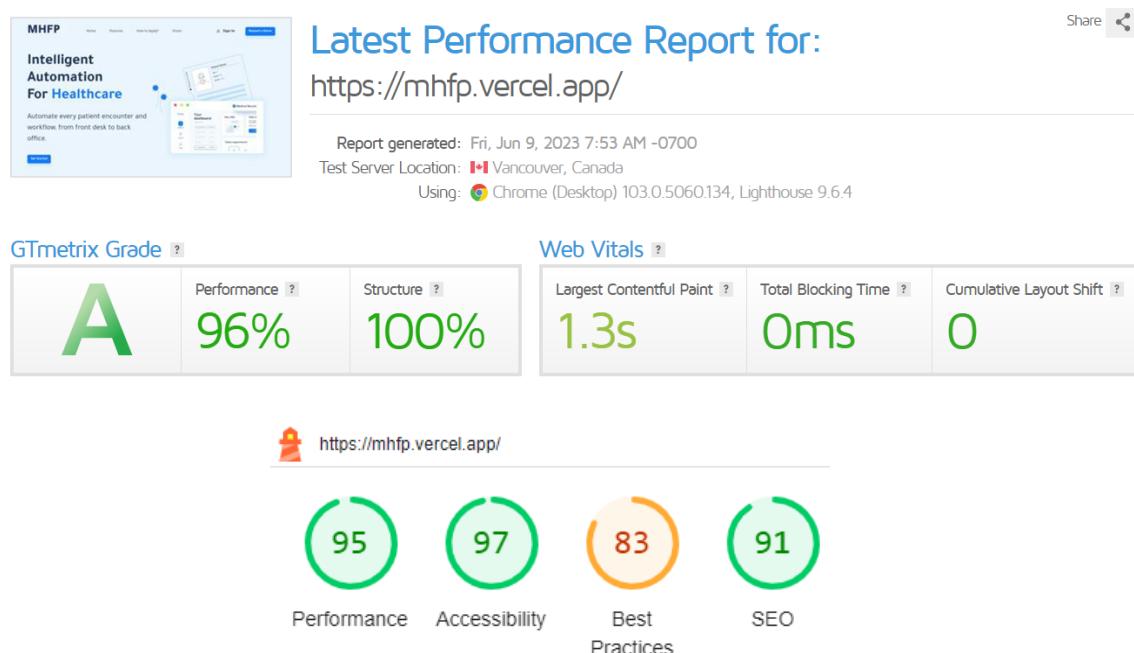
5.2 Non-Functional testing

5.2.1 Performance Testing

During the performance testing phase, we assess the responsiveness and efficiency of our EHR system under different workloads and scenarios. By simulating realistic usage patterns, we measure and analyze key performance metrics such as response time, throughput, and resource utilization. This testing verifies that our system performs optimally, meets performance goals, and provides a satisfactory user experience, even under high load conditions.

5.2.2 Stress Testing

Stress testing evaluates the system's stability and robustness by subjecting it to extreme workload conditions that exceed its normal operational capacity. This testing helps identify system bottlenecks, performance limitations, and potential failures. By applying heavy loads and stress factors, such as high user concurrency or large data volumes, we measure the system's behavior and assess its ability to handle stress. This ensures that our EHR system can withstand demanding situations without compromising its performance or integrity.



5.2.3 Additional test cases for suspicious behavior

In addition to the tests that have already passed for the basic and alternative flows of the system's stated functionalities, which ensured the acceptance of only valid inputs, rejection of repeated input in the database, successful execution of functionalities, and appropriate impact of execution, the following are additional tests for detecting suspicious behaviors:

Case description	Steps	Expected	Status
Inhibit patient manipulation by two users concurrently	User finds a patient and gets his data shown to view or manipulate Another user attempts to find the same patient	The second attempt is blocked indicating that the patient is in access by another user	Pass

Case description	Steps	Expected	Status
Inhibit unauthorized access	User (hacker) tries to access a UI unavailable option through browser's console	System checks the user's permissions in the DB to find incompatibility and inhibits	Pass

