

# Software Engineering

## Project Deliverable 3

### Version 2: Updates from the Previous Version

#### Section 1: Use Case Modifications

- Titles of Use Cases have been updated to eliminate the use of 'ing'.
- We deleted some out of scope steps and extensions. We left some extensions that we didn't implement as well (assuming that we didn't have to implement all extensions).
  - We updated other diagrams and operation contracts accordingly.

#### Section 2: Sequence Diagram Adjustments

- Removing redundant initializations from user to system across all SSDs based on feedback.
- Updated SSD 2 to include searchRecipient() and returnRecipient() methods to align with the corresponding use case and the sequence diagram.
- Modified SSD 3 by adding the patient's name as a parameter to ensure consistency with the use case description.
- For Deliverable 3 resubmission, I corrected the SSD by eliminating the double initialization of the interface and system for each use case and replacing it with a single initiation arrow from the user to the system.

#### Class Diagram Updates:

- Introduced 'System Database Handler' as a class in our class diagram. This class is not implemented in the code because Firebase Firestore is an external database that does not require a handler. Nonetheless, the methods defined in this class correspond to the functionalities we have implemented using our database.
- Removed the 'PatientDatabaseHandler' class and integrated its functionality into the 'System Database Handler', consolidating database operations into a single handler due to a unified database architecture.

#### Notes:

- Although the 'System Database Handler' class is detailed in our diagrams for completeness, it is not physically implemented in the codebase, as Firebase Firestore handles these operations externally.

**Team Members:** Symbat Bekzhigit, Sanjana Nambiar, Saamia Shafqat, and Levith Andrade Cuellar

## Use Case Scenarios

### Section 1 (5 points):

Include the five Use Case Scenarios that your team will move forward with.

**USE CASE: MESSAGE A USER**

**Primary Actor:** Doctor / Nurse

**Main success scenario:**

1. **Doctor / Nurse** accesses the platform's messaging section.
2. **System** displays the hospital's directory and an option to start a new conversation.
3. **Doctor / Nurse** selects the doctor or nurse they'd like to start a conversation with.
4. **System** displays a chat interface, a location for input, a keyboard and a send button.
5. **Doctor / Nurse** inputs their message and sends it.
6. **System** encrypts the message content using the relevant encryption standards.
7. **System** delivers the message to the recipient and updates the interface with the sent message.

**Extensions:**

Searching Directory:

- 3a. **System** displays the profiles in alphabetical order.
- 3b. **System** updates display based on changes in the database.

Acknowledgment Receipts:

- 8a. **System** displays if the message has been delivered.
- 8b. **System** displays if the message has been read.

**USE CASE: PAGE A USER** (Assuming the user is logged into the application)

**Primary Actor:** Doctor / Nurse

**Main success scenario:**

1. **Doctor / Nurse** accesses the platform's pager section.
2. **System** displays a set of options to construct the request: recipient name, location, notification type, and custom message.
3. **Doctor / Nurse** searches for the recipient's name using the hospital's directory.
4. **Doctor / Nurse** selects a location for the request and a type of notification using a set of options.
5. **Doctor / Nurse** inputs text for the custom message.
6. **Doctor / Nurse** sends the request.
7. **System** delivers the request to the designated recipient.

**Extensions:**

Request Drafting:

- 5a. **System** alerts the doctor/nurse when the word count is reached.
- 6a. **System** prevents the doctor/nurse from sending a request with missing information.

Notification Failure

- 7a. **System** alerts the doctor/nurse they're offline and asks them to use a different method for processing an urgent request.
- 7b. **System** logs the notification failure.

Notification Acknowledgement

8a. **System** forwards the notification to another healthcare professional in case the original recipient does not acknowledge the request in a timely manner.

#### **USE CASE: VIEW AN ELECTRONIC HEALTH RECORD (EHR)**

**Primary Actor:** Doctor / Nurse

**Main success scenario:**

1. **Doctor / Nurse** accesses the platform's records section.
2. **System** displays a list of patients (name, photo, age, gender).
3. **Doctor / Nurse** selects a patient whose information they'd like to view.
4. **System** displays the contents of the patient's Electronic Health Record (EHR): personal information, medical history, diagnoses, etc.

**Extensions:**

2a. **System** displays the list in order of treatment start date.

Invalid Search Criteria:

- 4a. **Doctor / Nurse** searches for a patient that they're not assigned to.
- 4b. **System** alerts the user that it was unable to locate the patient they are looking for.
- 4c. **System** allows the user to revise their search criteria or initiate a new search.

Database Connection Failure:

- 5a. **System** encounters issues connecting to the hospital's database during the data retrieval process.
- 5b. **System** displays an error message indicating the database connection failure and advises the user to check their internet connection or try again later.
- 5c. **Doctor / Nurse** can attempt to reconnect to the database.

#### **USE CASE: CREATE A REMINDER**

**Primary Actor:** Doctor / Nurse

**Main success scenario:**

1. **Doctor / Nurse** accesses the platform's reminder section.
2. **System** displays the current reminders and displays an option to create a new reminder.
3. **Doctor / Nurse** selects an option to create a new reminder.
4. **System** displays a set of options to construct the reminder: title, date and time.
5. **Doctor / Nurse** selects the appropriate date and time associated with the reminder.
6. **Doctor / Nurse** adds the reminder.
7. **System** saves the reminder and displays it with the other current reminders.

**Extensions:**

Arranging existing reminders

- 7a. **Doctor / Nurse** selects the option to delete a reminder.

7b. **Doctor / Nurse** selects the option to update a reminder.

Invalid Reminder Details

4a. **Doctor / Nurse** enters incomplete or invalid information for the reminder.

4a. **System** verifies the chosen date is in the future.

