Name of System: AI MedLink System

1. **Problem Analysis:**

* The lack of real-time doctor availability information causes frustration, wasted time, and potentially delayed treatment.
* Difficulty in navigating hospital facilities leads to confusion, delays, and additional stress for patients already dealing with health issues.
* The lack of effective communication with patients can result in missed appointments, improper medication usage, or non-compliance with treatment protocols, potentially harming health outcomes.
* Human error in medication dispensing is a significant risk factor in healthcare, leading to patient harm and increased hospital costs.
* Time-wasting in medical procedures leads to inefficiencies that waste valuable time, result in longer wait times, lower patient satisfaction, and reduce healthcare facilities' capacity to handle more patients.
* The lack of real-time patient feedback mechanisms limits healthcare providers' ability to identify and address issues, improve services, and enhance patient satisfaction.

**2. Analysis of the problems**

**Problem 1: Lack of Real-Time Doctor Availability Information**

**Description**: Patients often face challenges in determining if doctors are available for consultations, leading to delays in care.

**Stakeholders**: Patients, Doctor, Nurse

**Functionalities**: The AI MedLink shows real-time doctor availability, allowing patients to check if a doctor is on-call. Patients can also view schedules and book new appointments online, making it easy to arrange consultations. Additionally, patients can view their diagnosis records, reschedule appointments, and check upcoming appointments, providing comprehensive management of their healthcare needs.

Doctors can update their availability in real-time, ensuring that patients have accurate information about when they can schedule consultations. Additionally, they can record consultations and document diagnoses, streamlining patient management and ensuring that medical records are up-to-date and easily accessible.

**Non-functional requirements:**

**Performance:** The AI MedLink needs to be able to show information regarding doctor availability and appointment schedules in real-time and the system should response fast enough to display the information to the patients such that the system query wait time is not longer than 5 seconds. Besides, the system should handle the appointment booking in a sophisticated manner such that no more than one patient is allowed to book the same time slot for the same doctor which would probably cause clashes in schedules. Additionally, the system should be able to handle requests from multiple devices at the same time while providing responses simultaneously and accurately. To handle such requests on the system, when the system is being implemented, the system can be tested with high volumes of simulated users to test the system capability and performance. To further enhance the speed of system response, caching mechanisms can be applied for data that are frequently being accessed.

**Scalability:** For the system to be able to cope with high amount of concurrent users, the system should be designed for cloud-based infrastructure with auto-scaling features, for example AWS and Google Cloud.

**Portability:** To ensure the portability and accessibility of the system, the system must be able to run on both desktop and mobile platforms, which the system should support Windows, macOS, Android and iOS. Besides, the system should also be able to be opened using browsers like Google Chrome, Safari, Firefox, Microsoft Edge, etc. Multiple compatibility testing needs to be conducted on different platforms and environments as to ensure the responsiveness and smoothness of implementation when the system is put into official use.

**Compatibility:** To ensure the compatibility and seamlessness of the system, the system shall see its smooth integration with other systems in the hospital like hospital management systems. This is to prevent clashes when implementing the system to check for doctor’s availability and patients’ appointments system along with the hospital management systems which if clashes happen, fatal incidents might occur.

**Reliability:** The system should always be in good performance and the downtime of the system for maintenance should not be longer than 1 hour to ensure smoothness of operations. To ensure that this system is reliable enough, critical system failure should not occur more than once a year.

**Maintainability:** The system should be built using a commonly used architecture and language to ease the maintenance process and the maintenance source can be easily found. After detecting critical issues from the system, the repair must be done as soon as possible to prevent the issues from growing bigger and bigger.

**Availability:** The system should always be operational, which also means operating 24/7. This is to ensure that users can check for doctors’ availability and make appointments at any time of their convenience.

**Security:** To ensure that all user data are safe, the admin can set the data only to be accessed by specific high level hospital staff so that the risk of the data being leaked can be reduced. Also, when saving the user data in database, the data need to be encrypted with hashing algorithm with salt or other kind of secure encryption algorithm to protect the data, even when the data is being leaked, people can’t directly decipher the encryption. Besides, firewalls and other antivirus software can also be used to prevent hackers from hacking into the database to steal user data.

**Usability:** Considering some of the users might be old generation, the system must be easy to use and user friendly, so that users will know how to use the system just at first glance. The system shall also provide an option for users to choose their preferred language. To further improve the usability of the system, a way for the users to provide feedback is also included in the system.

**Problem 2: Difficulty in Navigating Hospital Facilities**

**Description**: Many patients find it difficult to navigate large hospital environments, not only during consultations but also throughout their visit or hospitalisation.

**Stakeholders**: patients, visitor

**Functionalities**: An AI-powered chatbot offers real-time facility and room directions, making it easier for patients and visitors to navigate the hospital efficiently. It assists in locating **Medical Departments and Clinics** such as Cardiology, Neurology, and Radiology, ensuring timely access to consultations and treatments. The chatbot also guides users to **Administrative and Support Areas**, including billing offices, registration desks, and the medical records department, simplifying administrative processes. Additionally, it provides directions to **Amenities** like the cafeteria, pharmacy, parking facilities, and prayer rooms, catering to the comfort and convenience of patients and their families throughout their visit. Lastly, the chatbot helps users locate **Patient Rooms**, ensuring they can easily find wards or private rooms for themselves or their loved ones during hospitalisation.

**Non-functional requirements:**

**Performance:** The AI powered chatbot needs to be able to respond to user queries within seconds and the system needs to be based on text-based interactions and voice-based interactions. The system needs to be powerful enough to present the directory and map as fast as possible which should also be done within seconds to provide a better performing system for users.

**Scalability:** For the system to be able to cope with high amount of concurrent users, the system should be designed for cloud-based infrastructure with auto-scaling features, for example AWS and Google Cloud.

**Portability:** To ensure the portability and accessibility of the system, the system must be able to run on both desktop and mobile platforms, which the system should support Windows, macOS, Android and iOS. Besides, the system should also be able to be opened using browsers like Google Chrome, Safari, Firefox, Microsoft Edge, etc. Multiple compatibility testing needs to be conducted on different platforms and environments as to ensure the responsiveness and smoothness of implementation when the system is put into official use.

**Compatibility:** The chatbot system can be integrated with major map applications like Google Map and Apple Map so that when users are in the hospital facility, they can use the map applications to access the system too. Besides, the chatbot system must also need to be integrated with hospital IT systems and management systems so that the chatbot system can know about the operational time, size of crowd, and availability of each department so that users can get to the location they need when they confirm that the department is available to accept new patient or customer.

**Reliability:** The system should always be in good performance and the downtime of the system for maintenance should not be longer than 1 hour to ensure smoothness of operations. To ensure that this system is reliable enough, critical system failure should not occur more than once a year.

**Maintainability:** The system should be built using a commonly used architecture and language to ease the maintenance process and the maintenance source can be easily found. After detecting critical issues from the system, the repair must be done as soon as possible to prevent the issues from growing bigger and bigger.

**Availability:** The system should always be operational, which also means operating 24/7. This is to ensure that users can check for the directories and departments at any time of their convenience.

**Security:** To protect users’ location history, the browsing and chat with the chatbot will be stored in cache memory and will be cleared after the users have done their activity. This is to prevent other users with bad intentions from tracking the history of the previous users and follow them to the location. In this system, there shouldn’t be any sensitive information from the hospital database and should only contain the necessary information needed for the chatbot to operate. This is to prevent hackers from using this chatbot as an access point to gain access to sensitive information.

**Usability:** Considering some of the users might be old generation, the system must be easy to use and user friendly, so that users will know how to use the system just at first glance. The system shall also provide an option for users to choose their preferred language. To further improve the usability of the system, a way for the users to provide feedback is also included in the system.

**Problem 3: Lack of Effective Communication with Patients**

**Description**: Patients may forget important details such as upcoming appointments or necessary precautions, especially after consultations.

**Stakeholders**: patients, doctors

**Functionalities**: The AI Medical System enhances patient care by automatically sending personalized emails with reminders about necessary precautions, upcoming appointments, personalized tips for improving health, and other essential information. These tailored messages ensure that patients stay informed and empowered in managing their healthcare.

It also empowers patients to upload images of their meals for analysis, generating detailed reports and tailored dietary recommendations, including any necessary restrictions. Additionally, patients can upload images of their exercise posture, allowing the AI to analyse and recommend proper posture, helping to prevent injuries and optimise workout effectiveness.

**Non-functional requirements:**

**Performance:** When the users successfully make an appointment, the email must be sent by the system within one minute to notify the users. This is to ensure that the users know that their appointment is being recorded and is in the system. For users’ personalized emails for health updates and reminders, the AI system needs to be appropriate and precise so that the users won’t miss any notifications and updates from the hospital about their appointments. For users to upload their daily meals for analysis, the system needs to support uploads of photos up to 100mb per users and the analysis of the meals should be done as fast as possible that the AI can complete the analysis within 30 seconds.

**Scalability:** For the system to be able to cope with high amount of concurrent users, the system should be designed for cloud-based infrastructure with auto-scaling features, for example AWS and Google Cloud.

**Portability:** To ensure the portability and accessibility of the system, the system must be able to run on both desktop and mobile platforms, which the system should support Windows, macOS, Android and iOS. Besides, the system should also be able to be opened using browsers like Google Chrome, Safari, Firefox, Microsoft Edge, etc. Multiple compatibility testing needs to be conducted on different platforms and environments as to ensure the responsiveness and smoothness of implementation when the system is put into official use.

**Compatibility:** Compatibility wise, the photo format for the system must be commonly and widely used like JPEG, PNG, JPG, HEIC, HEIF and so on. This is to ensure that the photos submitted by the users are compatible with the system and can be accessed by the system.

**Reliability:** The system should always be in good performance and the downtime of the system for maintenance should not be longer than 1 hour to ensure smoothness of operations. To ensure that this system is reliable enough, critical system failure should not occur more than once a year.

**Maintainability:** The system should be built using a commonly used architecture and language to ease the maintenance process and the maintenance source can be easily found. After detecting critical issues from the system, the repair must be done as soon as possible to prevent the issues from growing bigger and bigger.

**Availability:** The system should always be operational, which also means operating 24/7. This is to ensure that users can receive their personalized reports and upload their images for analysis at any time of their convenience.

**Security:** Since personalized users report need to access sensitive data from users, the data needs to be protected and encrypted in a safe and secure manner to prevent hackers from stealing the data. The photos uploaded by users also need to be store in a server that are safe from outer harm like hackers.

**Usability:** Considering some of the users might be old generation, the system must be easy to use and user friendly, so that users will know how to use the system just at first glance. The system shall also provide an option for users to choose their preferred language. To further improve the usability of the system, a way for the users to provide feedback is also included in the system.

**Problem 4: Human Error in Medication Dispensing**

**Description**: Dispensing the wrong medication or incorrect dosages can have serious consequences for patients' health, especially during times of high patient volume when errors are more likely to occur due to increased workload and stress.

**Stakeholders**: patients, pharmacist

**Functionalities**: The system uses computer vision to automatically count and dispense the medication according to prescribed quantity and type, with the help of human to verify the medications for double confirmation, ensuring that patients receive the correct medication in the right amount.

**Non-functional requirements:**

**Performance:** To prevent patients from waiting too long in the dispensary, the computer vision robot need to read the prescription list in 10 seconds and dispense the medication based on correct quantity and type. Robotic arms with high mobility can be applied here to speed up the dispensing process.

**Scalability:** For the system to be able to cope with high amount of concurrent users, the system should be designed for cloud-based infrastructure with auto-scaling features, for example AWS and Google Cloud.

**Portability:** To ensure the portability and accessibility of the system, the system must be able to run on both desktop and mobile platforms, which the system should support Windows, macOS, Android and iOS. Besides, the system should also be able to be opened using browsers like Google Chrome, Safari, Firefox, Microsoft Edge, etc. Multiple compatibility testing needs to be conducted on different platforms and environments as to ensure the responsiveness and smoothness of implementation when the system is put into official use.

**Compatibility:** The system needs to be able to scan the barcode and identify packaging of medicines with a feature to count the amount of medicines so that the system can perform the dispensing task.

**Reliability:** The system should always be in good performance and the downtime of the system for maintenance should not be longer than 1 hour to ensure smoothness of operations. To ensure that this system is reliable enough, critical system failure should not occur more than once a year.

**Maintainability:** The system should be built using a commonly used architecture and language to ease the maintenance process and the maintenance source can be easily found. After detecting critical issues from the system, the repair must be done as soon as possible to prevent the issues from growing bigger and bigger.

**Availability:** The system should always be operational, which also means operating 24/7. This is to ensure that the medicine dispensing robot is always ready to dispense medicine in case of emergency time.

**Security:** After the robot read the prescription and done performing the dispensing task, the memory about the prescription needs to be cleared to prevent any leak of patient’s disease information. Besides, to ensure medical security, the medicine dispensing robot needs to be trained with high accurate training and testing data consists of high quality photos of various medicine and their respective packaging to ensure a high accuracy in dispensing. This is to reduce the error rate of the robot.

