**Chapter 1: Introduction**

* **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.
* **Objectives**
  1. 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

* 1. To provide organized information about patients to help decision making. The information can include:

- medical and social history

- family linkage

- tests results

- allergies

* 1. To handle billing information and financial aspects
  2. To provide data visualization and reporting
  3. To allow data exchange with other systems
  4. To provide decision support
  5. To assure data safety and air tight security:

- foolproof identity validation

- data access tracking

- verifying patient presence during data access

- fine-grained authorization (project’s main ideology)

- Role-Based Access Control (RBAC)

* **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

* **Scope**

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 constrains:

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

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 constrains

1. **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. **Coding**
   * Deciding the programming paradigm
     + A combination of OOP (Object Oriented Programming) and functional programming paradigms
   * Determining programming languages and editors to be used
     + NodeJs as a back-end environment.
     + Modern client-side technologies (Flutter, React.js, etc.)
     + 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. **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. **Documentation**

In addition to this introduction, all the points mentioned above will be elaborated in this documentation once been complete

* **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**

* **Project planning**

1. **Feasibility study**

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

* **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 linkage to family history, and demographic data.

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 sessions (or appointments) scheduling and payment registration.

EHR systems keep track of billing information

EHR systems provide data visualization and reporting about the healthcare facility.

EHR systems allow secure patient data exchange with other EHR systems

This EHR system will allow providers to start a scheduled session and record findings, medications, or results. In addition to mark present findings or medications as obsolete (flagging them as from today’s history).

This system will allow providers to view a patient's medical history and receive relevant suggestions related to their case.

This EHR system will allow to dynamically format the components of patient form and the patient form it self with the guarantee of data consistency

This EHR system project will provide decision support through statistical calculations to display the likelihood of possible diagnosis, medications, etc.. to be related to a specific patient, based on the other patients’ data residing in the system.

This EHR system will allow to estimate an unknown information based on the other patients’ data residing in the system.

This EHR system will provide alerts for medications interactions and allergies

* **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.

1. **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

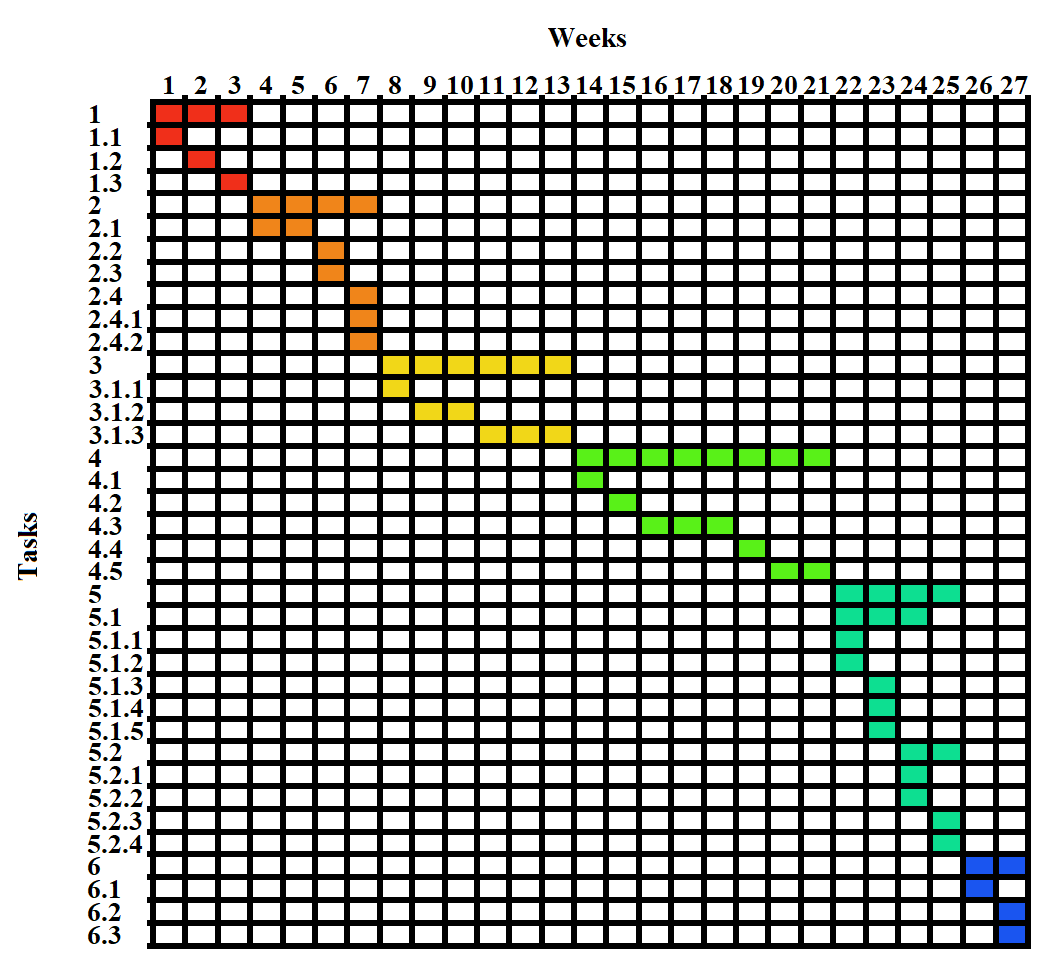
**For Organizations to purchase and set up the system:**

