Software Security

Project: Design of a Secure Healthcare System

Group 4

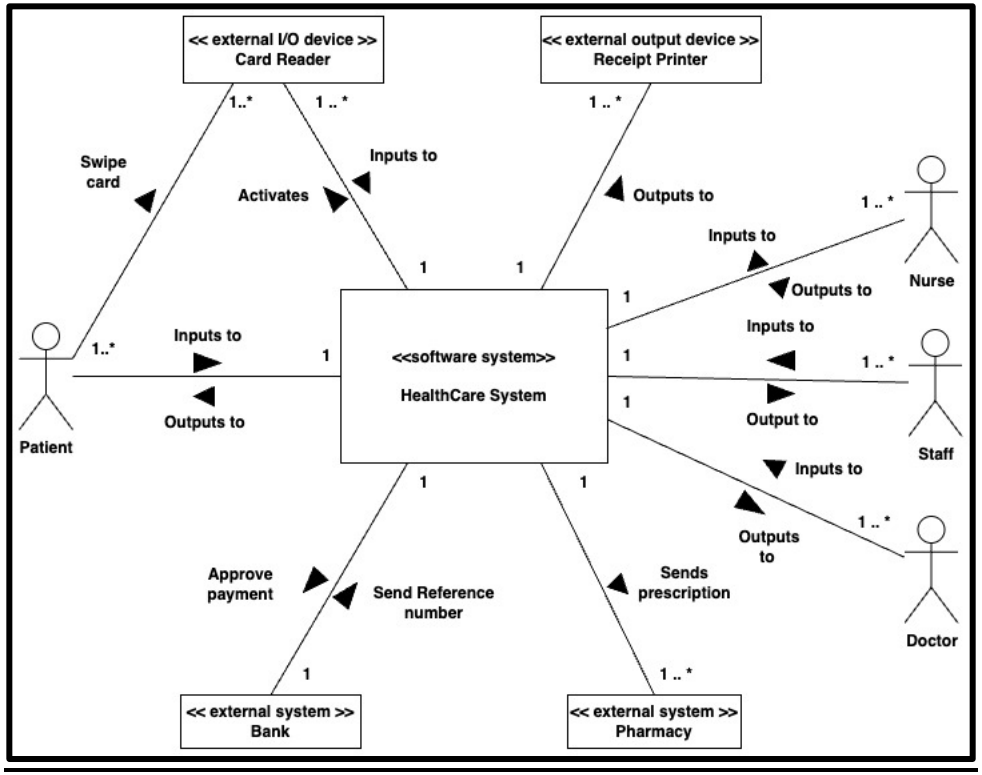
Member Name

Rushikesh Khamkar

System Boundary

Define the system boundary using a software system context model that shows how the system interfaces with the external environment. Develop the threat model for the software system context model, where threats are identified and analyzed. Also, specify the security use cases to mitigate the threats and develop test cases for the use cases.

Software system context model



Assumptions –

We have identified 4 external users in the HealthCare system – Patient, Staff, Doctor, Nurse and 2 external system – Pharmacy and Bank. We have considered many to one relationship from our Healthcare system to external system/users, because there could be many users (nurse, doctors, staffs or patients) connect with Healthcare system at a time.

Threat Model

Threat	Security Assets	Description	Vulnerability	Consequences	Likelihood
Unauthorized access / Unauthenticated patient	System	An unauthenticated patient can access the system.	The system does not authenticate or improperly authenticate patients.	Critical	100%
Data breaches	System	Threats that attempt to steal or compromise sensitive data, such as patient records, financial information, and medical history.	The system does not encrypt patient data to prevent data breach.	Critical	100%
Cross-site scripting attack	System	Attacker gains access to the system while updating the information.	Lack of input validation and output encoding communication	Critical	100%
DoS attack	System	A system can be degraded or stop services. Threats that attempt to disrupt the availability of the system, such as DDoS attacks, bot attacks.	The system does not monitor service requests flooded and block the IP address requesting continuous requests.	Critical	100%
Man-In-The-Middle Attack	System	If third party manages to insert themselves between healthcare and pharmacy, Victim can fetch patient information and use it for any illegal activities.	Modify patient record and change patient’s account settings	Critical	100%
Session Hijacking	System	Attacker can hijack user session by stealing session cookies or using other methods to take over.	It allows attackers to impersonate the users and perform action on.	Critical	100%
Insider threats	System	Threats that come from within the organization, such as malicious employees, negligence, and unintentional actions.	The system does not implement access controls to limit employees' access to critical use cases.	Critical	90%