**USE CASE: PERFORM AI-POWERED REQUEST**

**Primary Actor:** Doctor / Nurse

**Main success scenario:**

1. **Doctor / Nurse** accesses the platform's records section.
2. **Doctor / Nurse** selects a specific patient's record to work with.
3. **System** presents predetermined prompts with possible queries that can be made to the AI module.
4. **Doctor / Nurse** selects the query.
5. **System** utilizes the AI module to process the user's query.
6. **System** displays the AI-generated query results.
7. **Doctor / Nurse** reviews the query results.

**Extensions:**

Inaccurate Summarization:

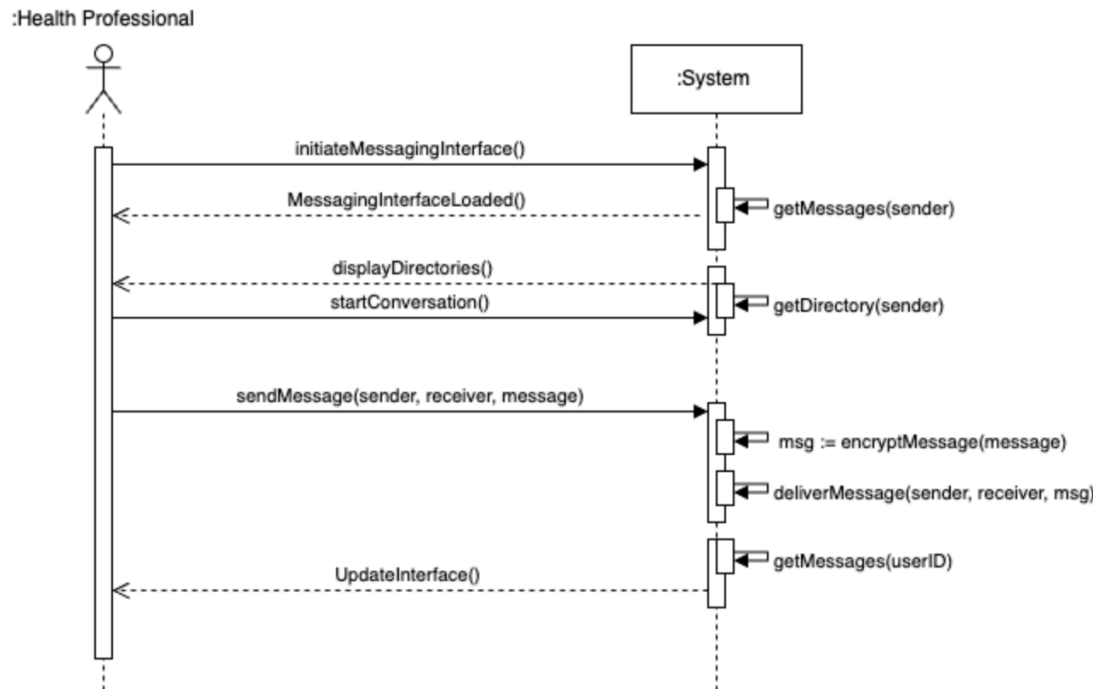
7a. **Doctor / Nurse** may choose to manually review the patient records for accuracy or request further clarification from the AI module with additional queries.

Unresponsive AI:

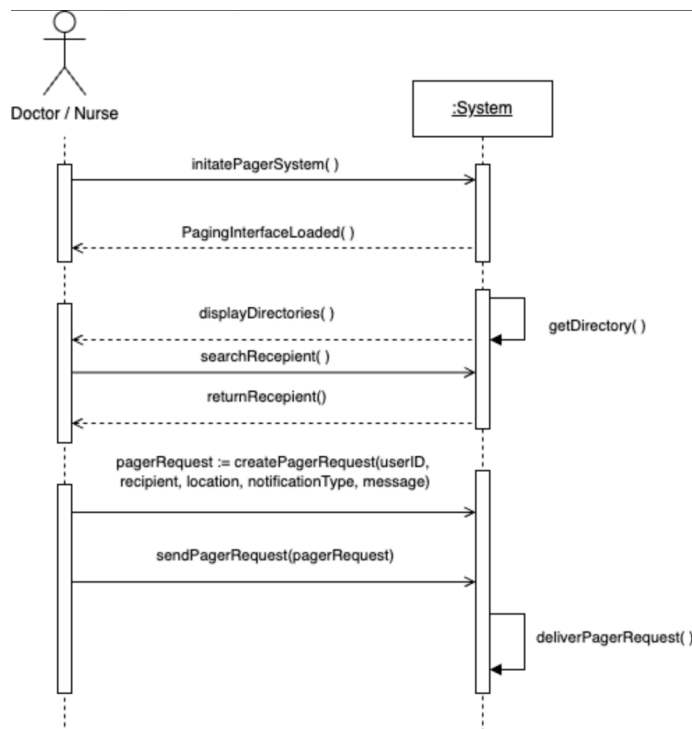
- 1a. **System** displays an error message indicating the unavailability of the AI service and suggests trying again later.
- 2a. **Doctor / Nurse** may opt to wait for the AI service to become responsive again or proceed with manual review of the patient records in the meantime.