-The annual license fee for the software application might be $8,000

-For the customers adopting electronic documentation for the first time, there would be a hardware budget, including, essentially, computers, mobile devices, database servers, and network devices.

Opting for a cloud-based solution can save the customer money on expensive database servers; additionally, this can enable secure backups stored off-premises.

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

1. **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

* **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.

* **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.

* **Analysis of the new system**

1. **System requirements**

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

* Hardware requirements
  + Database server
  + Network devices
  + Computer devices
  + Hand-held devices (Mobile devices or tablets)
  + Printer devices
* Software requirements
  + Operating system
  + Web browser
  + Mobile compatibility
* Performance requirements
  + Automatic load balancing
  + Stand-by redundant (back-up) server
  + High speed internet
* Security requirements
  + Authentication, authorization, and access control
  + Physical security
  + Operating system with firewall software installed
  + Firewall router devices
  + Back-up database
* Compliance requirements
  + Raising staff awareness
  + Setting policies

1. **User requirements**

The user expects the system to support:

* + Ease of use
  + Security and privacy
  + Accessibility and availability
  + Patient data access and manipulation
  + Appointment scheduling
  + Billing
  + Reporting
  + Decision support
  + Interoperability

1. **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 by newly-hired users

1. **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 (restricted to user’s organization) |
| --- | --- |
| prerequisites | Internet connection  User’s account has “create account” permission |
| Pre-conditions | User is logged in  User has navigated to create account 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 | View a patient’s profile |
| --- | --- |
| prerequisites | Internet connection  User’s account has “access patient data” permission |
| 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 | View a patient’s medical record |
| --- | --- |
| prerequisites | Internet connection  User’s account has “access patient data” permission |
| 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, each marked with either active or stopped |
| Alternative flow | If entered ID is invalid or not found 1-System prompts a message specifying the issue   1. Step 1 in basic flow   If there is no available information  1-System Shows “nothing” page |
| Post-conditions | Desired patient’s medical record is viewed |

| Name | View a patient’s Scans |
| --- | --- |
| prerequisites | Internet connection  User’s account has “access patient data” permission |
| 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 scans tab  4-System shows scan images |
| Alternative flow | If entered ID is invalid or not found 1-System prompts a message specifying the issue  2-Step 1 in basic flow  If there is no available information  1-System Shows “nothing” page |
| Post-conditions | Scans are viewed |

| Name | Create patient profile |
| --- | --- |
| prerequisites | Internet connection  User’s account has “create patient profile” permission |
| Pre-conditions | User is logged in  User has navigated to create patient page |
| Basic flow | 1-User enters new patient’s ID, first name, last name, phone number, phone number of first repsonders (at least 2), address, secondary address (optional), image (optional),  2- System shows “profile created” message |
| Alternative flow | If entered data is invalid (incomplete fields or invalid ID or phone numbers..etc.) 1-System prompts a message specifying the issue  2-Step 1 in basic flow |
| Post-conditions | Patient profile is created |