Security Use-Cases to mitigate threats and Test cases

Mitigation for Threat 1 - Unauthorized access / Unauthenticated patient –

Security Use case: Two-factor authentication with time limitation.

Summary: The system verifies the patient's identity using two-factor authentication and applies a time limitation to enter the second credentials.

Actor: Patient (Healthcare System)

Precondition: The patient is registered in the system.

Main sequence:

1. Patient enters ID and password.
2. The System checks the correctness of the patient's ID and password.
3. If the patient's credentials are correct, the system generates a one-time password (OTP) and sends it to the patient's registered email or phone number.
4. The system prompts the patient to enter the OTP within a specified time limit.
5. The Patient enters the OTP and submits it to the system.
6. The system verifies the correctness of the OTP and grants access to the patient if the OTP is correct.
7. System displays a welcome message if the patient's credentials and OTP are correct.

Alternative sequence:

- Step 2: The system terminates if the patient's credentials are incorrect.
- Step 2: The system terminates if the patient's credentials are not entered correctly the second time within the specified time limit.
- Step 4: The system terminates if the OTP is not entered within the specified time limit.
- Step 6: If the OTP is incorrect, the system prompts the patient to enter the correct OTP. The system terminates if the patient does not provide the correct OTP for the second time.

Postcondition: The system has verified the patient's identity using two-factor authentication with time limitation. The patient has been granted access to the system.

Test Case -

Assumption: Patient Name: John Doe ID: johndoe123 Password: Abcd@123 OTP: 456789 Time Limit: 60 seconds

T1: John Doe, johndoe123, Abcd@123, 456789, 60 seconds // Valid

T2: John Doe, johndoe123, Abcd@123, 123456, 60 seconds // Invalid OTP

Mitigation for Threat 2 – Data Breach

Security use case: Encrypt Patient Information to Prevent Data Breach

Summary: The system encrypts patient information to prevent data breach.

Actor: Healthcare Provider (System)

Precondition: None

Main Sequence:

1. System retrieves a key for encryption.
2. System encrypts the patient information using the key if the key is available.
3. System returns the encrypted credit card information if the key is valid.

Alternatives:

- Step 2: The system displays a key error if the key is unavailable.
- Step 3: The system returns an encryption error if the key is invalid.
- Step 3: The system denies access if an unauthorized person tries to access the encrypted patient information.

Postcondition: Patient information has been encrypted; only authorized personnel can access it.

Test Case -

Assumption: Patient Name: Jane Smith Information: Blood Group - A+ , Height - 5'5" , Weight - 130 lbs

T1: Jane Smith, Key Available, Blood Group - Encrypted, Height - Encrypted, Weight - Encrypted // Valid

T2: Jane Smith, Key Unavailable, Blood Group - NA, Height - NA, Weight - NA // Invalid key not available for encryption.

Mitigation for Threat 3 – Cross-site scripting attack

Security use case: Prevent Cross-site scripting attack

Summary: The system prevents cross-site scripting attacks during the patient's profile update. Actor: System Precondition: The patient is updating their profile, and a malicious attacker is attempting a cross-site scripting attack.