**Usability:** Considering some of the users might be old generation, the system must be easy to use and user friendly, so that users will know how to use the system just at first glance. The system shall also provide an option for users to choose their preferred language. To further improve the usability of the system, a way for the users to provide feedback is also included in the system.

**Problem 5: Time-Wasting in Medical Procedures**

**Description**: Patients often experience unnecessary delays due to manual processes and administrative tasks.

**Stakeholders**: patients, doctor, nurse

**Functionalities**: The AI Medical System streamlines processes, reducing time spent on administrative tasks and enabling quicker patient care. It also includes a conversational AI feature that analyses patient interactions, offering general recommendations based on the conversation and checking if the user would like to book an appointment. Additionally, the system allows patients to book medication refill appointments, ensuring that their ongoing treatments are managed efficiently.

**Non-functional requirements:**

**Performance:** The system should respond within 30 seconds to handle patients’ appointment bookings and medication refill requests. This is because this system emphasizes on time-saving and speed, so the system should also handle general recommendations and interactions with patients in real time speed.

**Scalability:** For the system to be able to cope with high amount of concurrent users, the system should be designed for cloud-based infrastructure with auto-scaling features, for example AWS and Google Cloud.

**Portability:** To ensure the portability and accessibility of the system, the system must be able to run on both desktop and mobile platforms, which the system should support Windows, macOS, Android and iOS. Besides, the system should also be able to be opened using browsers like Google Chrome, Safari, Firefox, Microsoft Edge, etc. Multiple compatibility testing needs to be conducted on different platforms and environments as to ensure the responsiveness and smoothness of implementation when the system is put into official use.

**Compatibility:** This system needs to be able to communicate and share information with other systems in the hospital like the appointment booking system and patient profile so that patients can conveniently apply for their medicine refill.

**Reliability:** The system should always be in good performance and the downtime of the system for maintenance should not be longer than 1 hour to ensure smoothness of operations. To ensure that this system is reliable enough, critical system failure should not occur more than once a year.

**Maintainability:** The system should be built using a commonly used architecture and language to ease the maintenance process and the maintenance source can be easily found. After detecting critical issues from the system, the repair must be done as soon as possible to prevent the issues from growing bigger and bigger.

**Availability:** The system should always be operational, which also means operating 24/7. This is to ensure that users can perform their booking requests and medicine refill request at any time of their convenience.

**Security:** Since this system needs to communicate with other systems to gain knowledge and information about patients, hence the data storage and security for this system need to be extremely meticulous. Extra security care like high-end encryption and firewall can be invested to safeguard the data.

**Usability:** Considering some of the users might be old generation, the system must be easy to use and user friendly, so that users will know how to use the system just at first glance. The system shall also provide an option for users to choose their preferred language. To further improve the usability of the system, a way for the users to provide feedback is also included in the system.

**Problem 6: Lack of Real-Time Patient Feedback Mechanisms**

**Description:** Hospitals and medical staff may not receive timely feedback from patients regarding the quality of care, facilities, or overall experience.

**Stakeholder:** patients, doctor, nurse

**Functionalities**: An AI-powered chatbot offers real-time facility and room directions, making it easier for patients and visitors to navigate the hospital efficiently. It assists in locating **Medical Departments and Clinics** such as Cardiology, Neurology, and Radiology, ensuring timely access to consultations and treatments. The chatbot also guides users to **Administrative and Support Areas**, including billing offices, registration desks, and the medical records department, simplifying administrative processes. Additionally, it provides directions to **Amenities** like the cafeteria, pharmacy, parking facilities, and prayer rooms, catering to the comfort and convenience of patients and their families throughout their visit. It further supports users by enabling them to **mark their parking spot and retrieve its location later**, eliminating the hassle of searching for their vehicle in large hospital parking areas. Lastly, the chatbot helps users locate **Patient Rooms**, ensuring they can easily find wards or private rooms for themselves or their loved ones during hospitalisation.

The chatbot checks the availability and operational status of wheelchair-accessible paths, escalators, or elevators in real time. It then provides optimised navigation routes based on the user's mobility needs, ensuring a seamless and accessible experience for all visitors.

**Non-functional requirements:**

**Performance:** The AI chatbot needs to be fast enough to provide directions, safe parking spot, and help the users to provide feedback submissions within 10 seconds wait time.

**Scalability:** For the system to be able to cope with high amount of concurrent users, the system should be designed for cloud-based infrastructure with auto-scaling features, for example AWS and Google Cloud.

**Portability:** To ensure the portability and accessibility of the system, the system must be able to run on both desktop and mobile platforms, which the system should support Windows, macOS, Android and iOS. Besides, the system should also be able to be opened using browsers like Google Chrome, Safari, Firefox, Microsoft Edge, etc. Multiple compatibility testing needs to be conducted on different platforms and environments as to ensure the responsiveness and smoothness of implementation when the system is put into official use.

**Compatibility:** The chatbot can be implemented with third party map applications like Google Map and Apple map so that users can also use their apps to navigate in the hospital. The chatbot needs to be able to speak multiple languages to make sure that every legitimate user can access and interact with the system.

**Reliability:** The system should always be in good performance and the downtime of the system for maintenance should not be longer than 1 hour to ensure smoothness of operations. To ensure that this system is reliable enough, critical system failure should not occur more than once a year.

**Maintainability:** The system should be built using a commonly used architecture and language to ease the maintenance process and the maintenance source can be easily found. After detecting critical issues from the system, the repair must be done as soon as possible to prevent the issues from growing bigger and bigger.

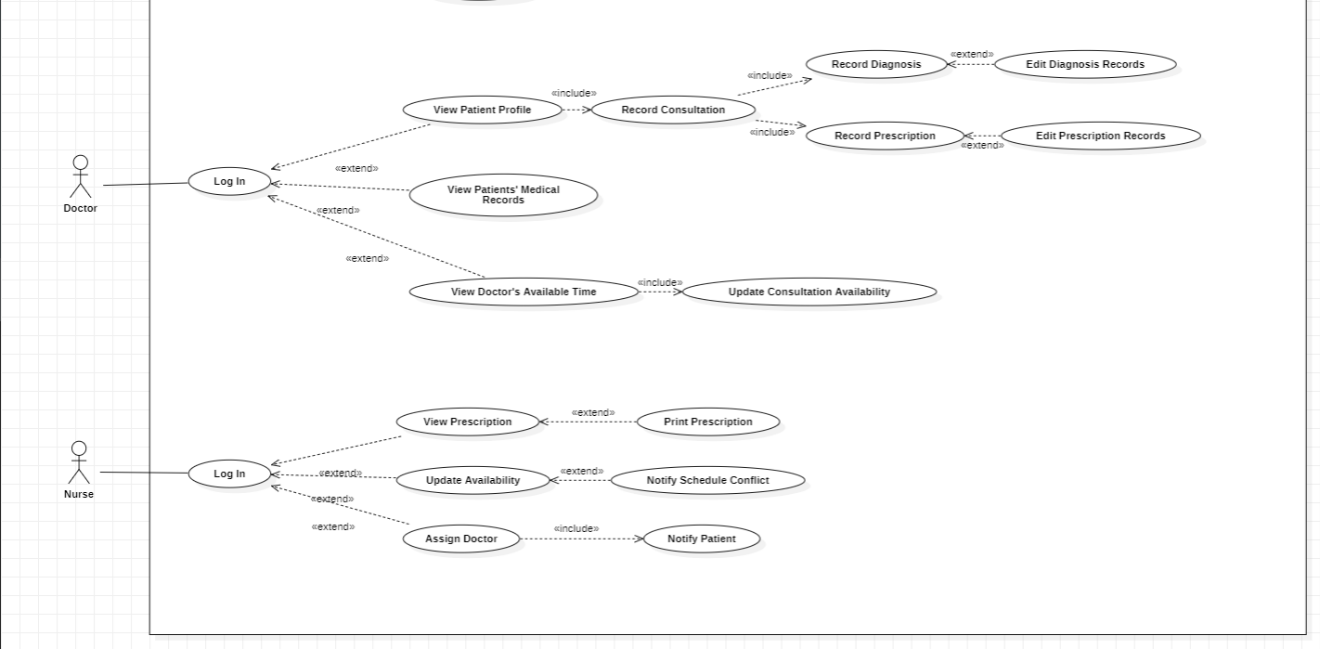
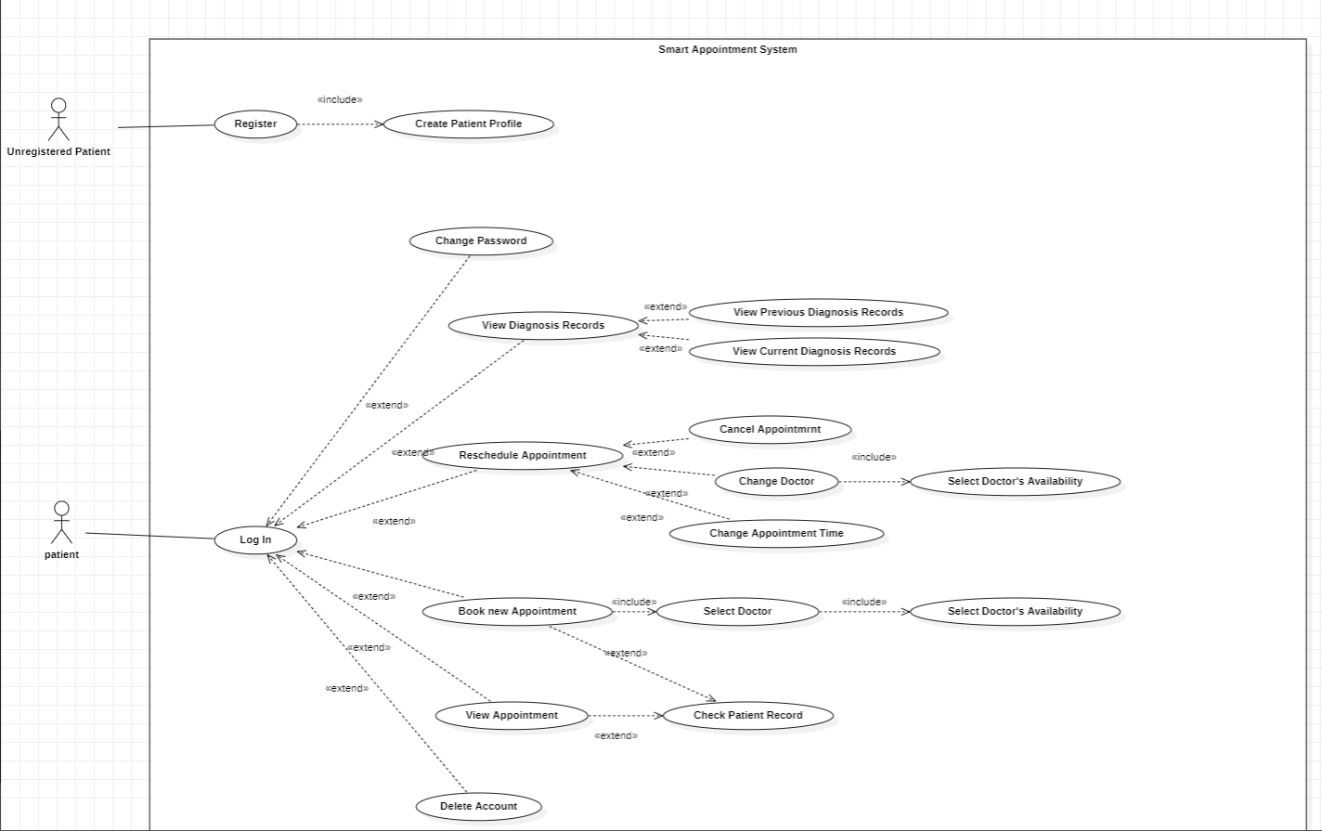
**Availability:** The system should always be operational, which also means operating 24/7. This is to ensure that users can check for the directories and departments at any time of their convenience.