| Name | Update a patient’s profile |
| --- | --- |
| prerequisites | Internet connection  User’s account has “update patient profile” permission |
| 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 plus update button next to each field  5-User clicks on the update button for the field he wishes to update  6-System prompts to enter the new value 7-User enters the new value 8-System updates the field |
| Alternative flow | If entered value is invalid: 1-System prompts a message specifying the issue  2-Step 7 in basic flow |
| Post-conditions | Patient profile is updated |

| Name | Add a patient’s medical information |
| --- | --- |
| prerequisites | Internet connection  User’s account has “update patient medical data” permission |
| 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-The system displays the medical record with (+) sign  5-User clicks on (+) sign  6-The system pops up a list of components (symptom, medication, allergy.. etc.)  7-User selects the desired component  8-The system prompts the user to fill in related information corresponding to DB table fields (it may need to retrieve data for a dropdown list)  9-User enters information.  The system saves the new information along with the date and place of entry. |
| Alternative flow | If entered data is invalid (incomplete) 1-System prompts a message specifying the issue  2-Step 9 in basic flow |
| Post-conditions | Medical information is added for the desired patient |

| Name | Stop a patient’s medical information |
| --- | --- |
| prerequisites | Internet connection  User’s account has “update patient medical data” permission |
| 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 the medical information with stoppage button next to each piece of information  5-User clicks the button for the information he wants to mark as stopped  6-System prompts a confirmation prompt  7-User confirms 8-System hashes the information and saves the stoppage date next to it |
| Alternative flow | If user doesn’t confirm:  1-step 4 in basic flow |
| Post-conditions | Medical information for the desired patient is marked as stopped |

| Name | Add a patient’s scan |
| --- | --- |
| prerequisites | Internet connection  User’s account has “update patient medical data” permission |
| 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 scan tab  4-System shows scan images with (+) sign  5-User clicks on (+) sign  6-System trigger the Users OS to open the file selector dialog  7-User picks the image he wishes to add  8-System saves |
| Alternative flow | -- |
| Post-conditions | Scan image is added for the desired patient |

| Name | Add billing\* |
| --- | --- |
| prerequisites | Internet connection  User’s account has “add billing” permission |
| Pre-conditions | -User is logged in  - |
| Basic flow |  |
| Alternative flow | -- |
| Post-conditions | Scan image is added for the desired patient |

| Name | Assign role to an account (restricted to account in user’s same organization) |
| --- | --- |
| prerequisites | Internet connection  User’s account has “assign roles” permission |
| Pre-conditions | User is logged in  User has navigated to assign 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 | Formulate Role (Restricted) |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Formulate role” permission |
| Pre-conditions | User is logged in  User has navigated to formulate role page |
| Basic flow | 1. System shows a list of permissions (excluding admin only permissions) 2. User picks permissions he wants to from a role by, and enters the role name and submits 3. System saves the role to the user’s organization |
| Alternative flow | -- |
| Post-conditions | Role is saved in the user’s same organization |

| Name | Create Organization |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| Pre-conditions | User is logged in  User has navigated to create organization page |
| Basic flow | 1. User fills in the data of the new organization (Name, address.. etc.) and submits 2. 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 | Assign role to any account |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| Pre-conditions | User is logged in  User has navigated to assign roles 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 | Scan image is added for the desired patient |

| Name | Formulate Role |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| Pre-conditions | User is logged in  User has navigated to formulate role page |
| Basic flow | 1-System shows a list of permissions (including admin only permission)  2-User picks 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 | Account is created in the desired organization |

| Name | Create account in any organization |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| Pre-conditions | User is logged in  User has navigated to create account page |
| Basic flow | 1. System shows a list of all the organizations in the system 2. 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 2-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 | Scan image is added for the desired patient |

| Name | Create Database Collection (Table) |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| Pre-conditions | User is logged in  User has navigated to collections page |
| Basic flow | 1. System shows all collections created with (+) sign 2. User clicks on (+) sign 3. System shows a form to add fields to by a (+) sign 4. User adds the number of fields he wants, specifies the type of each (text, number, select field), and names the collection and submits 5. System saves the collection |
| Alternative flow | -- |
| Post-conditions | New collection is created in the Database |

| Name | Update Database Collection (Table) |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| 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 update 3. System shows the fields of the collection with (+) and (-) signs 4. User adds or removes a field through the (-) and (+) signs, and submit the changes 5. System saves |
| Alternative flow | -- |
| Post-conditions | Desired collection is updated in the Database |