Actor: Healthcare Provider (System)

Precondition: None

Main Sequence:

1. The system validates all input data entered by the patient for possible malicious code.
2. System checks if the input data contains any special characters or HTML code that could be used in a cross-site scripting attack.
3. System sanitizes the input data by removing any special characters or HTML code that could be used in a cross-site scripting attack.
4. System displays the sanitized data to the patient to ensure that it is still accurate and complete.

Alternatives:

- Step 3: If the system detects any malicious code in the input data, it displays an error message and prompts the patient to enter the correct information. Postcondition: The patient's profile has been updated securely, and the system has prevented any cross-site scripting attacks.

Postcondition: The patient's profile has been updated securely, and the system has prevented any cross-site scripting attacks

Test Case

Assumption: Patient Name - John Smith, Update Profile - "Hello, <script>alert('XSS')</script> World"

T1: (John Smith, Hello, World) // Valid Input, the system sanitizes input data and removes any malicious code.

Assumption: Patient Name - Sarah Jones, Update Profile - "Hello, World"

T1: (Sarah Jones, Hello, World) // Valid Input, the system validates and sanitizes the input data, no malicious code detected.

Mitigation for Threat 4 – DOS Attack –

Security Use case: Check DOS Attack and Block IP Address.

Summary: The system monitors and controls service requests to prevent Denial-of-Service (DoS) attacks.

Actor: Bot (Healthcare System)

Precondition: A threshold for service requests per minute from the same IP has been defined.

Main Sequence:

1. Bot sends a service request.
2. The system records the IP address of the service requester.
3. System checks if the IP address is in the blacklist or if it receives service requests from the IP address over the threshold.
4. System allows the service request if the IP is not on a blacklist and the number of requests does not exceed the threshold.

Alternative Sequence:

- Step 4: If the IP address is in the blacklist, the system blocks the service request and logs the event.
- Step 4: If the service request is over the threshold, the system blocks the request, adds the IP address to the blacklist, and logs the event.

Postcondition: The system has checked for DoS attacks and blocked the IP address if necessary to prevent further attacks.

Test Case -

Assumption: IP Address - 192.168.1.1, Number of Requests - 100

T1: (192.168.1.1, 100) // Request allowed, not blacklisted, and below threshold.

Assumption: IP Address - 10.0.0.1, Number of Requests - 200

T2: (10.0.0.1, 200) // Request blocked, IP address added to the blacklist, and event logged.

Mitigation for Threat 5 – Man-In-The-Middle Attack –

Security Use case: Secure Connection Establishment

Summary: The system securely establishes a connection with the user to prevent Man-in-the-Middle attacks.

Actor: Healthcare System

Precondition: The user attempts to access the system.

Main Sequence:

1. The system generates a unique session ID and encrypts it using a secure algorithm.
2. The system sends the encrypted session ID to the user's browser.
3. The user's browser decrypts the session ID and sends it back to the system.
4. The system verifies the session ID and establishes a secure connection with the user.

Alternative Sequence:

- If the system detects any suspicious activity or the session ID is invalid, the system terminates the connection and logs the event.

Test Case -

Assumption: User IP Address - 192.168.0.10, Security Protocol - HTTPS

T1: (192.168.0.10, HTTPS) // Valid Input, the system establishes a secure connection using HTTPS protocol.

Assumption: User IP Address - 172.16.0.5, Security Protocol - HTTP

T2: (172.16.0.5, HTTP) // Invalid Input, the system does not establish a secure connection using an insecure HTTP protocol.

Mitigation for Threat 6 – Session Hijacking Attack –

Security Use case: Prevent Session Hijacking

Summary: The healthcare system prevents session hijacking attacks to ensure the staff's secure login and protect sensitive patient data.

Actor: Healthcare System

Precondition: The staff has logged in securely.