**Security:** The chat history between users and chatbot will be discarded immediately to provide privacy protection.

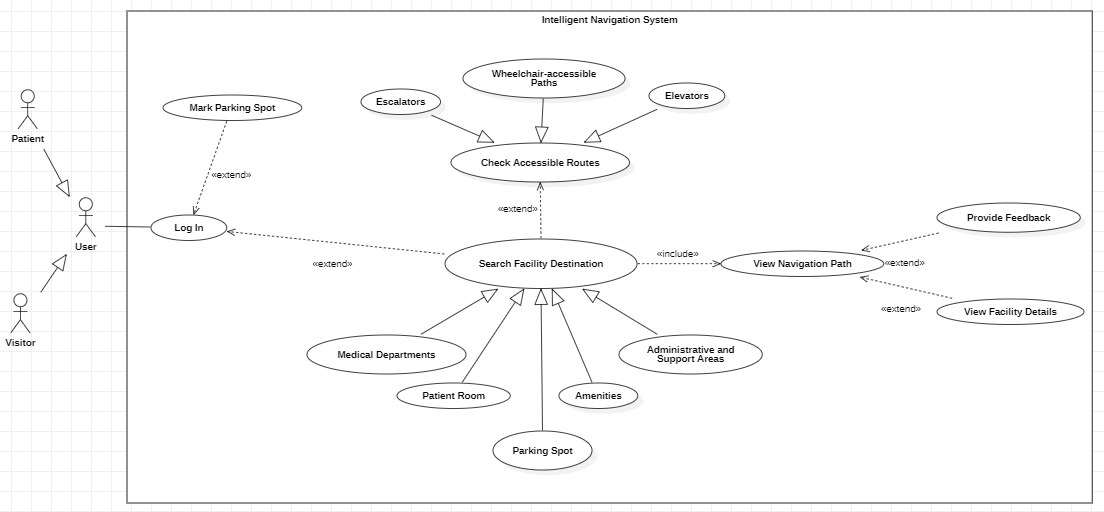
**Usability:** Considering some of the users might be old generation, the system must be easy to use and user friendly, so that users will know how to use the system just at first glance. The system shall also provide an option for users to choose their preferred language. To further improve the usability of the system, a way for the users to provide feedback is also included in the system.

**Use Case Diagram**

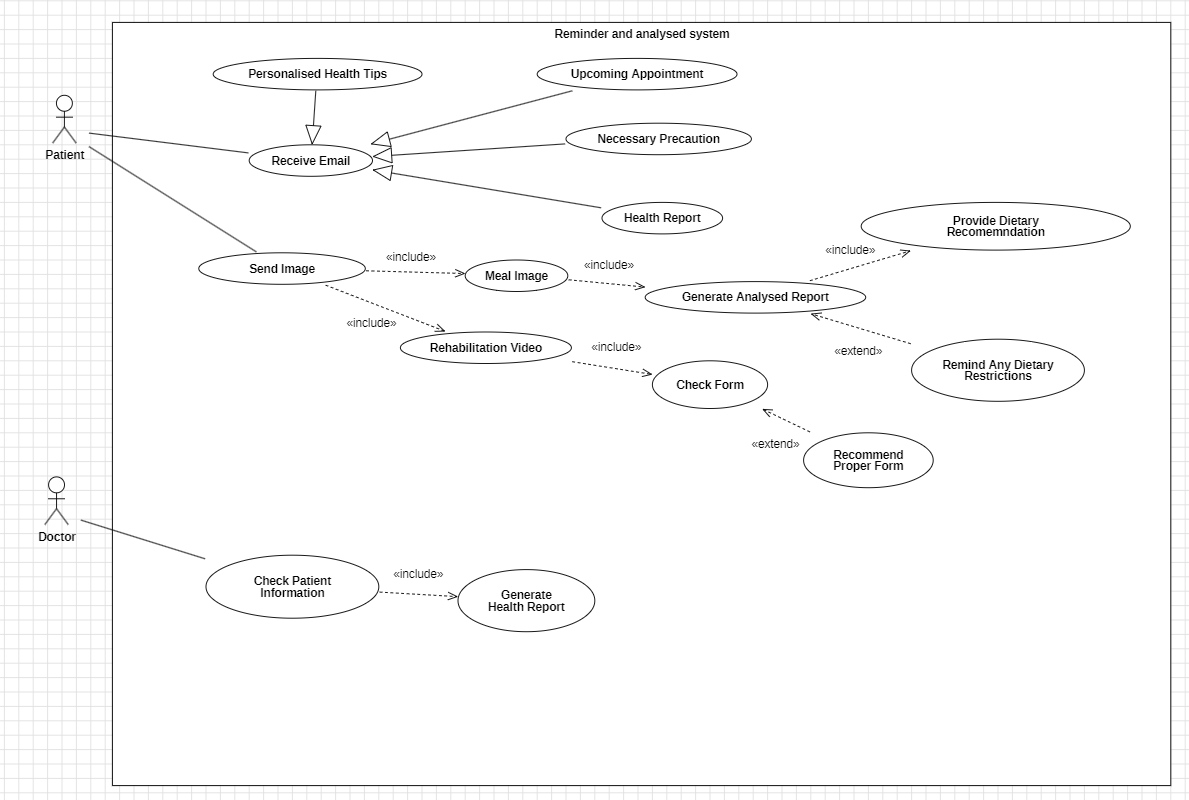
[Problem 1] Smart Appointment System



[Problem 2] Intelligent Navigation System



[Problem 3] Reminder and Analysis System

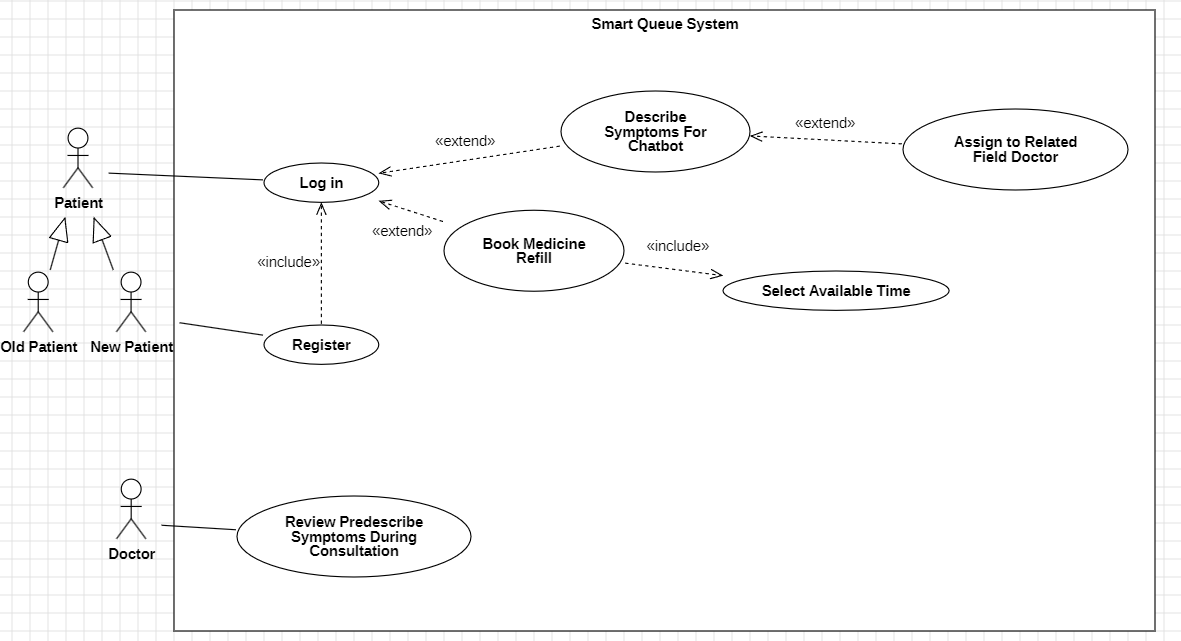


[Problem 4] Computer Vision Supervised Medicine Dispensing System

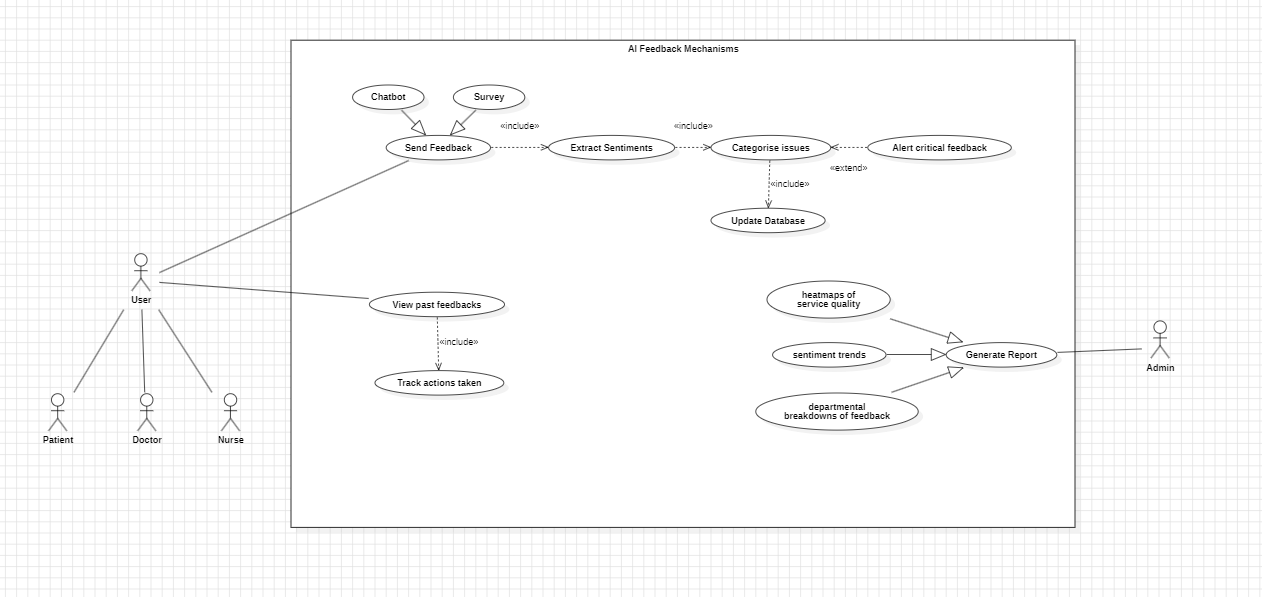
A diagram of a company

Description automatically generated

[Problem 5] Smart Queue System

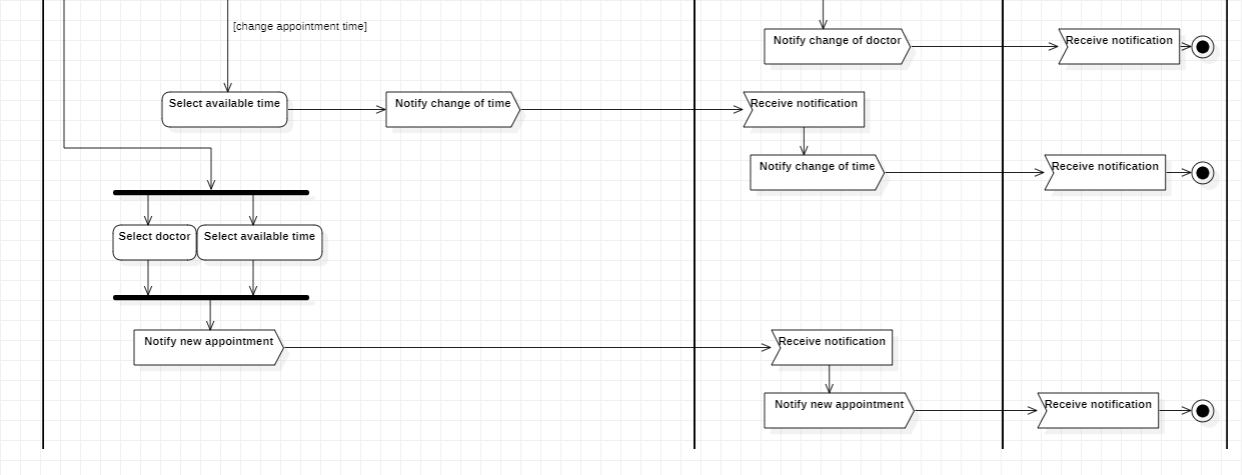
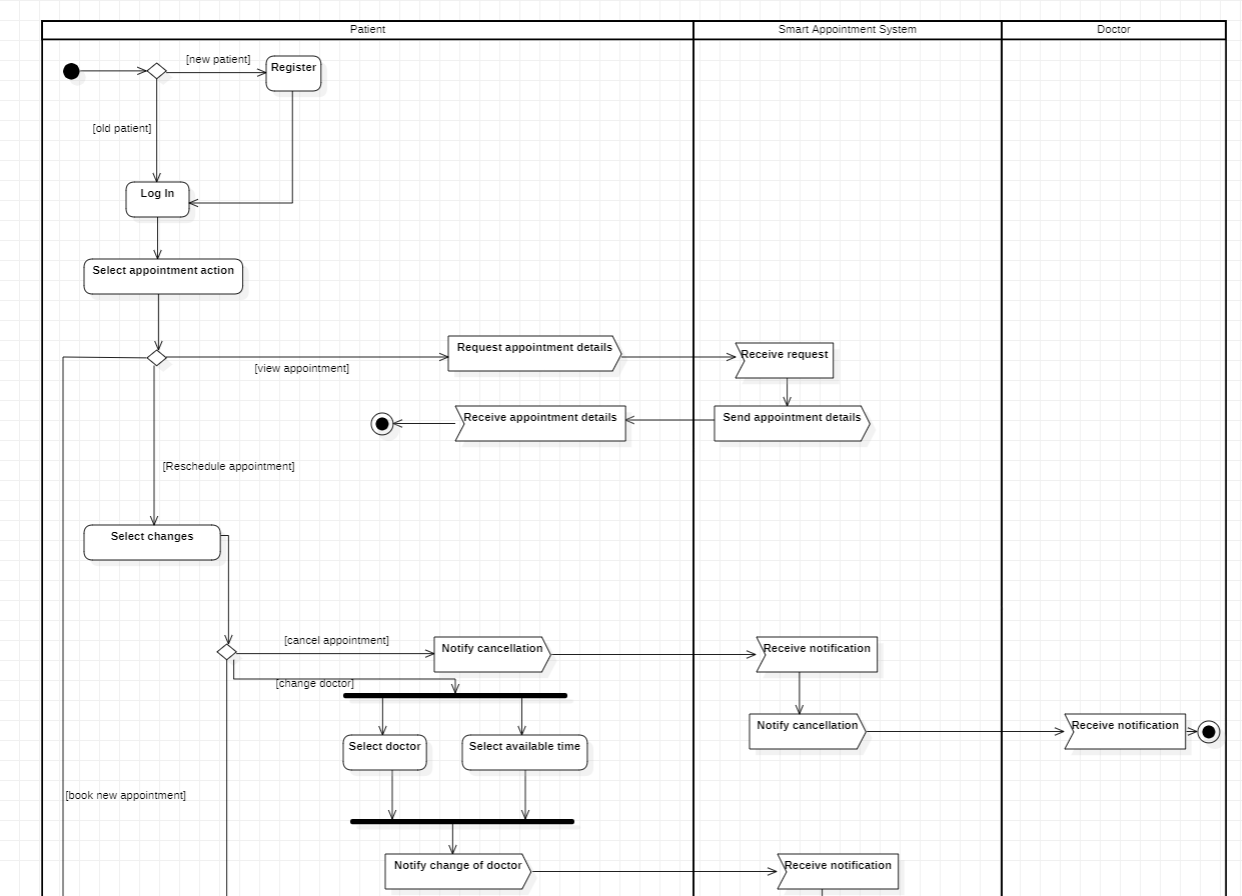