| Name | Delete Database Collection (Table) |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| Pre-conditions | User is logged in  User has navigated to collections page |
| Basic flow | 1. System shows all collections created by name with a delete button associated with each 2. User clicks on the delete button for the collection he wishes to delete 3. System prompts a confirmation prompts 4. User confirms 5-System deleltes |
| Alternative flow | If user doesn’t confirm:  1-step 1 in basic flow |
| Post-conditions | Desired collection is deleted from the Database |

| Name | Update a patient’s medical info. |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| 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 the medical information with update button next to each piece of information  5-User clicks the button for the information he wants to change  6-System opens the component of the information editable  7-User enters the new values 8-System saves |
| Alternative flow | -- |
| Post-conditions | Desired Medical information for the desired patient is updated |

| Name | Delete a patient |
| --- | --- |
| prerequisites | Internet connection  User’s account has “Admin only” permission |
| 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 and a delete button  3-User clicks on the delete button  4-System prompts a confirmation prompt  5-User confirms  6-System deletes |
| Alternative flow | If user doesn’t confirm:  1-step 2 in basic flow |
| Post-conditions | Desired patient is deleted |

1. **Non-functional requirements**

1-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

2-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.

3-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).

4-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.

5-Portability: The system should run on various operating systems and devices like desktop and mobiles.

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

7-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.

8-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.

9-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.

10-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.

1. **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, ensuring 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.

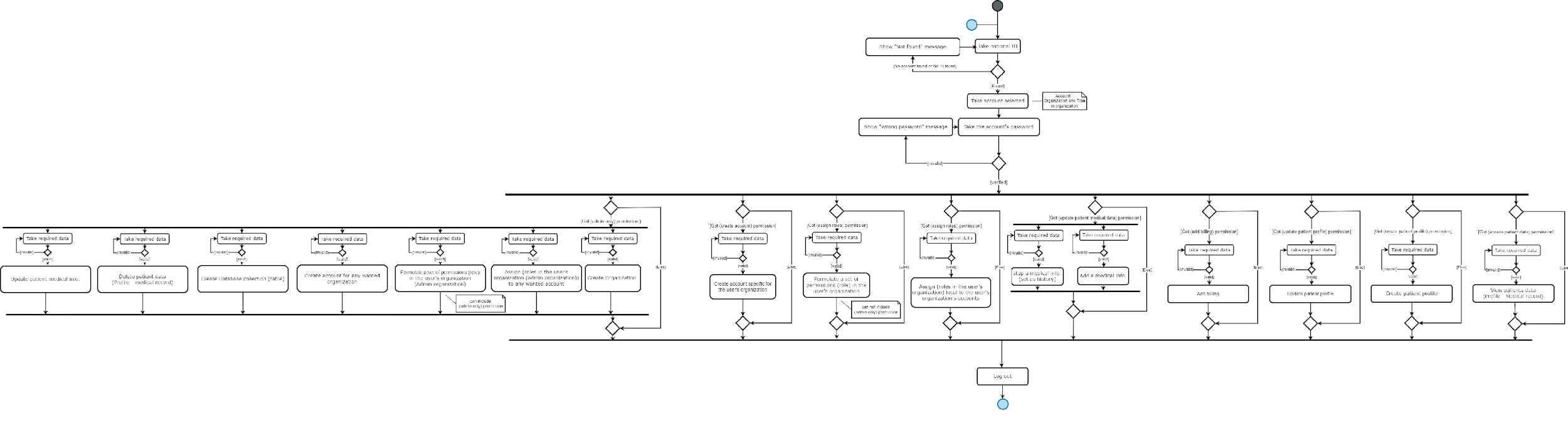
Decision support features, leveraging statistical calculations and patient data to provide recommendations and improve diagnostic accuracy.