## System Sequence Diagram (SSD)

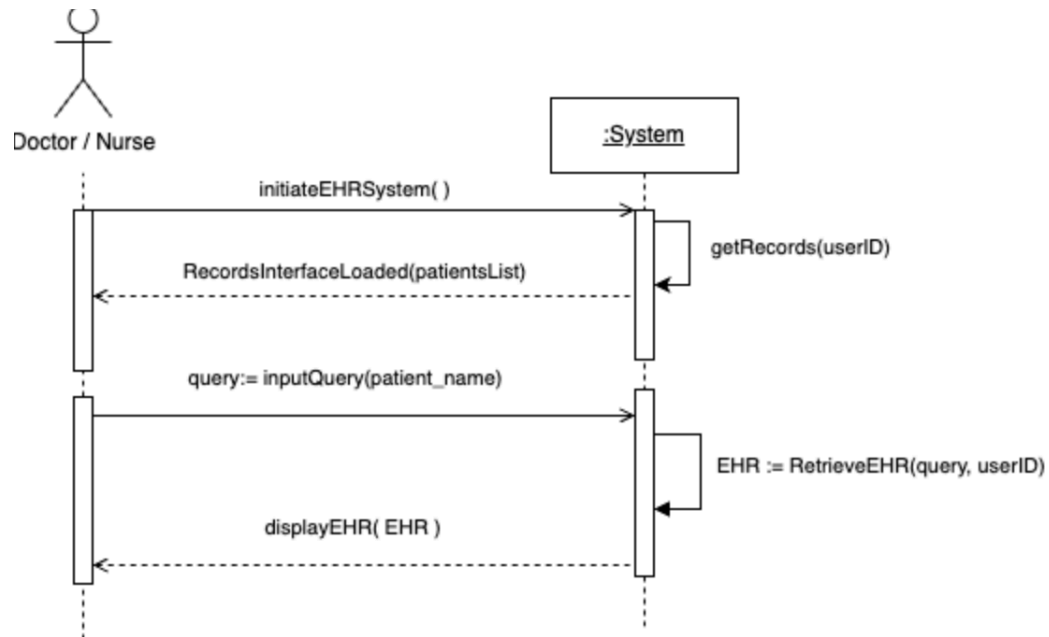
### 1. Messaging SSD



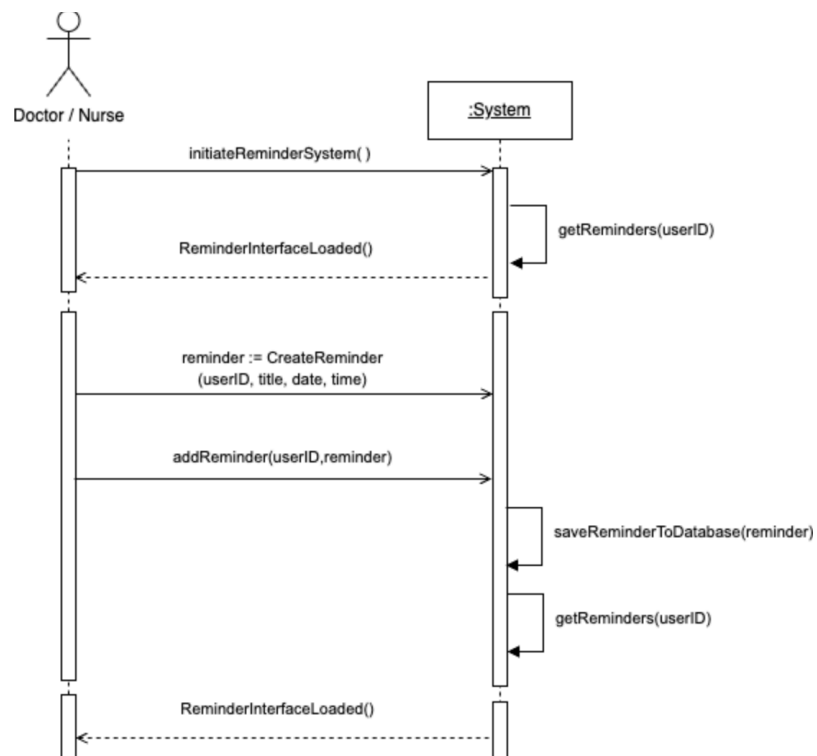
## 2. Pager SSD



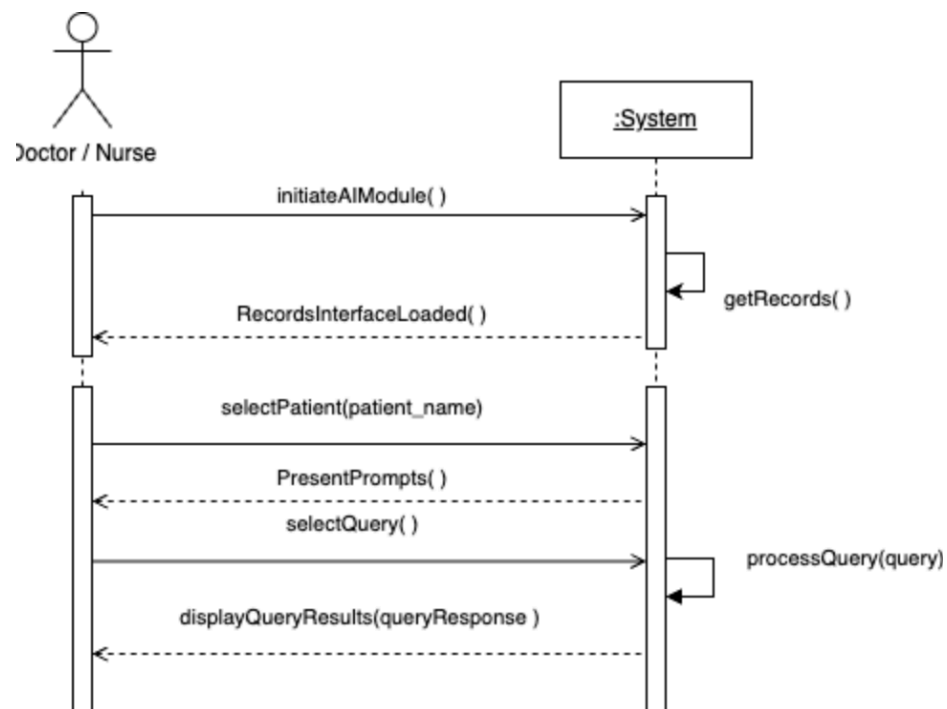
## 3. Viewing EHR SSD



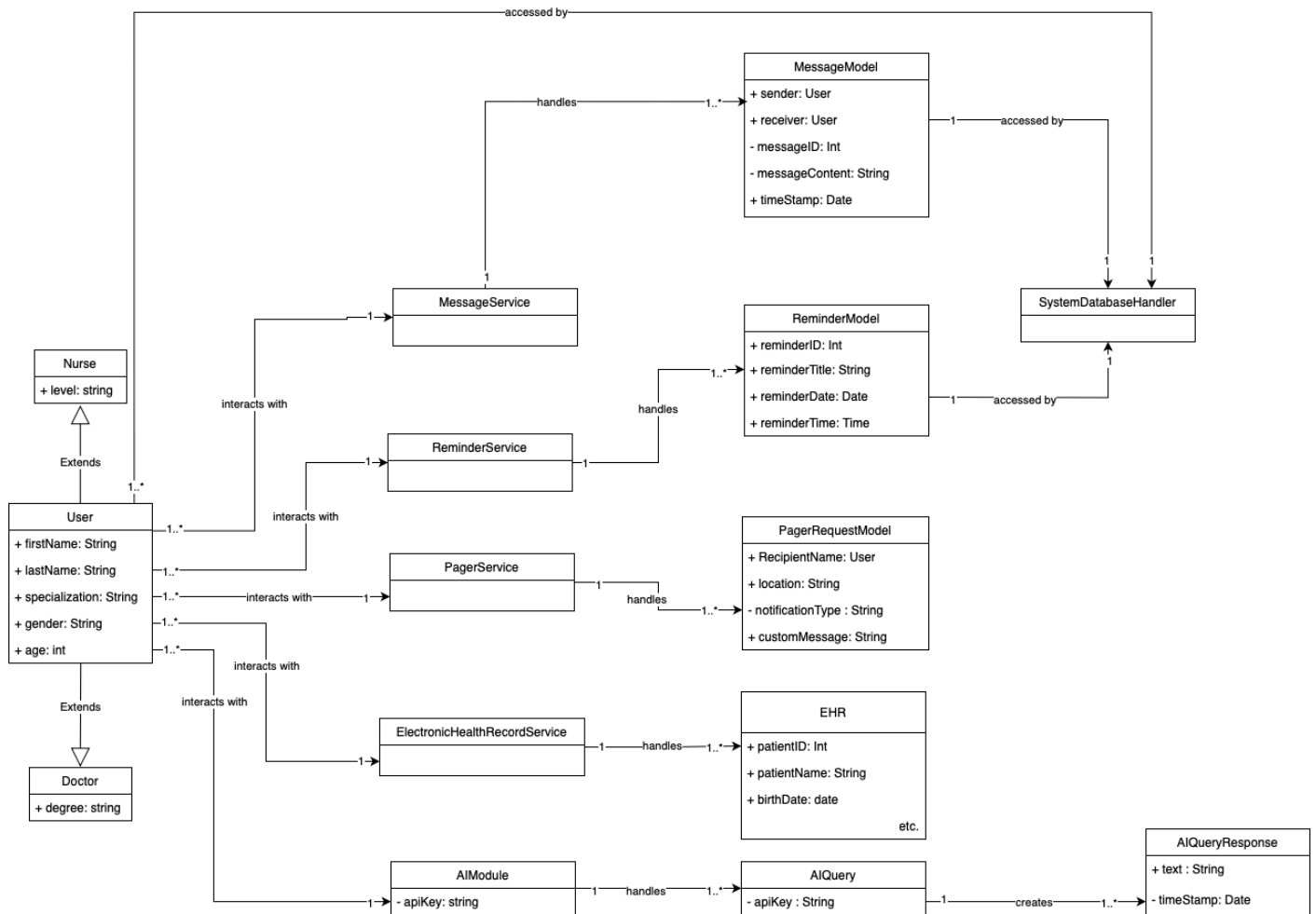
#### 4. Creating Reminder SSD



#### 5. AI Module SSD



## Domain Model



## Operation Contracts

### USE CASE: MESSAGE

<b>Name:</b>	sendMessage(message)
<b>Responsibilities:</b>	Facilitates sending a message from one user (sender) to another (recipient).
<b>Type:</b>	System
<b>Cross References:</b>	Use Cases: Message a user
<b>Notes:</b>	
<b>Exceptions:</b>	<p>If the message is empty or null, return an error.</p> <p>If the message couldn't be sent then return an error.</p> <p>If the system cannot encrypt or deliver the message, return an error.</p>
<b>Output:</b>	<p>Confirmation that the message was sent.</p> <p>Updates to the messaging interface to reflect the new message.</p>
<b>Pre-conditions:</b>	<p>User IDs, for both the sender and recipient, are known to the system.</p> <p>All arguments were provided.</p>



**Post-conditions:**

- A new Message object is created with the message content, senderID, and recipientID (*instance creation*).
- Message added to the *conversationList* between the users (*attribute modification*)

**USE CASE: PAGER**

**Name:** createPagerRequest(userID, recipient, location, notificationType, message)

**Responsibilities:** Enter pager request details and create a request object.

**Type:** System

**Cross References:** Use Cases: Page a User

**Notes:**

**Exceptions:** If User ID is not valid, indicate an error.

If an argument is missing, indicate an error.

**Output:** Notification received by the recipient

**Pre-conditions:** User ID is known to the system.

All arguments were provided.