[Problem6] AI Feedback Systems

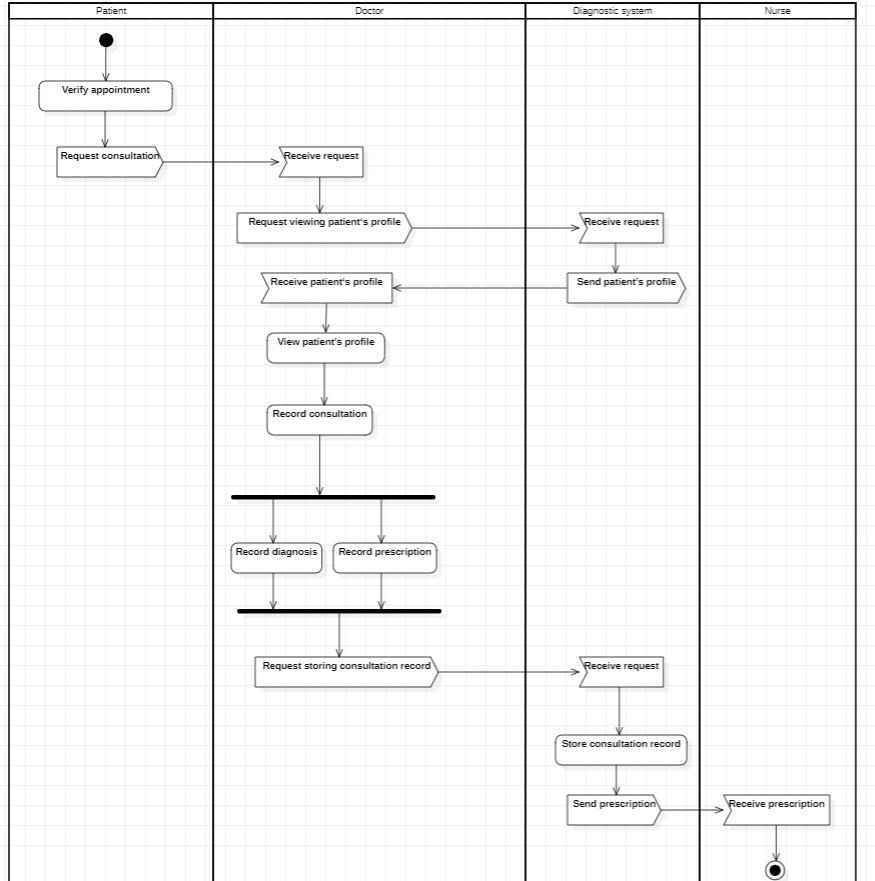


**Activity Diagram**

[Problem 1] Smart Appointment System



[Problem 2] Diagnostic system

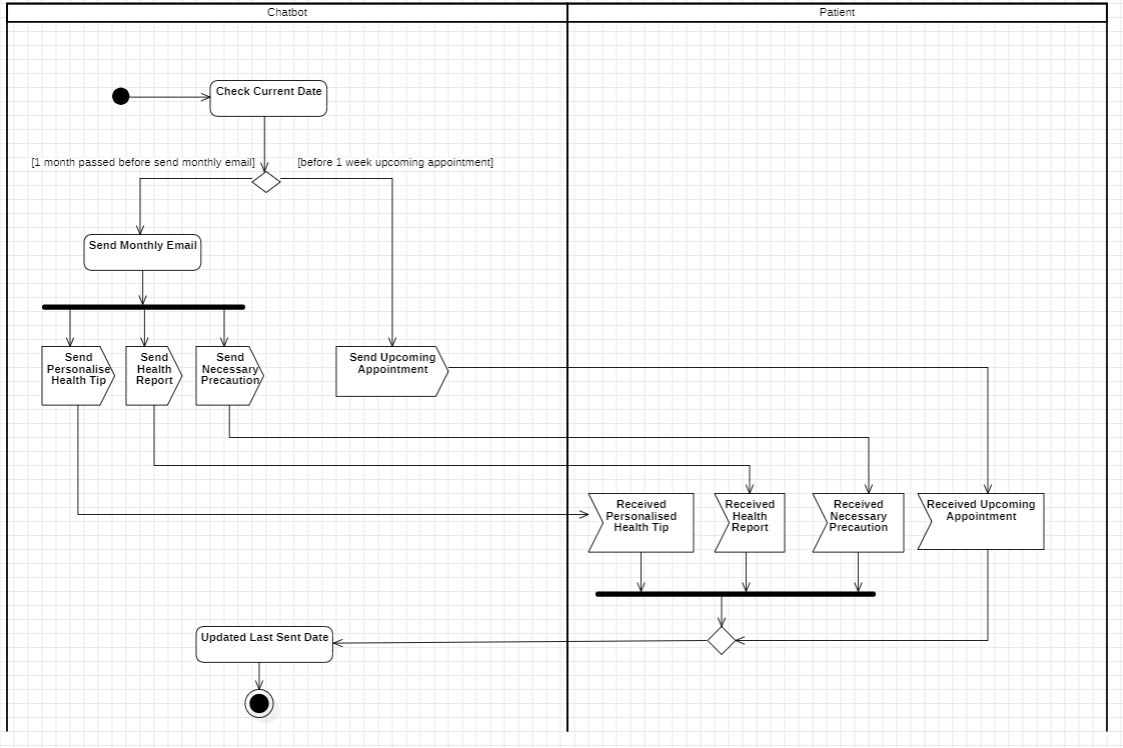


[Problem 2] Intelligent Navigation System

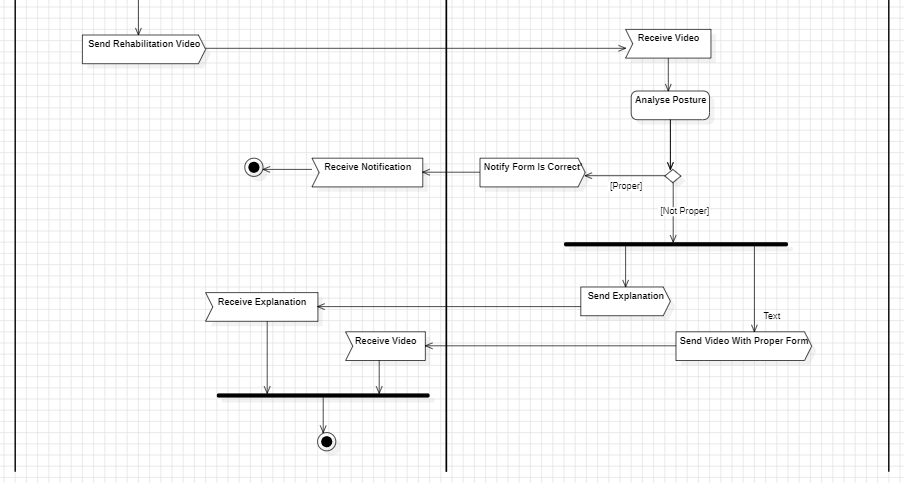
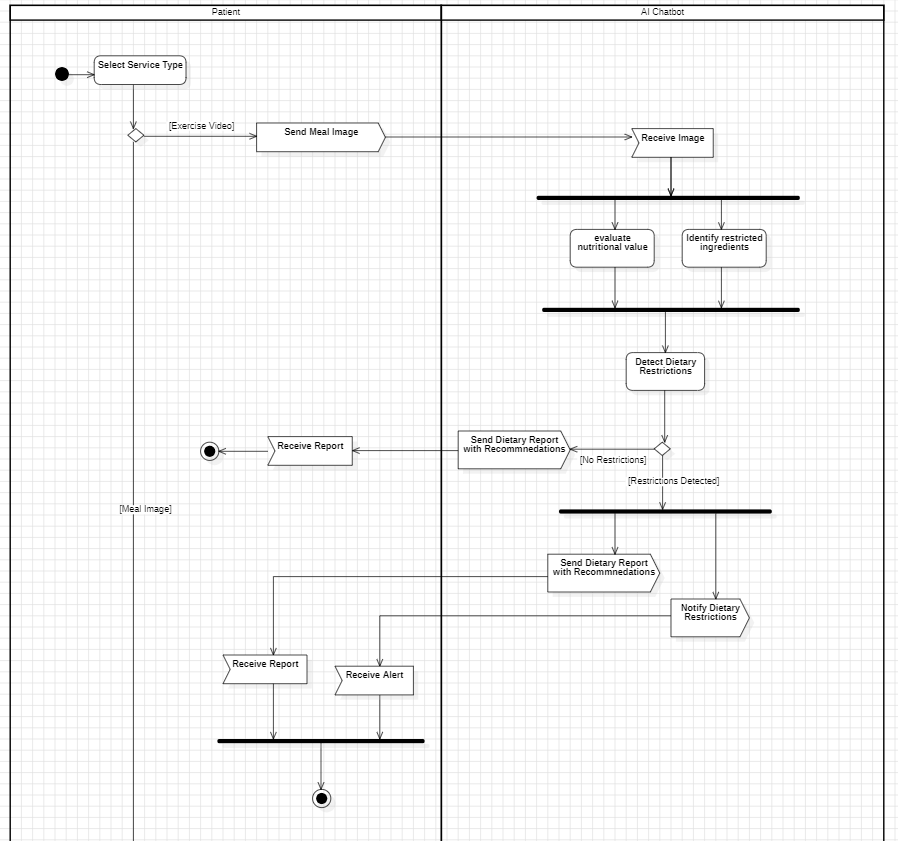
A diagram of a software application

Description automatically generated

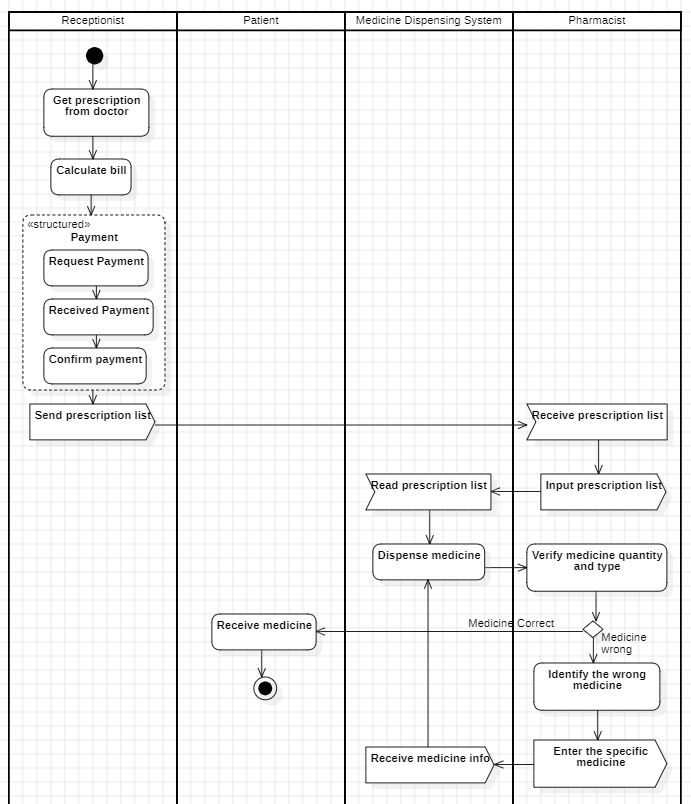
[Problem 3] Personalised AI Generated Email