Main Sequence:

1. The system generates a unique session ID for the staff upon successful login.
2. The system stores the session ID securely in a server-side database.
3. The system associates the session ID with the staff's user account and IP address.
4. The system sets a session timeout period and invalidates the session if there is no activity within the specified period.
5. The system checks the session ID on each request and only allows access if the session ID matches the one associated with the staff's user account and IP address.
6. The system encrypts the session ID and data transmitted over the network to prevent eavesdropping and data interception.
7. The system monitors the network traffic for any suspicious activity, such as repeated failed login attempts or abnormal session activity.
8. The system logs all user activity, including login attempts, session creation, and session expiration.

Alternative Sequence:

- If the system detects any suspicious activity, it terminates the session and prompts the staff to re-authenticate.

Postcondition: The healthcare system prevents session hijacking attacks and ensures the staff's secure login, protecting sensitive patient data.

Test Case -

Assumption: Doctor Name - Dr. Patel, Appointment Date - May 5th, 2023, 2:30 PM

T1: (Dr. Patel, May 5th, 2023, 2:30 PM) // Valid Input, the system schedules an appointment for the given date and time with Dr. Patel.

Assumption: Doctor Name - Dr. Lee, Appointment Date - April 30th, 2023, 10:00 AM

T2: (Dr. Lee, April 30th, 2023, 10:00 AM) // Valid Input, the system schedules an appointment for the given date and time with Dr. Lee.

Mitigation for Threat 7 – Insider Threats Attack –

Security Use case: Detect and Respond to Suspicious Access Attempts

Summary: To mitigate insider threats, the system will monitor and respond to suspicious access attempts to sensitive healthcare data.

Actor: Healthcare System

Precondition: The system is operational and staff members have access to sensitive healthcare

Main Sequence:

1. The system logs all access attempts to sensitive healthcare data, including the date, time, user, and location of the attempt.
2. The system identifies access attempts that deviate from the normal access patterns of the user or location, such as accessing sensitive data outside of business hours or from a different location than usual.
3. The system generates an alert for security analysts to investigate the suspicious access attempt.
4. Security analysts review the alert and take appropriate action, such as blocking the user's access or escalating the alert to management.
5. The system administrator reviews the security analyst's actions and updates access controls and security policies as necessary.

Alternative Sequence:

- Step 4: If the suspicious access attempt is determined to be a false positive, security analysts will close the alert and update the system to prevent future false positives.

Postcondition: The system can detect and respond to suspicious access attempts to sensitive healthcare data, mitigating the risk of insider threats.

Test Case -

Assumption: User - John Smith, Location - New York, Date and Time - April 28th 2023, 11:00 AM

T1: (John Smith, New York, April 28th 2023, 11:00 AM)// Valid access attempt to sensitive healthcare data

Assumption: User - Jane Doe, Location - Los Angeles, Date and Time - April 28th 2023, 2:00 AM

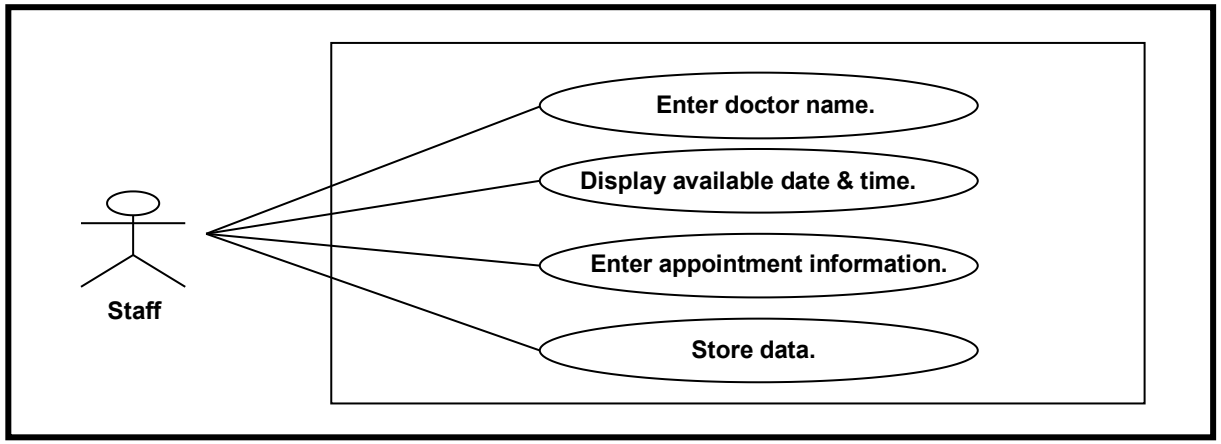
T2: (Jane Doe, Los Angeles, April 28th 2023, 2:00 AM)// Suspicious access attempt outside of business hours