**Post-conditions:**

- If a new request, a *PagerRequest* was created (*instance creation*)
- *PagerRequest.recipient* was set to *recipient* (*attribute modification*)
- *PagerRequest.location* was set to *location* (*attribute modification*)
- *PagerRequest.notificationType* was set to *notificationType* (*attribute modification*)
- *PagerRequest.message* was set to *message* (*attribute modification*)

**USE CASE: CREATE REMINDER**

**Name:** createReminder(userID, title, type, date, time)

**Responsibilities:** Enter reminder details, create a reminder and add it to the user's reminders.

**Type:** System

**Cross References:** Use Cases: Create Reminder

**Notes:**

**Exceptions:** If User ID is not valid, indicate an error.

If an argument is missing, indicate an error.

**Output:** Reminder created in database and loaded on the interface.

**Pre-conditions:** User ID is known to the system.

All arguments were provided.

**Post-conditions:**

- If a new reminder, a *Reminder* was created (*instance creation*)

- If a new reminder, the new *Reminder* was associated with the *User Reminders* that matches *user ID* (*association formed*).
- *Reminder.title* was set to *title* (*attribute modification*)
- *Reminder.date* was set to *date* (*attribute modification*)
- *Reminder.time* was set to *time* (*attribute modification*)

### **USE CASE: VIEW EHR**

<b>Name:</b>	RetrieveEHR(userID, query)
<b>Responsibilities:</b>	To display an Electronic Health Record based on a specific query by the user. System Operation
<b>Type:</b>	Use Cases: View an Electronic Health Record
<b>Cross References:</b>	This operation is integral to the system's functionalities, allowing users to access and review EHRs.
<b>Notes:</b>	
<b>Exceptions:</b>	If the query does not match any records, the system should inform the user that no matching records were found.
<b>Output:</b>	Displays the Electronic Health Record that matches the query.
<b>Pre-conditions:</b>	The user who is searching must have access to the records they're querying. A query must be provided by the user to specify which EHR is to be retrieved.
<b>Post-conditions:</b>	<ul style="list-style-type: none"> <li>• If a new request, <i>EHR</i> was created (<i>instance creation</i>)</li> <li>• <i>EHR.patientName</i> was set based on the retrieved record (<i>attribute modification</i>)</li> <li>• <i>EHR.details</i> is filled with the medical history, diagnosis, treatment plans, and notes contained within the EHR (<i>attribute modification</i>).</li> </ul>

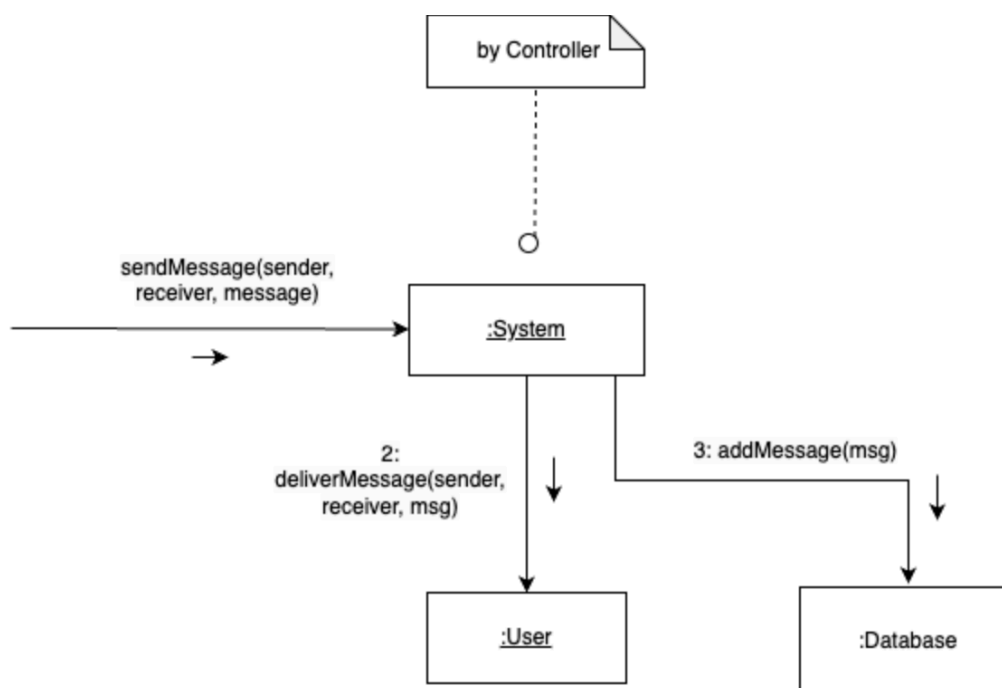
### **USE CASE: PERFORM AI-POWERED REQUEST**

<b>Name:</b>	processQuery(ehr, query)
<b>Responsibilities:</b>	To perform a specific AI query relating to a selected EHR.
<b>Type:</b>	System Operation
<b>Cross References:</b>	Use Cases: Perform an AI Query
<b>Notes:</b>	Will adhere to healthcare data compliance standards.
<b>Exceptions:</b>	If the AI server is down, then an error warning will be generated
<b>Output:</b>	An AI-generated summary or analysis result of the patient's health record based on the query.