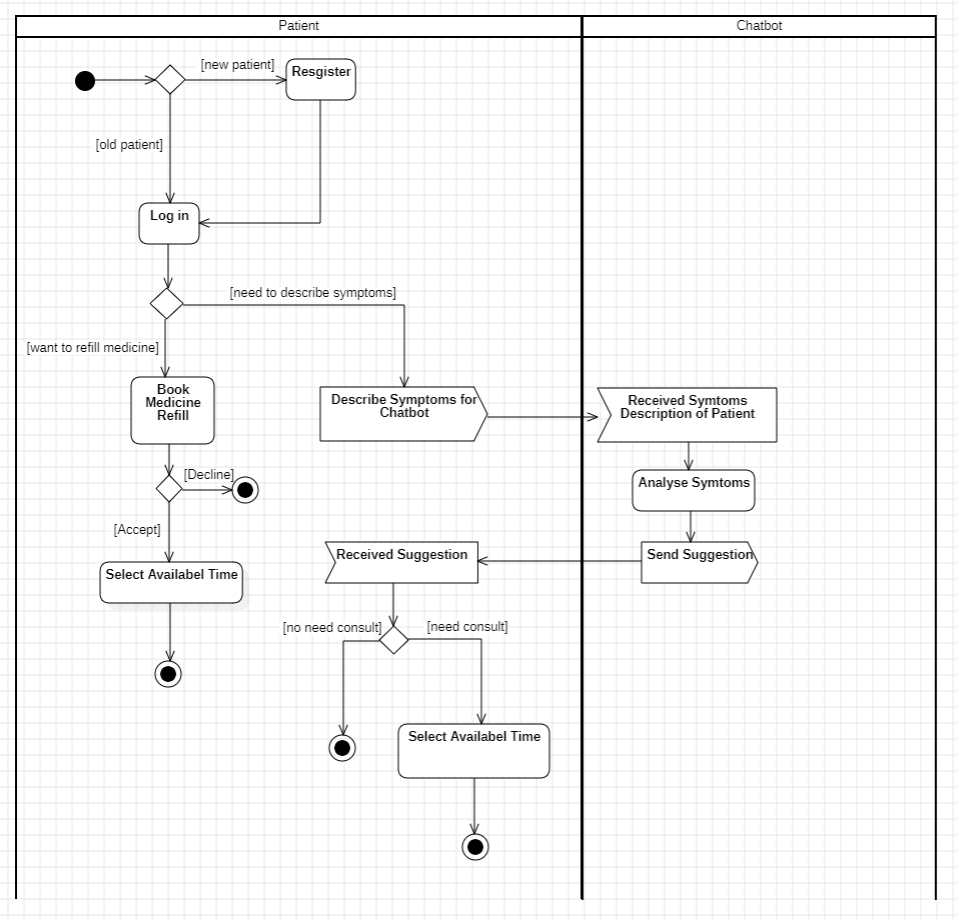
[Problem 3] AI Analysis system



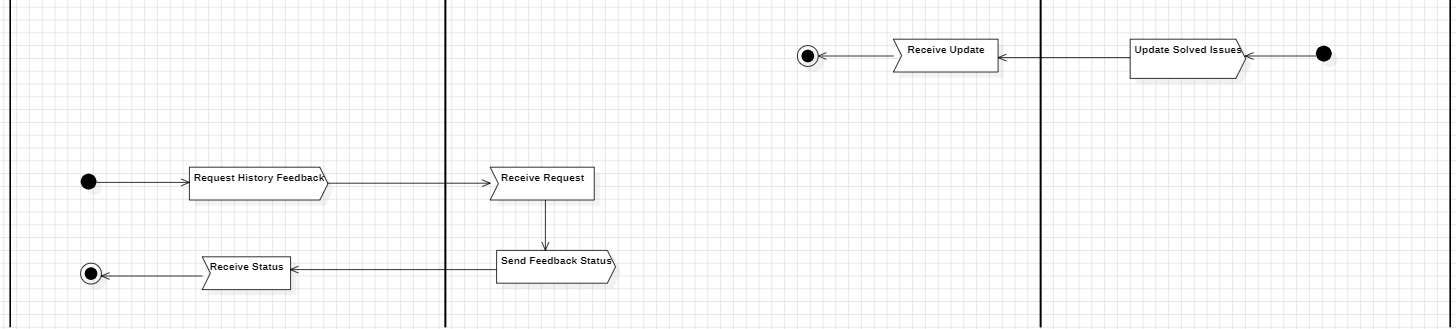
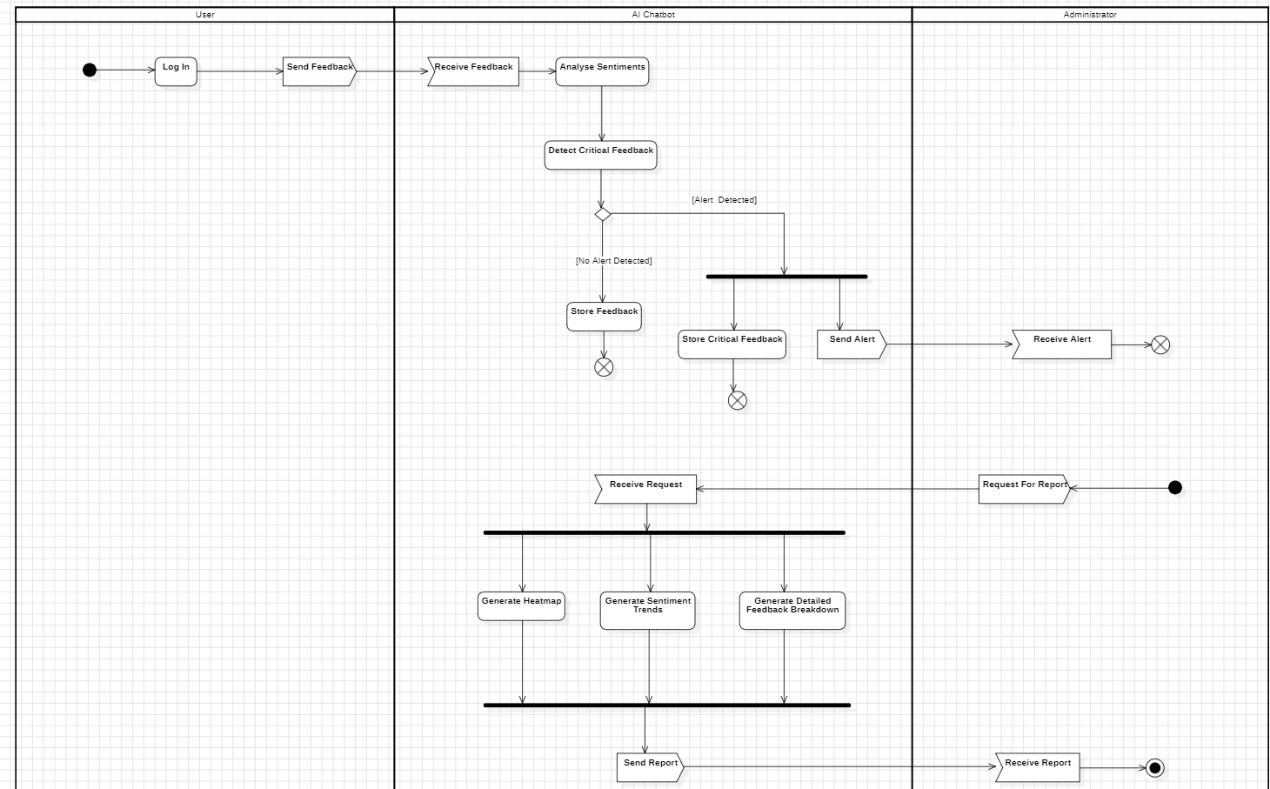
[Problem 4] Computer Vision Supervised Medicine Dispensing System