Case description	Steps	Expected	Status
Inhibit scripts injections	user attempts to inject malicious script in an input field	System doesn't execute the script, and instead processes it as plain text	Pass

Case description	Steps	Expected	Status
Limit rate of requests	user sends multiple requests in short period (would block the service (DDOS attack))	The system utilizes rate limiting to ignore all incoming requests from a user after exceeding a certain number of requests within a specific duration.	Pass



MHFP

Chapter 6

Results and Discussion

6.1 Results

6.1.1 Expected Results

Role-Based System:

- Operates on a role-based access control, granting users specific permissions that dictate their actions within the system.
- Users can perform functionalities according to the permissions they have been assigned.

User Account Management:

- Users can create accounts for others (or for themselves), provided they have the necessary permission.
- The system allows users with the necessary permissions to create organizations and associate user accounts with these organizations at account creation.
- User with the necessary permissions, can assign an account the permissions to create roles within their organization, create accounts within their organization, and assign roles to accounts within their organization. This set of permissions may be under a role named “Organization owner” in the user’s organization

Role Formulation and Assignment:

- Users can create custom roles within their organization, defining a name and adding permissions to the role.
- Role formulation can either be unrestricted, allowing users to include any permission in a new role and save it to his organization, or restricted, limiting the permissions to be added to a role, based on the user's own permissions, and this may be found in the “Organization owner” role
- The first users of the system (super users) possess all permissions and may be associated with the “System organization”, the first organization created during system implementation.

Patient Profile Management:

- Users can provide patient profile data to create patient profiles.
- Users can search for patients by their national ID and view their data.
- Users can add medical information to patient's medical record or scans

Dynamic Formatting of Medical Records:

- The system allows users to create custom categories for patient medical records, offering flexibility and dynamic formatting.
- Categories' content can be either user-specific or shareable between patients.
- User-specific categories contain content specific to each patient, establishing a many-to-one relationship, such as test results category
- Shareable categories store common information that can be shared among patients, establishing a many-to-many relationship, such as medications category.

Category Creation and Template Design:

- Users can create categories and define a template consisting of form fields to structure the data entry process.
- The template design provides flexibility in arranging fields and naming them accordingly.
- Users can determine whether a category necessitates the storage of shareable information -as records of the fields specified- in a database collection. This allows users to add such information using the template provided. On the other hand, if a category does not require shareable information and is just patient-specific, the template is exclusively used for filling data to patients within that category.
- Patient medical records initially do not include any categories until users add desired categories and enter data using the specified templates.
- Users can create categories that are not associated with a patient medical record. These categories are intended for the purpose of storing information using its template, and the categories then are rather said to be mere collections
- User can add more templates for the same collection (category) for viewing and printing purposes
- User can update a template by adding more fields

Statistical-Based Recommendations:

- The system enables providers to view a patient's medical history and receive statistical-based recommendations related to the patient's case.
- Periodic statistical calculations are performed on the system's patient data, generating insights into possible medications to be prescribed for a specific case
- When accessing a patient's medical record, the user is presented with a form-like display indicating the percentage match for various medications relevant to the patient.

6.1.2 Actual Results

Except for future work, the actual results meet the expected results. The final product achieves what it was expected to achieve. Here are some screenshots from the system's user interface

Patients page:

This screenshot shows the 'Patient' profile page. The left sidebar includes 'Dashboard', 'Patient' (selected), 'Collections', 'Organizations', 'Roles', and 'Settings'. A user profile 'KA Kerolous Amged' and a 'Signout' link are also present. The main area displays a 'Full Name' section with fields for First Name ('galaluddin'), Middle Name ('owais'), and Last Name ('gal'). A 'Save Profile' button is located at the top right. To the right, a search bar contains the ID '30106122101956' with a 'Reset' button below it. Below the name fields is an 'Addresses' section with a note: 'You Must Have At Least One Address'. It shows a city ('giza'), district ('omrania'), and postal code ('12511'). A street name field ('Ahmed hussein') is also present. At the bottom are fields for 'Date Of Birth' ('06/12/2001'), 'Gender' ('male'), and 'Phone Number' ('01143113410').