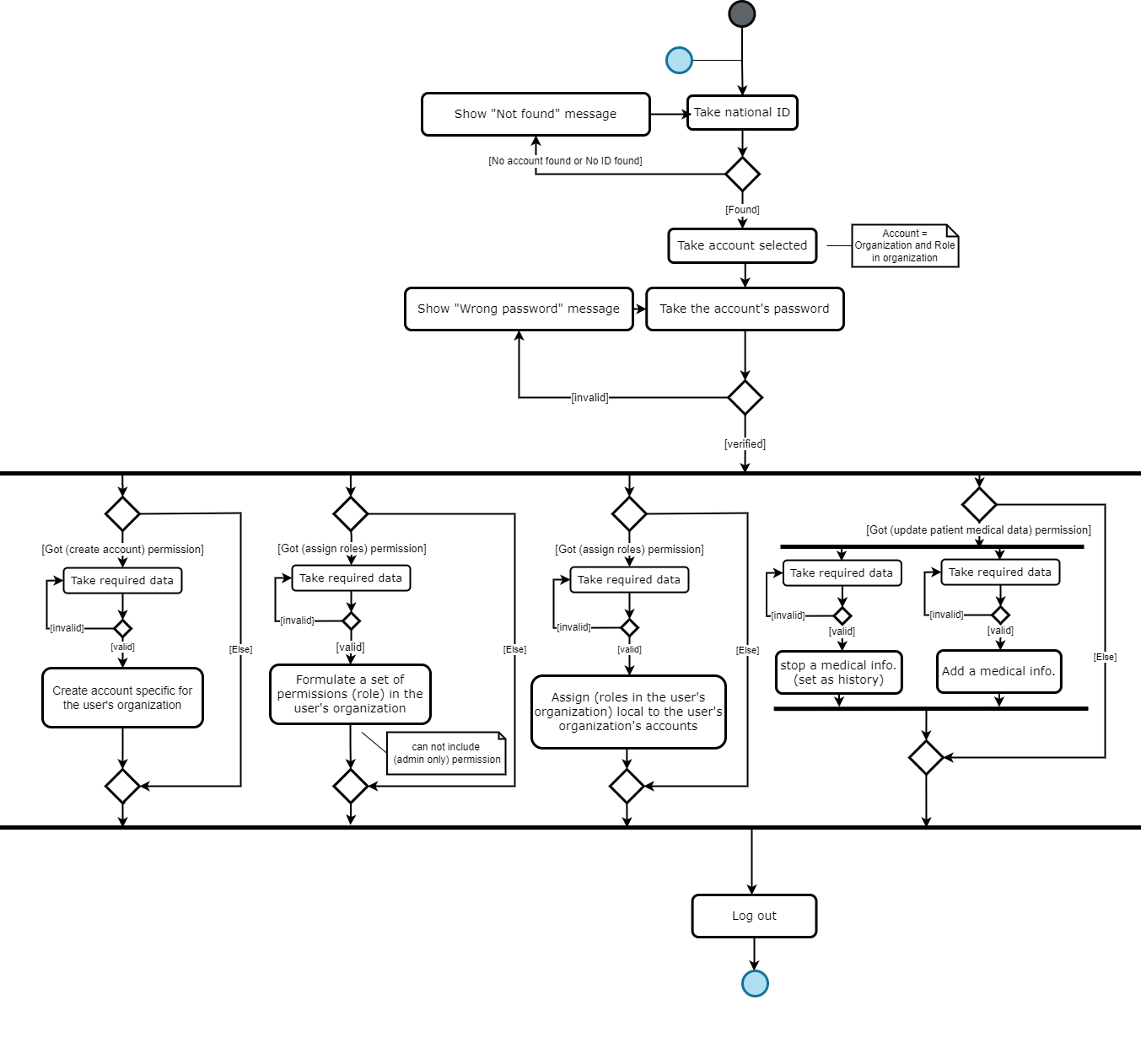
Alerts for medication interactions, enhancing patient safety and reducing the risk of adverse events.