[Problem 5] Smart Queue System

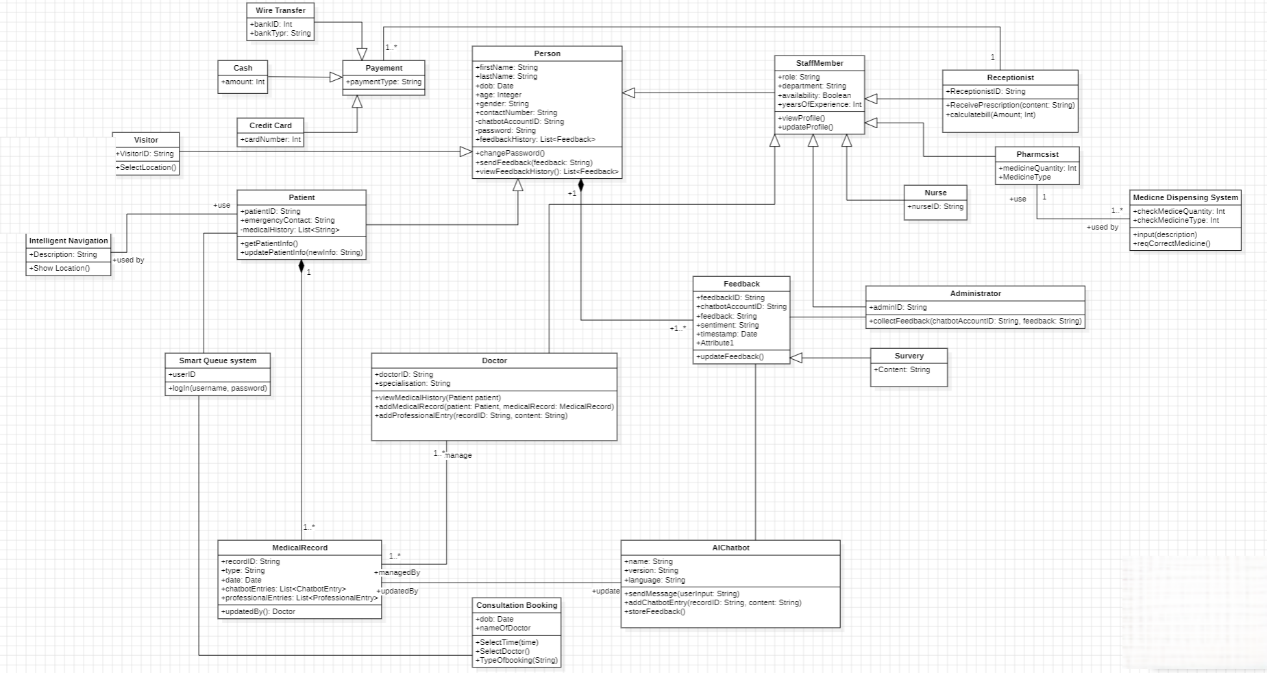


[Problem 6] AI Feedback Mechanisms

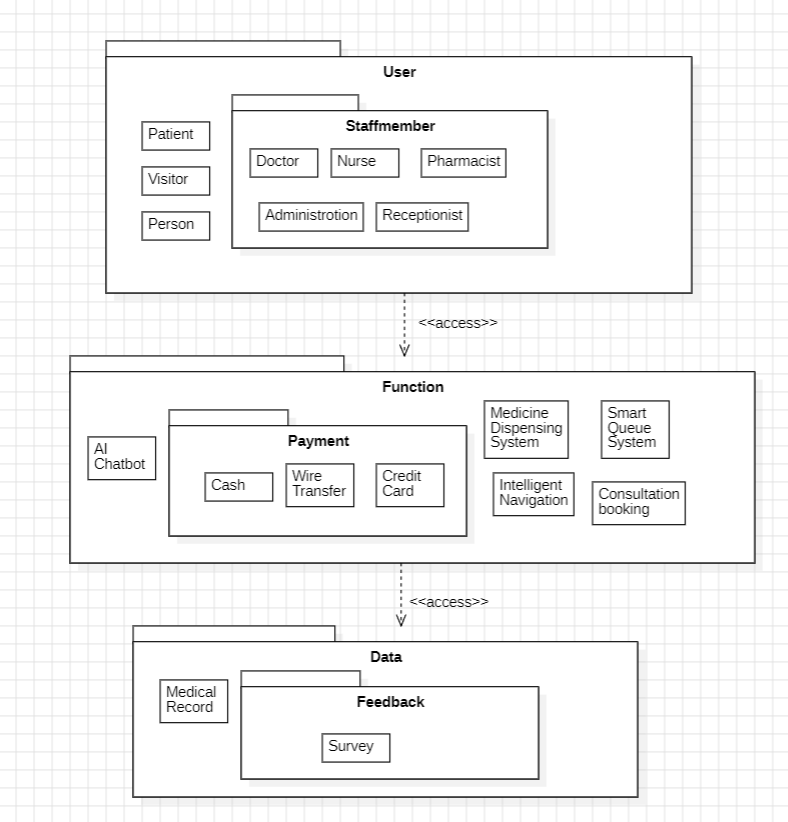


**Class & Package diagram**

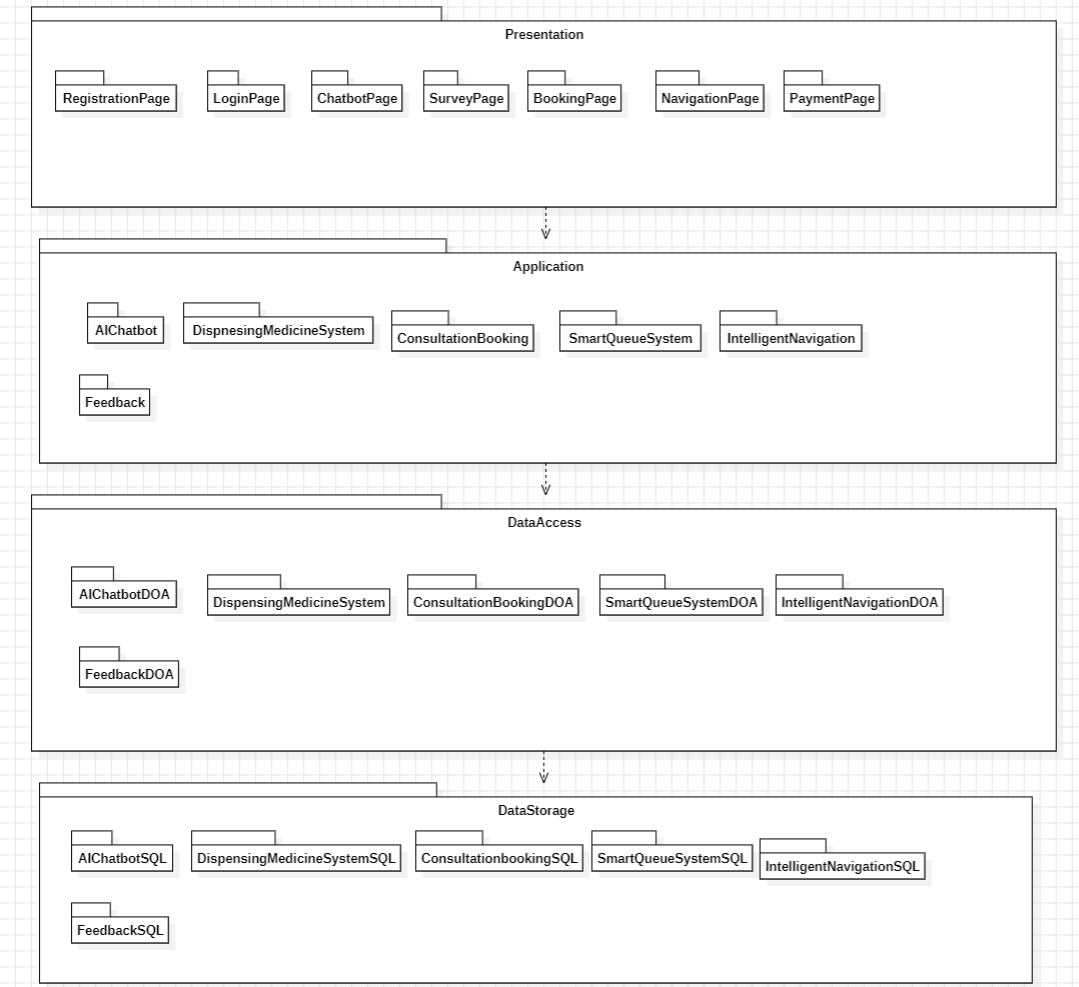
Level 1



Level 2

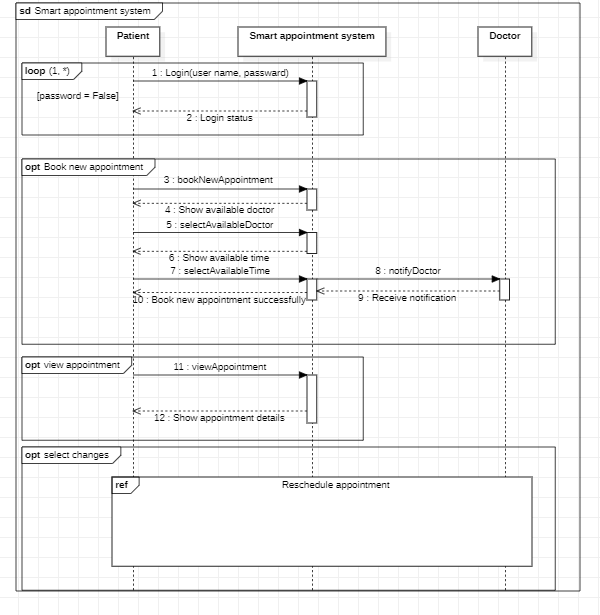


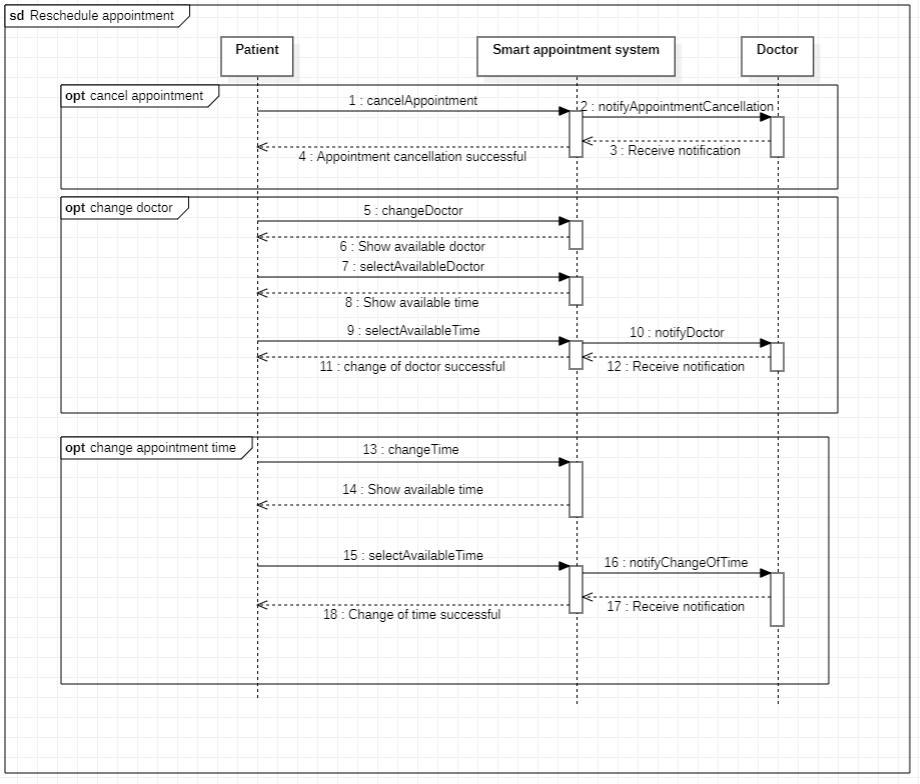
Level 3



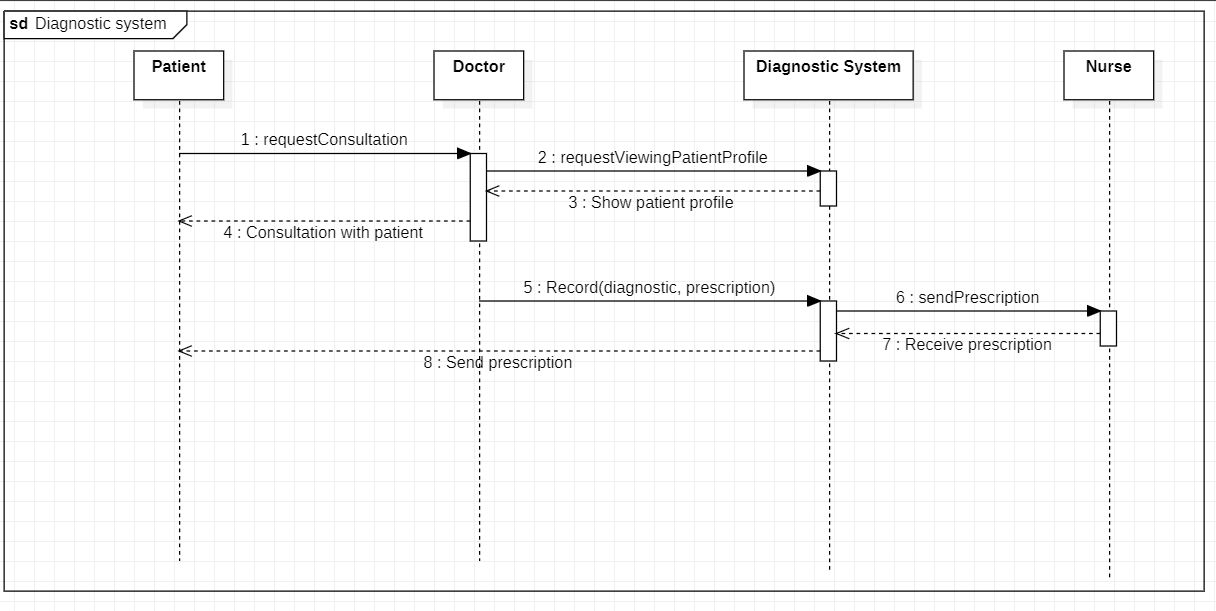
**Sequence Diagram**

[Problem 1] Smart Appointment System





[Problem 1] Diagnostic System



[Problem 2] Intelligent Navigation System

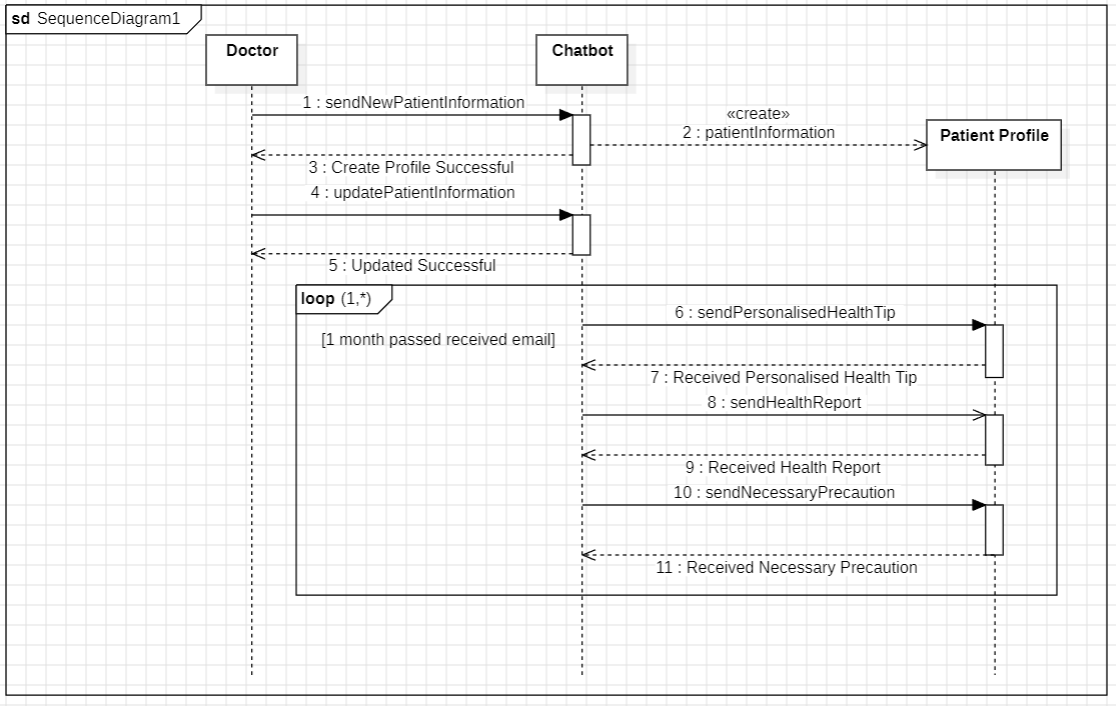
A diagram of a system

Description automatically generated