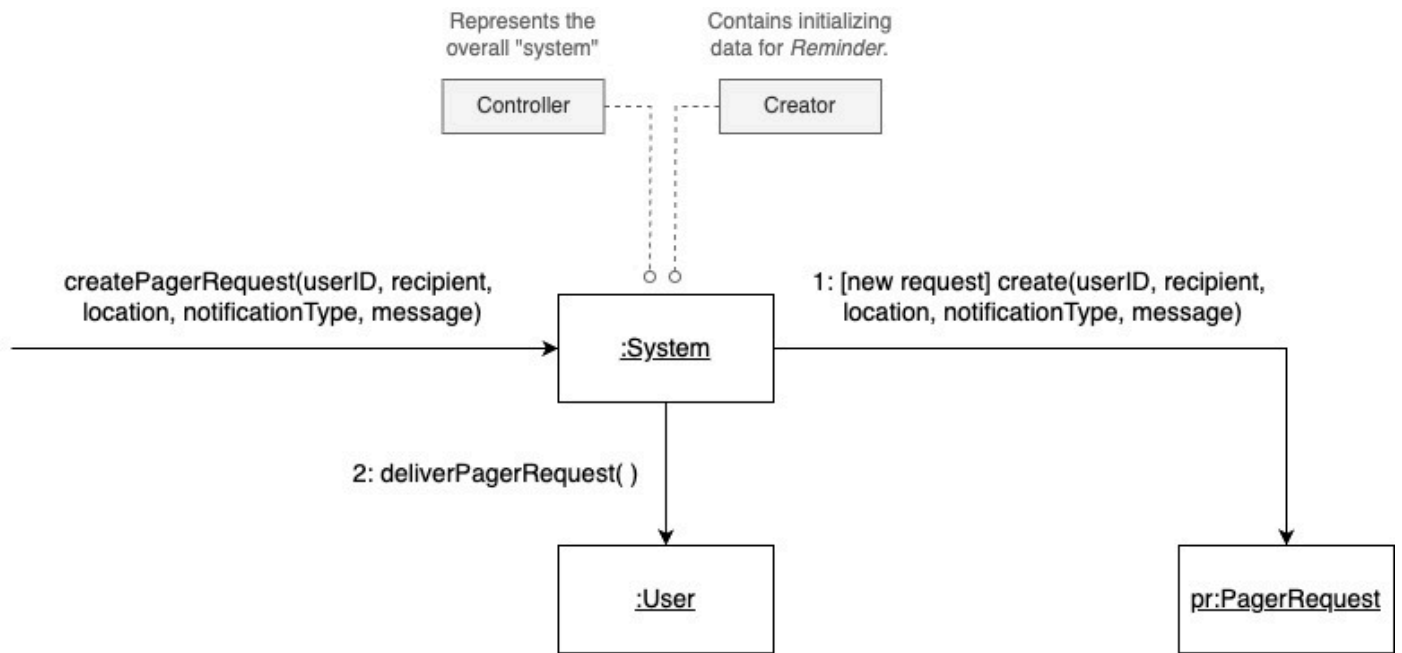
<b>Pre-conditions:</b>	<p>The user has already retrieved the EHR record of the patient.</p> <p>The user has selected the query to be given to the AI.</p>
<b>Post-conditions:</b>	<ul style="list-style-type: none"> <li>• If a new query, a <i>queryResponse</i> was created (<i>instance creation</i>)</li> <li>• <i>queryResponse.text</i> was set to <i>AI query results</i> (<i>attribute modification</i>).</li> </ul>