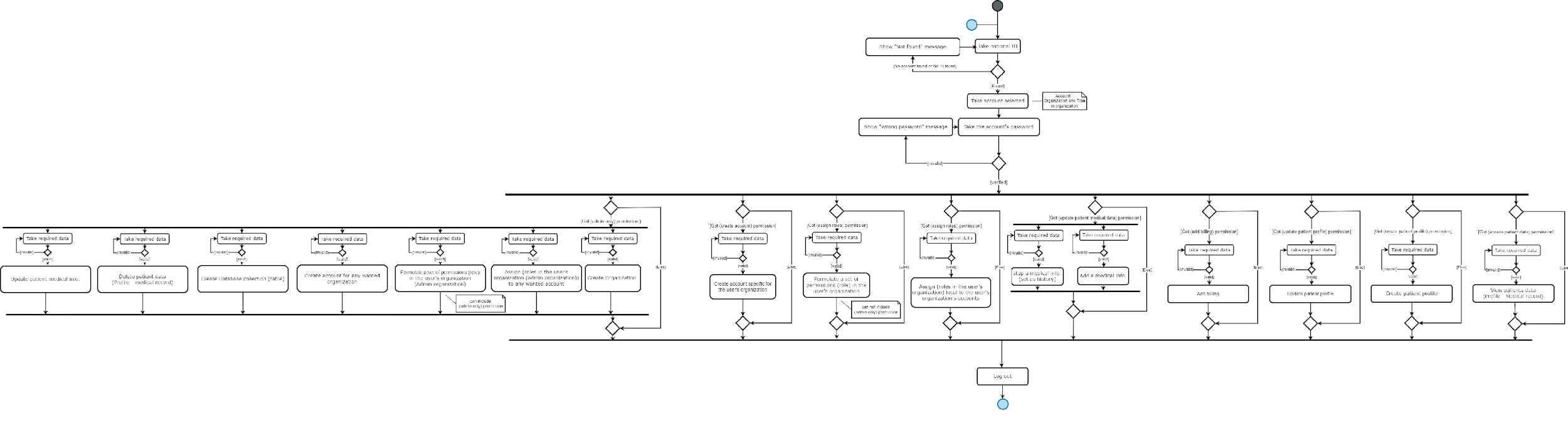
1. **Risk and risk management**

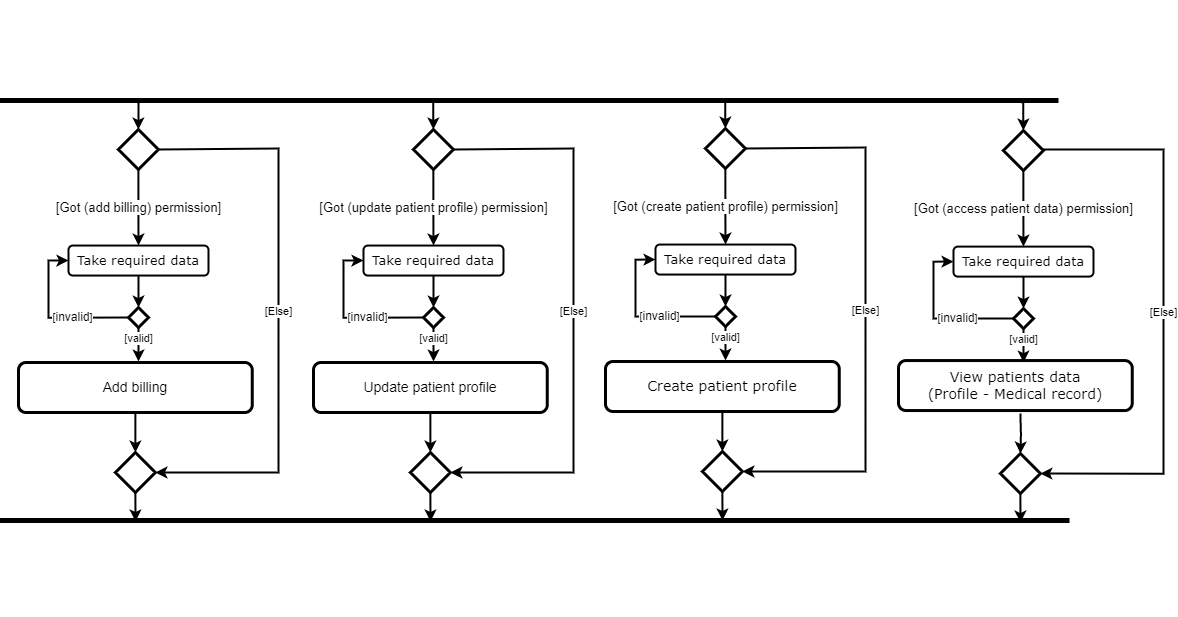
| **Risk** | **Risk**  **avoidance** | **Risk**  **management** | **Probability of occurrence** | **Level of impact** |
| --- | --- | --- | --- | --- |
| **Security breaches leading to unauthorized access or data breaches** | **Implement robust security measures, such as encryption and user authentication** | **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**