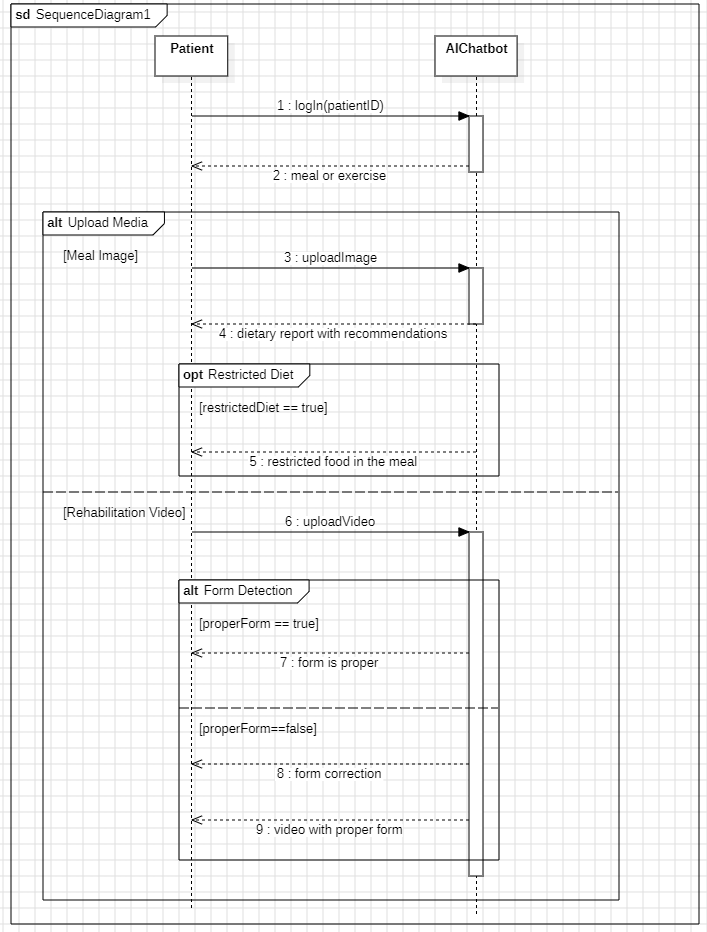
A diagram of a user

Description automatically generated

[Problem 3] Personalised AI-Generated Email



[Problem 3] AI Analysis system

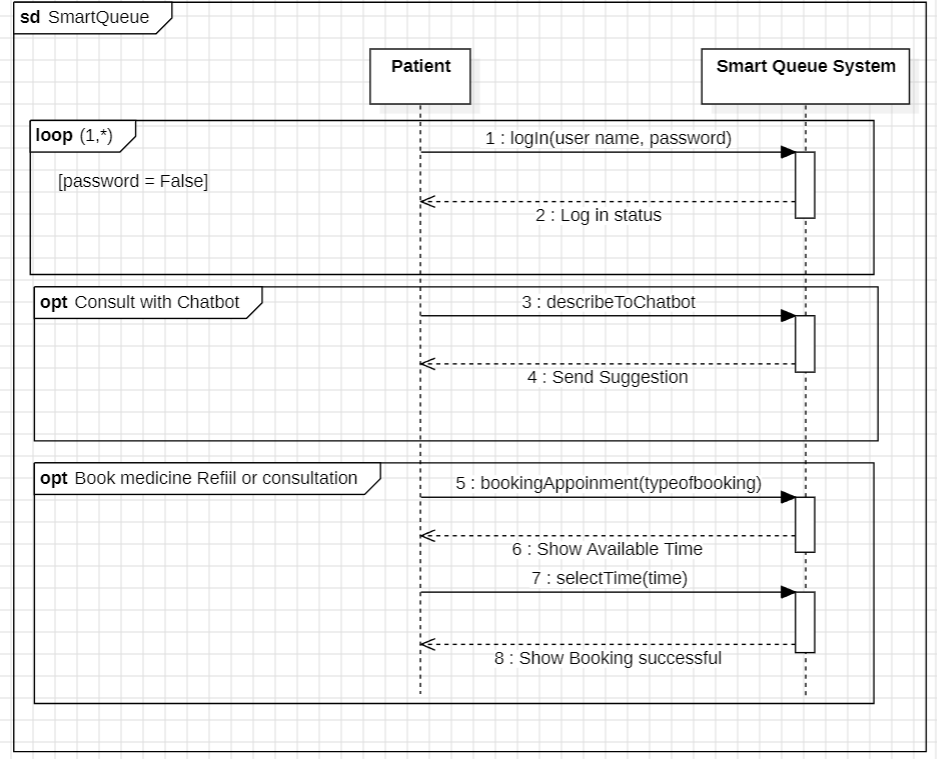


[Problem 4] Computer Vision Supervised Medicine Dispensing System

A diagram of a program

Description automatically generated with medium confidence

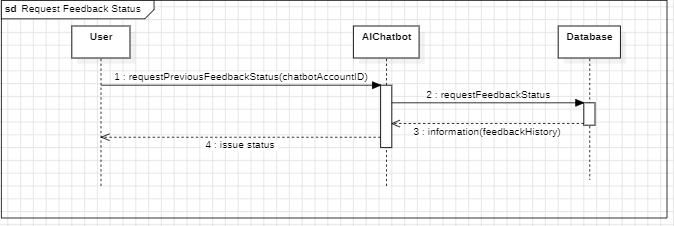
[Problem 5] Smart Queue System



[Problem 6] AI Feedback Mechanisms A diagram of a project

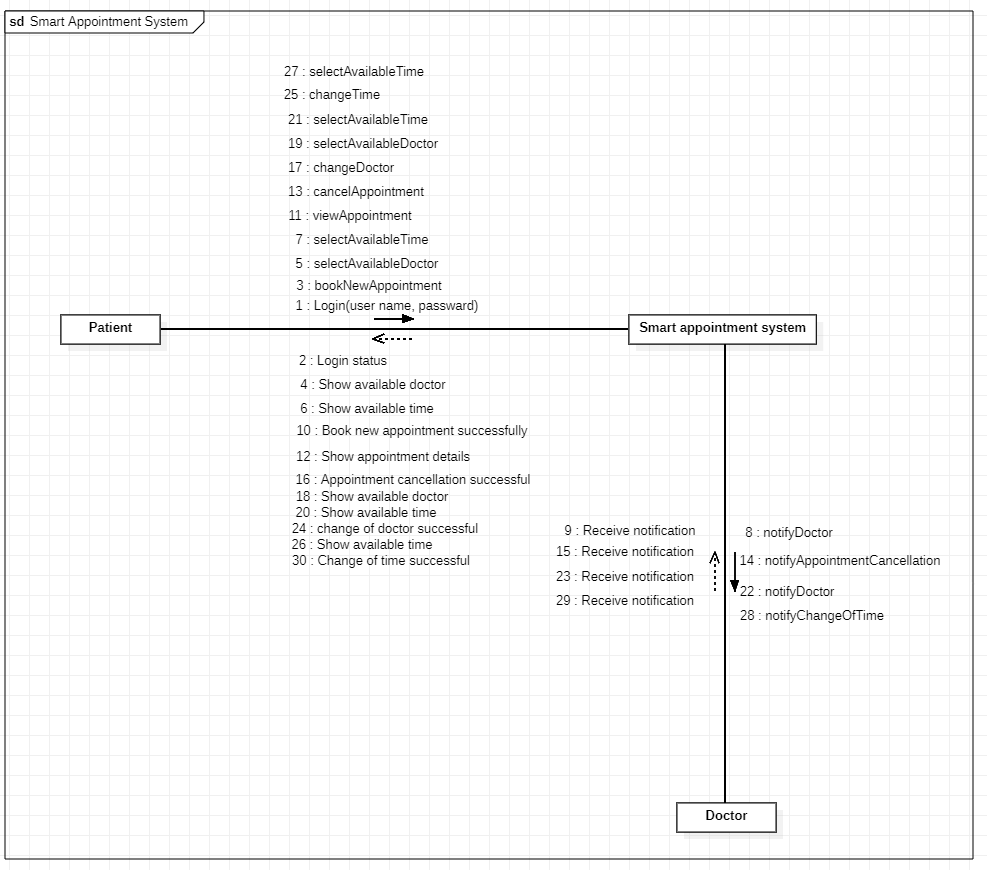
Description automatically generatedA diagram of a process