## UML Interaction Diagram

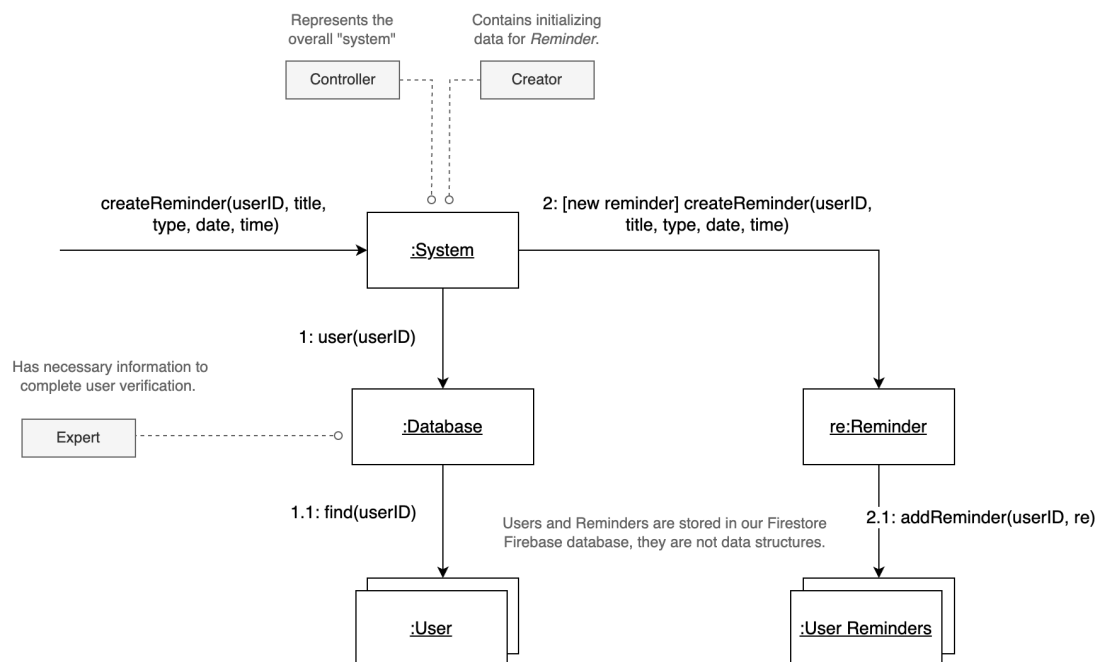
### USE CASE: MESSAGE



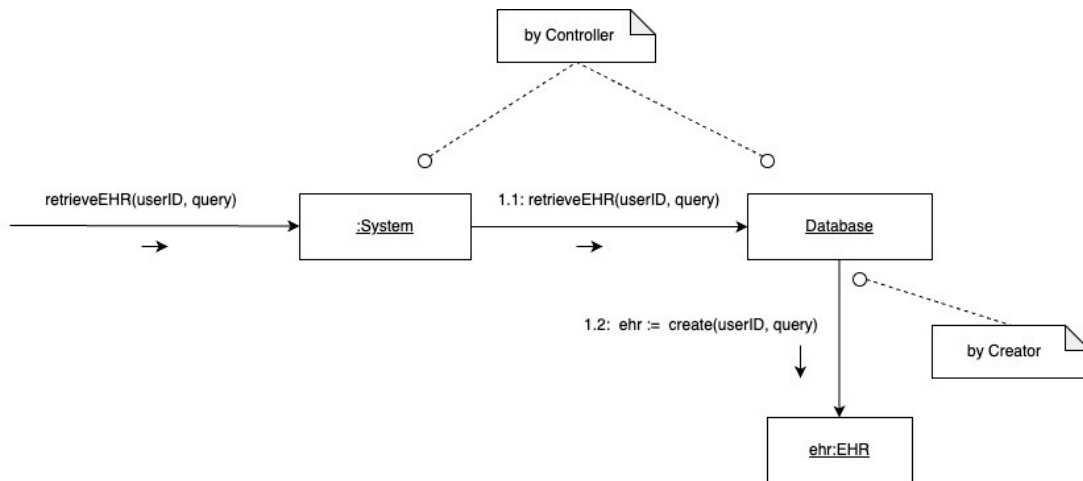
### USE CASE: PAGER



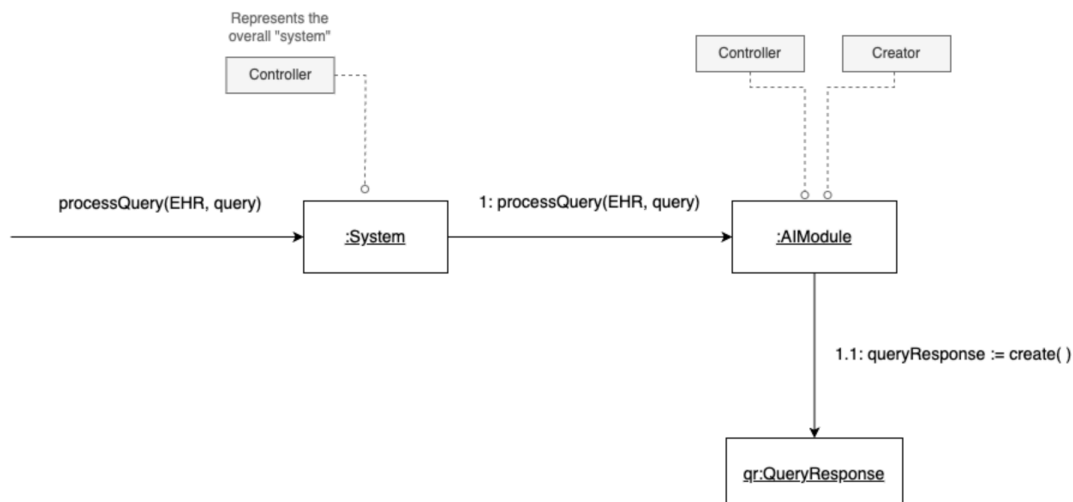
## USE CASE: CREATE REMINDER



## USE CASE: VIEW EHR



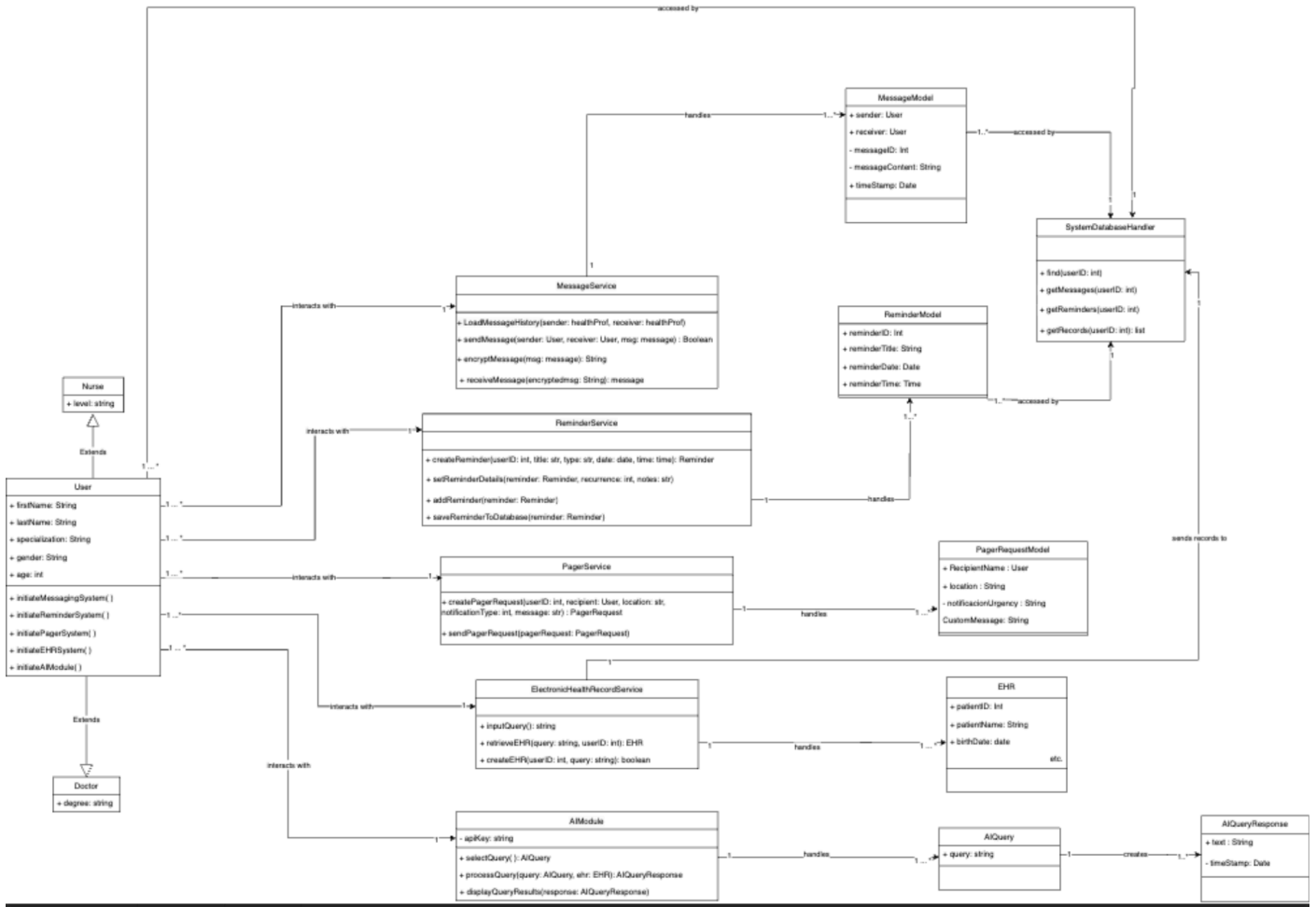
### USE CASE: PERFORM AI-POWERED REQUEST