This screenshot shows the 'Record' tab of the Patient page. The left sidebar is identical to the previous screenshot. The main area features a navigation bar with tabs: Diseases, Medications (selected), Symptoms, Social History, Care Plan, Family History, Treatment, Immunizations, Allergies, Test, and Add. Below this is a search bar with 'Field Names' dropdown and a 'Search...' input field. The 'Medications' section displays two entries: 'Omez' and 'Flagyl'. Each entry includes details such as start date, end date, weight, administration route, and dose frequency.

Name	Start_date	End_date	Weight	Administration_route	Dose
Omez	2021-01-09		40mg	Oral	twice a day
Flagyl	2008-06-10	2015-06-10	100mg	Oral	twice a day

MHFP Helwan University Hospital

Profile Record

Diseases Medications Symptoms Social Care Family Treatment Immunizations Allergies Test Add

Patient

Search For Patient
30106122101956 Reset

Symptoms

Name: Acid reflux

Severity: Moderate

Progression: Slow

Persistence: Relatively long

Aggravating factors: Digestive issues

Body site: Digestive system

Start date: 2023-01-01

End date: 2023-12-31

Save Details

MHFP Helwan University Hospital

Profile Record

Diseases Medications Symptoms Social Care Family Treatment Immunizations Allergies Test Add

Patient

Search For Patient
30106122101956 Reset

Symptoms

Name: Acid reflux

Severity: Moderate

Progression: Slow

Persistence: Relatively long

Aggravating factors: Digestive issues

Body site: Digestive system

Start date: 2023-01-01

End date: 2023-12-31

Select Name

- Select Name
- Fatigue
- Polyuria
- Polydipsia
- Weight Loss
- Blurtness
- Chronic Wounds
- Shortness Of Breath
- Chronic Cough
- Wheezing
- Chest Pain
- Heartburn
- Acid Reflux
- Headache
- Muscle Pain
- Joint Pain
- Abdominal Pain
- Nausea
- Dizziness
- Insomnia

Collections page:

The screenshot shows the 'Collections' page of the MHFP application. On the left is a sidebar with navigation links: Dashboard, Patient, Collections (which is selected and highlighted in blue), Organizations, Roles, and Settings. Below the sidebar is a user profile section with a circular icon labeled 'KA' and the name 'Kerolous Amged', and a 'Signout' button. The main content area is titled 'Helwan University Hospital' and 'All Templates Templates Store'. It features a grid of eight cards, each representing a template type: 'Create A Collection', 'Diseases', 'Medications', 'Test_results', 'Family_history', 'Social_history', 'Immunizations', and 'Symptoms'. Each card has a small icon and a plus sign.

The screenshot shows the 'Create A Collection' page for the 'symptoms' template. The left sidebar is identical to the previous screenshot. The main form is titled 'Helwan University Hospital'. It includes fields for 'Name' (dropdown menu 'Select Name'), 'Severity' (dropdown menu 'Select Severity'), 'Progression' (dropdown menu 'Select Progression'), 'Persistence' (dropdown menu 'Select Persistence'), 'Aggravating Factors' (text input 'aggravating factors'), 'Body Site' (dropdown menu 'Select Body Site'), 'Start Date' (date input 'mm/dd/yyyy'), and 'End Date' (date input 'mm/dd/yyyy'). To the right of the form is a panel titled 'Template Details' containing fields for 'Collection Name' ('symptoms'), 'Description' ('patient-shared clinical symptoms'), and 'Template Name' ('patient'). There are also sections for 'Preferences' with checkboxes for 'Printable', 'Patient Profile' (which is checked), 'Patient Specific', and 'Public'. A large blue 'Save' button is at the bottom right.