Description automatically generated

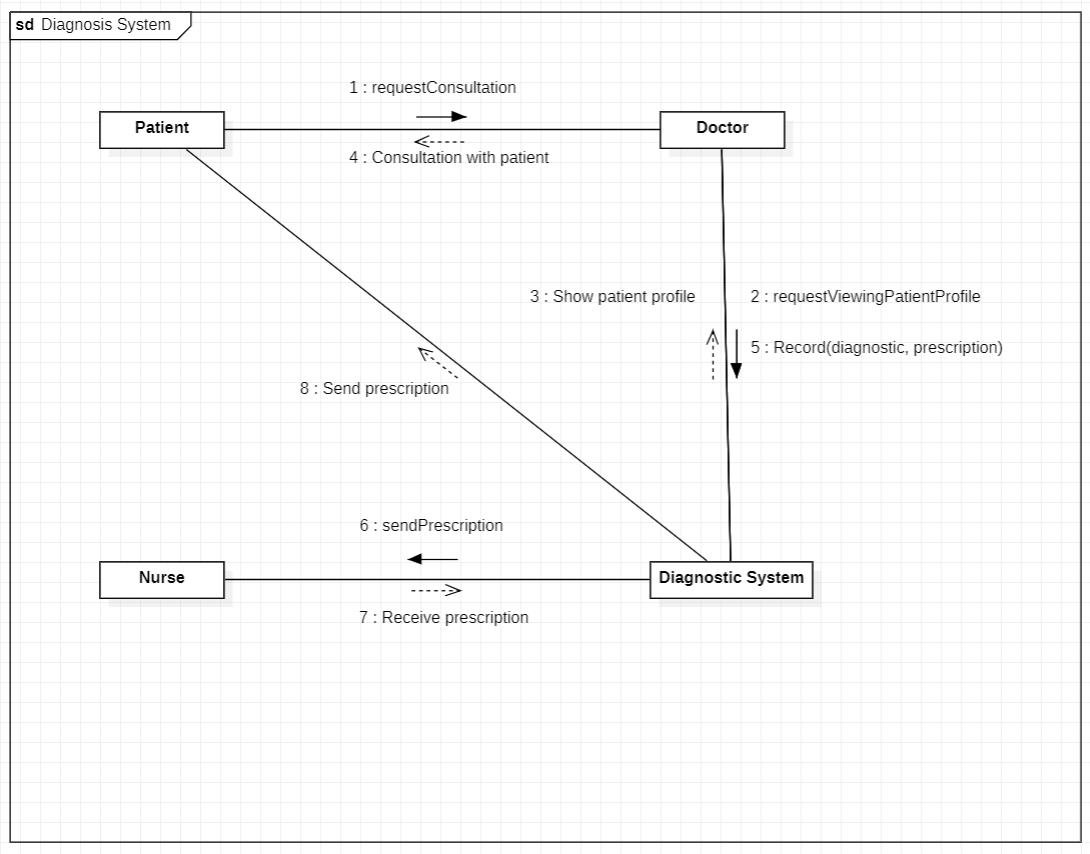


**Collaboration/ Communication Diagram**

[Problem 1] Smart Appointment System



[Problem 1] Diagnostic System

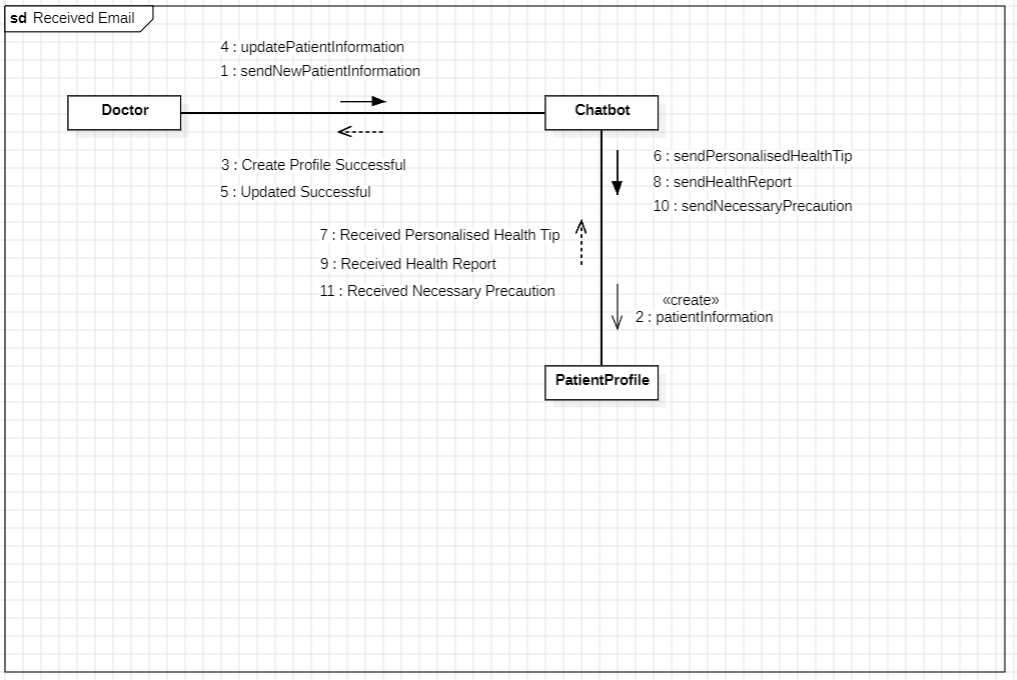


[Problem 2] Intelligent Navigation System

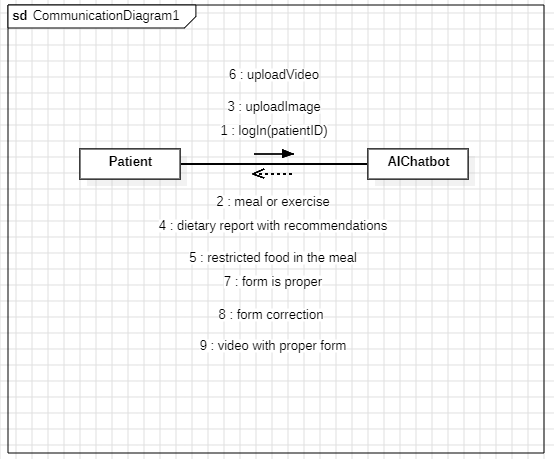
A screenshot of a computer program

Description automatically generated

[Problem 3] Personalised AI-Generated Email



[Problem 3] AI Analysis system

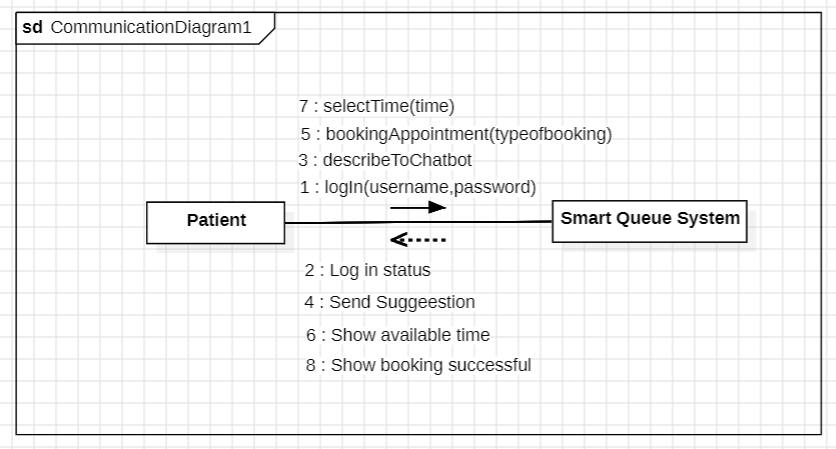


[Problem 4] Computer Vision Supervised Medicine Dispensing System

A diagram of a diagram

Description automatically generated

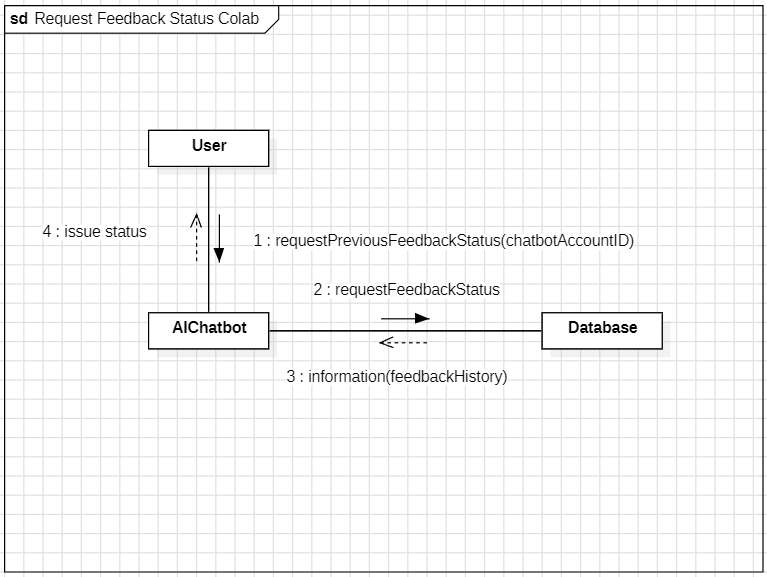
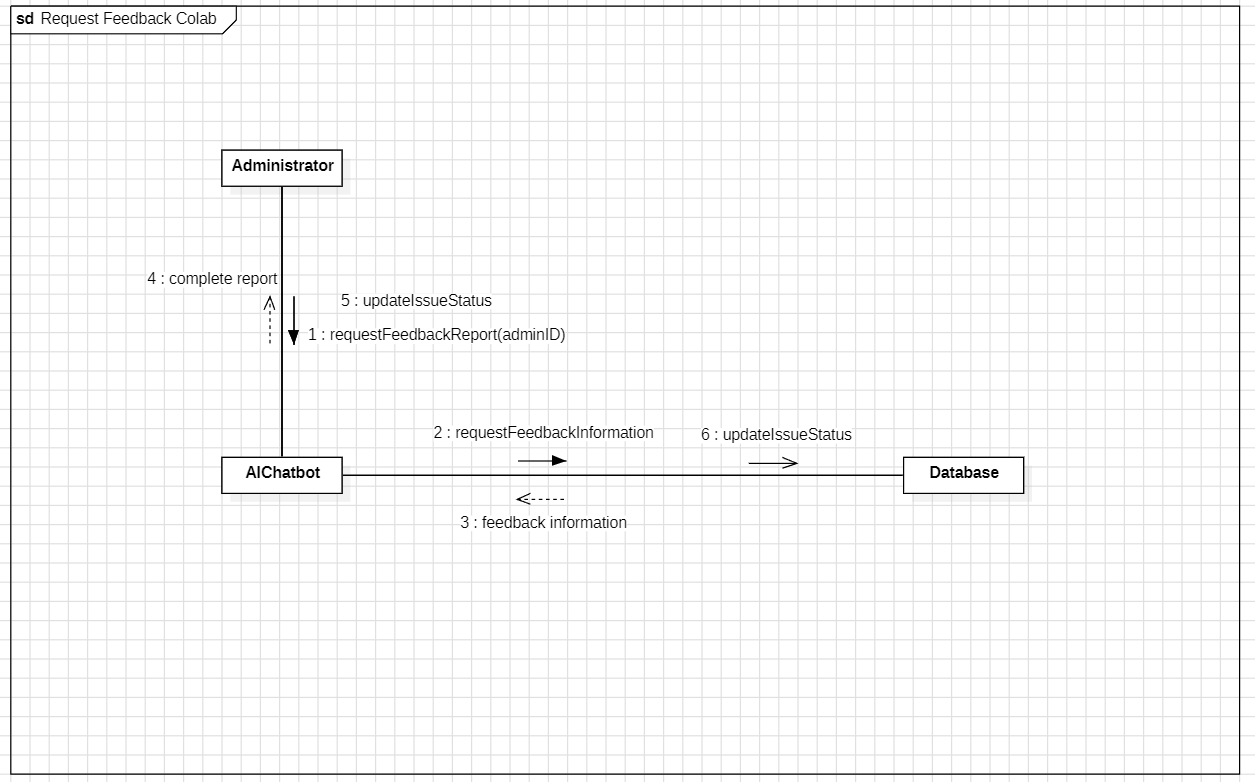
[Problem 5] Smart Queue System



[Problem 6] AI Feedback Mechanisms A diagram of survey

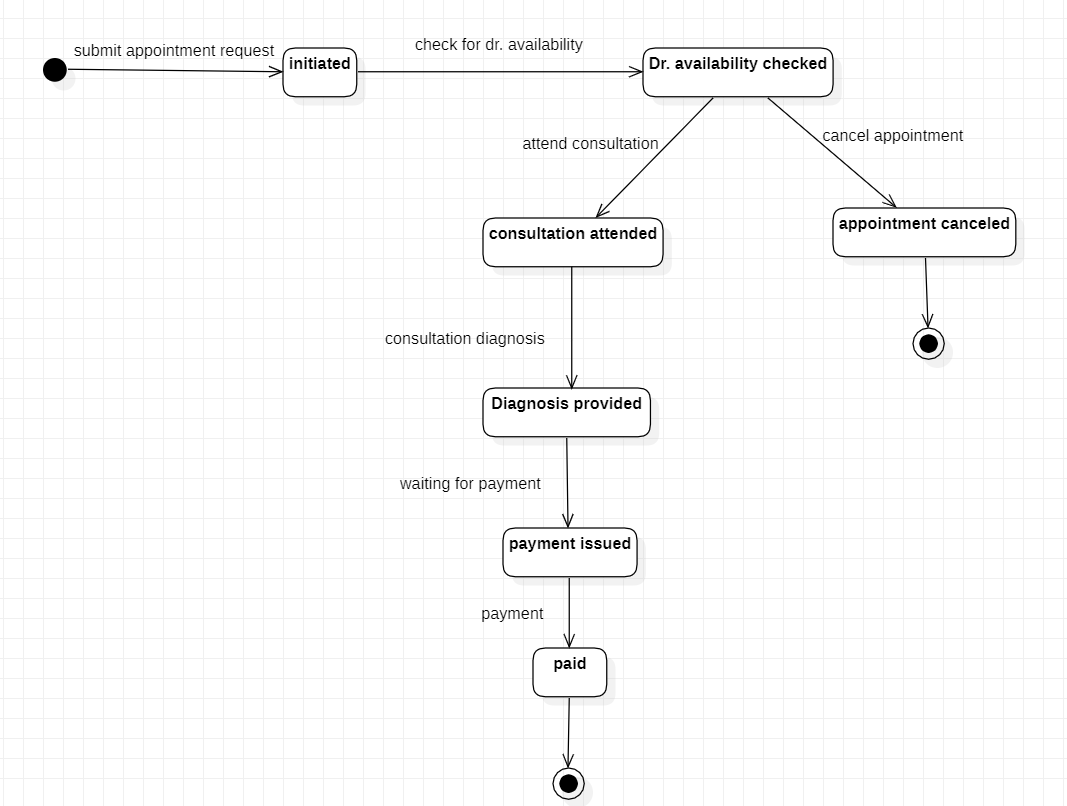
Description automatically generated

[Problem 6] AI Feedback Mechanisms



**State Machine Diagram**

[Problem 1]



[Problem 4] Computer Vision Supervised Medicine Dispensing System

A diagram of a medical procedure

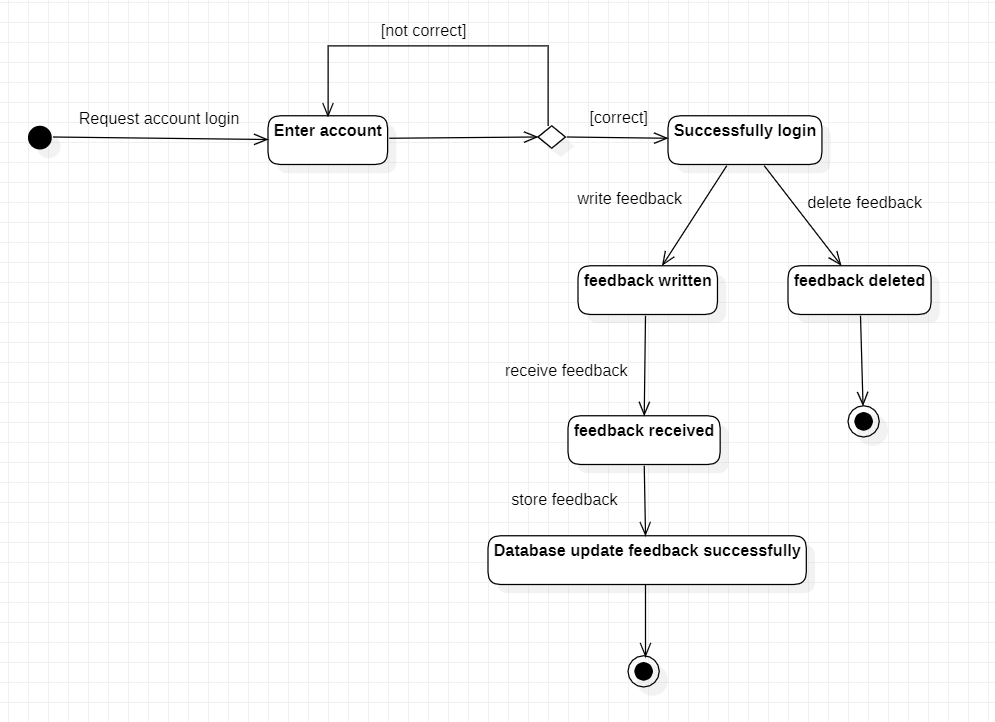
Description automatically generated

[Problem 5] Smart Queue System

A diagram of a computer

Description automatically generated

[Problem 6] AI Feedback Mechanisms



**Component and Deployment**