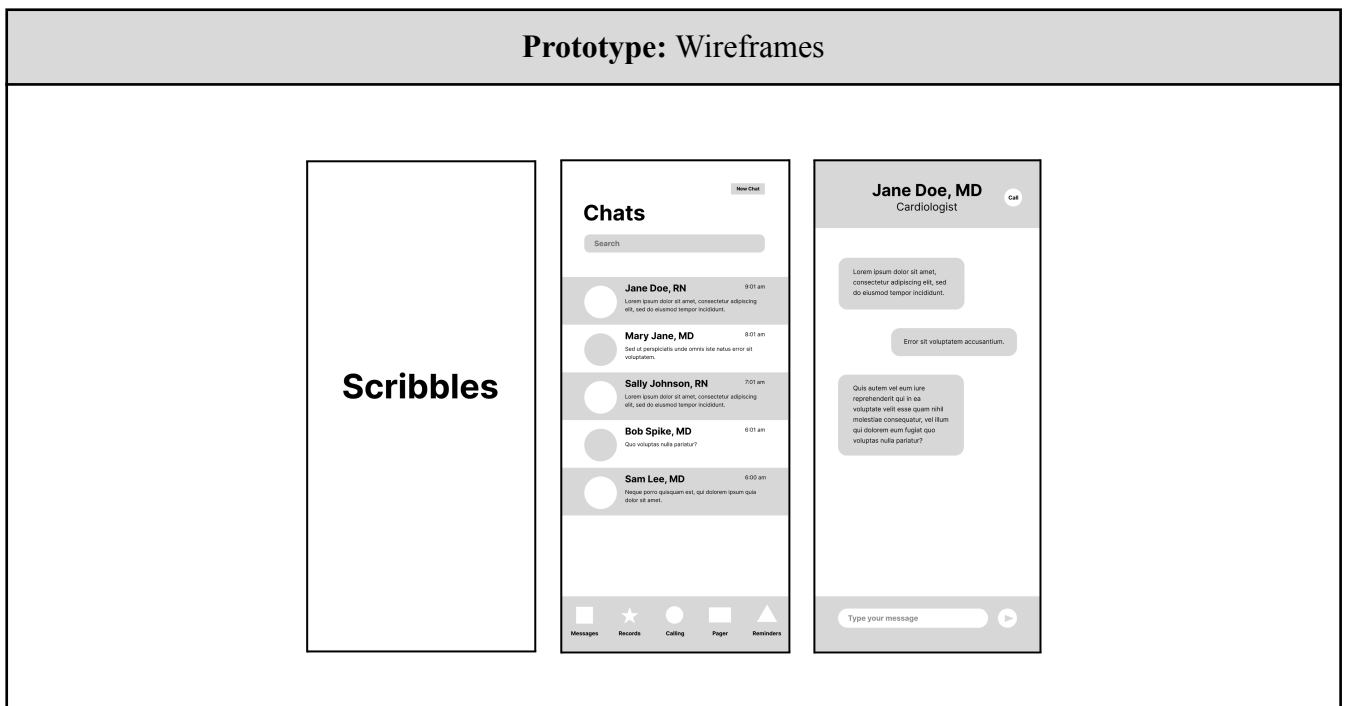
### **Class Diagram**

#### Notes:

We have included 'System Database Handler' as a class in our class diagram, although we did not implement this class in our code because we are using Firebase Firestore as an external database and it does not require a handler. However, the methods in this class accurately represent the functionalities we have implemented using our database.



## Prototype Screens



Patient Record

Robert Brown

Patient

A.I. Summary

A.I. Highlights

Pager

Search for a contact

Select a room number

1-A

1-B

1-C

Select notification type

Urgent

Semi-Urgent

Type a custom message

Send

Reminders

Today

Give pills to patient in 1-A.

9:01 am

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt.

Call Mary Jane, MD.

8:01 am

Sed ut perspiciatis unde omnis iste natus error sit voluptatem.

Call Jane Doe, MD.

7:01 am

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt.

Tomorrow

Take vitals for 1-C patient.

8:01 am

Consectetur adipiscing elit, sed do eiusmod tempor incididunt.

Sit up patient in 1-B.

8:00 am

Tempora porro quisquam est, qui accusam ipsum quia dolor sit amet.

Messages

Records

Calling

Pager

Reminders