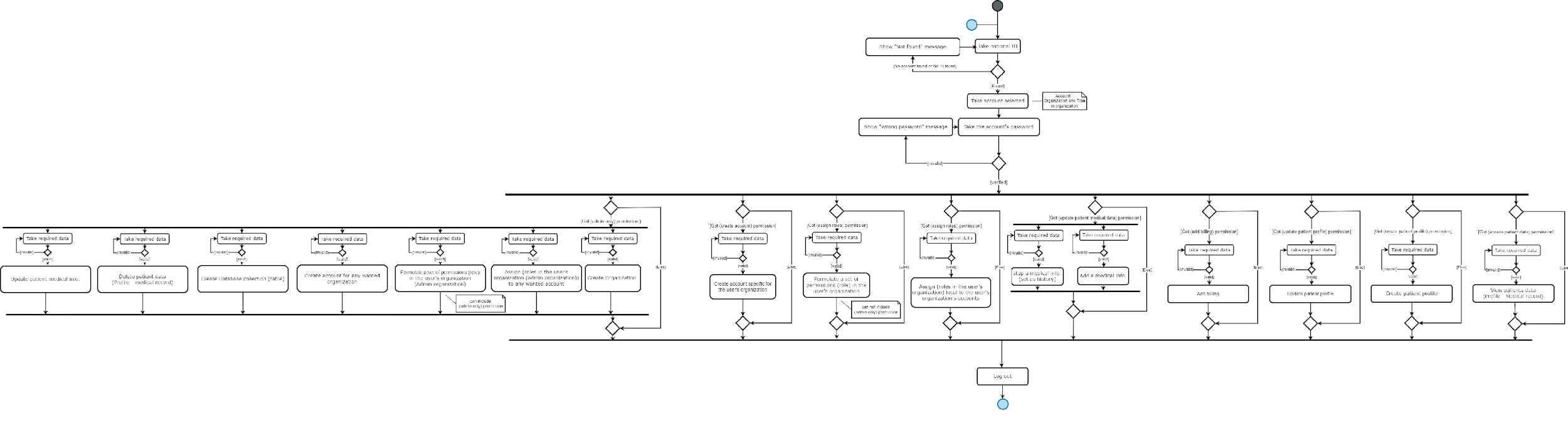
1. **Class diagram**
2. **Use case diagram**
3. **Sequence diagrams**
4. **Activity diagram**