MHFP

Helwan University Hospital

Name

- Fatigue
- Polyuria
- Polydipsia
- Weight Loss
- Blurrness
- Chronic Wounds
- Shortness Of Breath
- Chronic Cough
- Wheezing
- Chest Pain
- Heartburn
- Acid Reflux
- Headache
- Muscle Pain
- Joint Pain
- Abdominal Pain
- Nausea
- Dizziness
- Insomnia

Persistence

Body Site

Select Persistence

End Date

Enable builder mode

Template Details

Collection Name: symptoms

Description: patient-shared clinical symptoms

Template Name: patient

Preferences:

Printable

Patient Profile

Patient Specific

Public

Save

MHFP

Helwan University Hospital

Name

Severity

Progression

Persistence

Aggravating Factors

Body Site

Start Date

End Date

Disable builder mode

Template Details

Collection Name: symptoms

Description: patient-shared clinical symptoms

Template Name: patient

Preferences:

Printable

Patient Profile

Patient Specific

Public

Save

Roles page:

6.2 Discussion

Some minor modifications were made to streamline the flow of certain functionalities, making the system more user-friendly and intuitive to use.



MHFP

Chapter 7

Conclusion

In conclusion, the electronic health record (EHR) system has successfully achieved its objectives of simplifying documentation and recording, providing easy and fast data access and updating, facilitating the tracking of patient history, offering an intuitive user interface, streamlining the onboarding process for employees, and ensuring data safety and security. The system's role-based access control, user account management, dynamic formatting of medical records, and statistical-based recommendations have been effectively implemented. Through comprehensive testing, the system has demonstrated reliability, performance, and adherence to requirements.

Looking ahead, the EHR system can be further enhanced to meet the evolving needs of healthcare. One area of focus should be to leverage paid APIs and datasets to adopt advanced features, such as drug-drug interaction alerts and other decision support tools, to ensure medication safety and optimize patient care. By integrating these external resources, the system can provide healthcare providers with real-time alerts and recommendations regarding potential drug interactions, allergies, and adverse effects.

Additionally, other paid enhancements, such as advanced data visualization tools for reporting and analytics, can be explored to derive meaningful insights from the collected patient data, enabling evidence-based decision making and improving overall healthcare outcomes.



Chapter 8

Future work

In terms of future work, several areas can be further developed to enhance the electronic health record (EHR) system.

Firstly, there is a need to include billing handling capabilities within the system, which will streamline financial processes such as billing, insurance claims, and payment management. This addition will contribute to a more seamless and efficient administrative workflow.

Secondly, improvements can be made to the appointment handling system. By optimizing scheduling processes and resource allocation, patient waiting times can be minimized.

Furthermore, the integration of drug-drug or drug-allergy interaction alerts is crucial to enhancing patient safety. By leveraging relevant databases and clinical decision support systems, the EHR system can provide real-time warnings to prevent adverse drug events, ensuring optimal medication management.

In addition, investing in advanced artificial intelligence (AI) decision support capabilities will enable the system to analyze patient data to further improve decision-making and patient outcomes.

Lastly, there is a need to enhance data visualization capabilities within the system. By incorporating interactive charts, graphs, and dashboards, healthcare providers can gain intuitive insights into patient trends, outcomes, and population health metrics. This would support informative and impactful reporting.

Bibliography

Abdelgaber, Sayed & Abdel-Fattah, Manal & M., Sameh & M.helal, Sameh. (2017). A Roadmap to Implement EHR Nationwide in Egypt. Communications on Applied Electronics.

Boshnak, Hussein & Abdelgaber, Sayed & Yehia, Engy & Abdo, A.. (2019). Ontology-Based Knowledge Modelling for Clinical Data Representation in Electronic Health Records. International Journal of Computer Science and Information Security.

Yehia, Engy & Boshnak, Hussein & Abdelgaber, Sayed & Abdo, A. & Elzanfaly, Doaa. (2019). Ontology-Based Clinical Information Extraction from Physician's Free-Text Notes. Journal of Biomedical Informatics.

Bernd Bruegge, Allen H. Dieton , Object-Oriented Software Engineering: Using UML, Patterns and Java, 3rd Edition, Prentice Hall 2009.