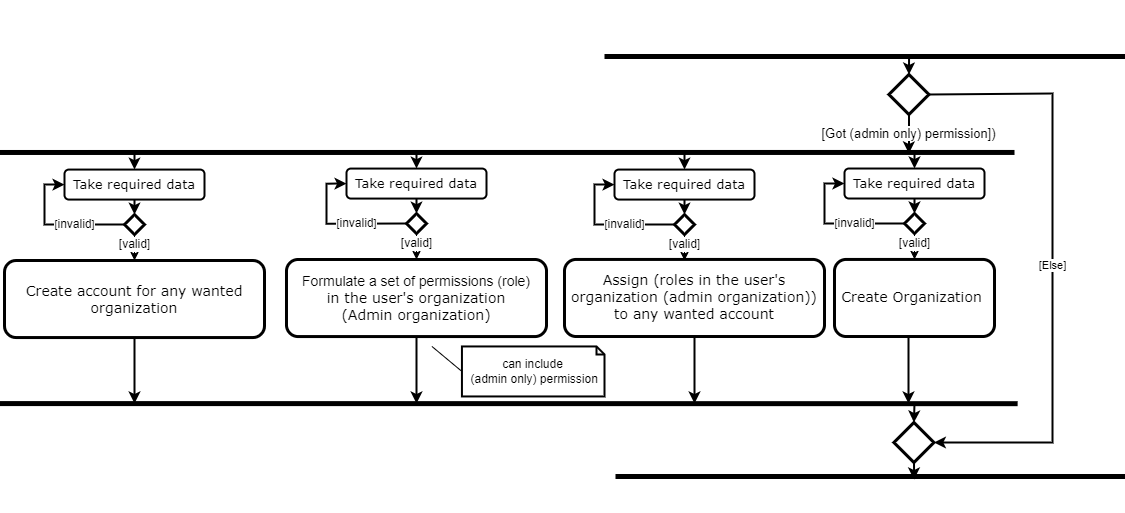




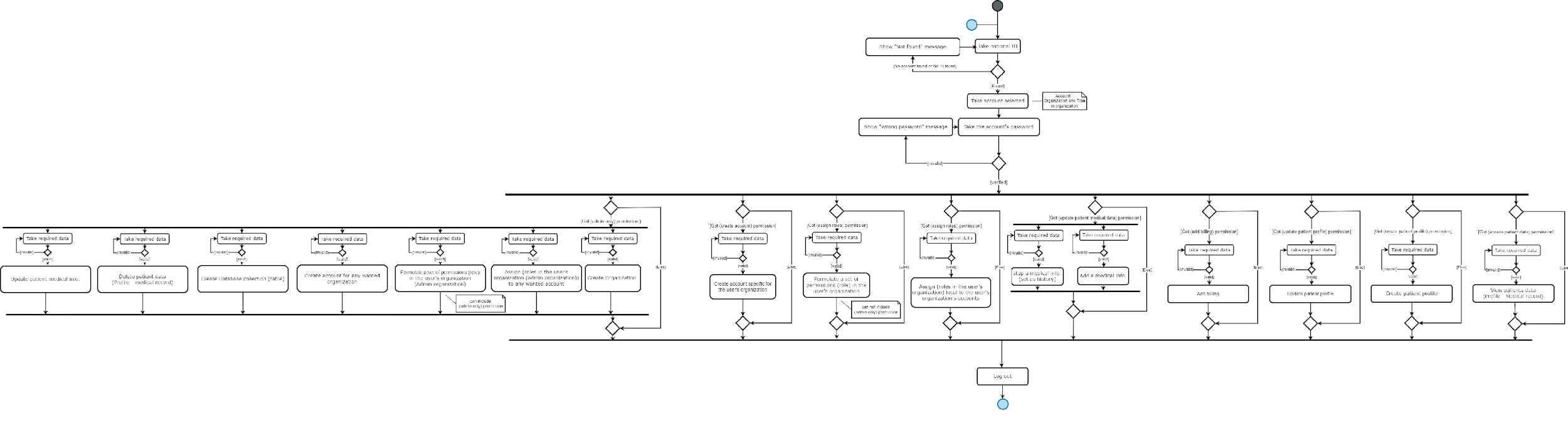


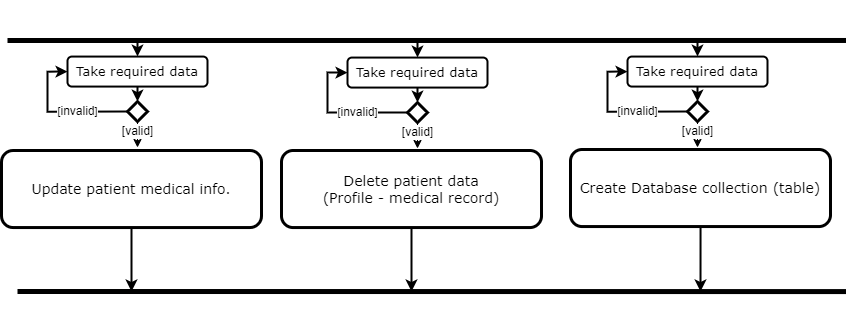






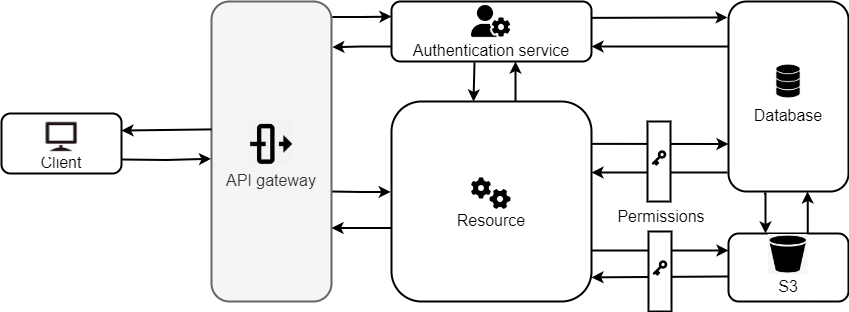






**Chapter 4: Implementation**

1. **Software architecture**



1. **Flowcharts**