Software Requirements

Identify and analyze threats to each use case to develop the secure use case model. Specify the security use cases to mitigate the threats and indicate where the security use cases extend the application use cases using extension points. Develop test cases for the application use cases and security use cases.

Software security use case model

Use case 1: Make Appointment



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1.Improper patient information	Enter appiontment information (input)	Improper data information from patient	The system does not validate the Patient information	Critical	90%
2.DOS attack	Enter appiontment information (input)	The patient information can be released	patient information input or does not check malware to intercept the input.	Critical	90%

Assumptions:

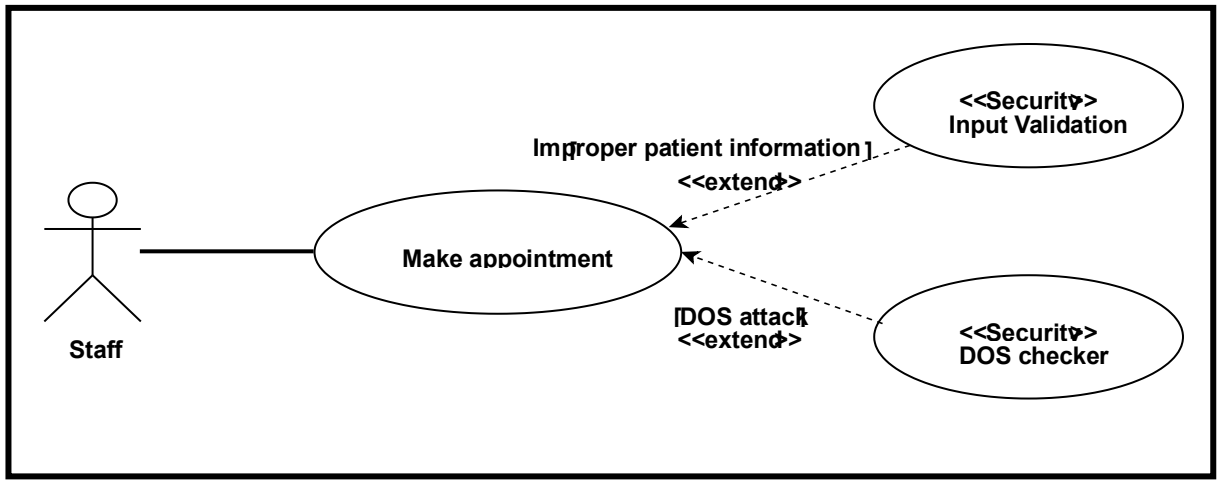
1.Improper Patient information:

The preceding analysis assumes that the system is a healthcare appointment scheduling system that maintains sensitive patient information and, as such, necessitates adequate security measures to secure this information. This likelihood is likely to arise since there may be human error on the patient information side, and the system may be unable to authenticate the information entered by employees.

2. DoS attack:

It also implies that the system is web-based and that the workforce accesses it over the internet, requiring adequate network security measures to prevent DOS assaults. The likelihood is moderate because, while healthcare appointment scheduling systems are not the major target for DOS attacks, they can still be vulnerable if sufficient security measures are not in place. Furthermore, the consequences of a successful DOS attack on the healthcare system could be severe, making it an appealing target for attackers.

Security Use Case:



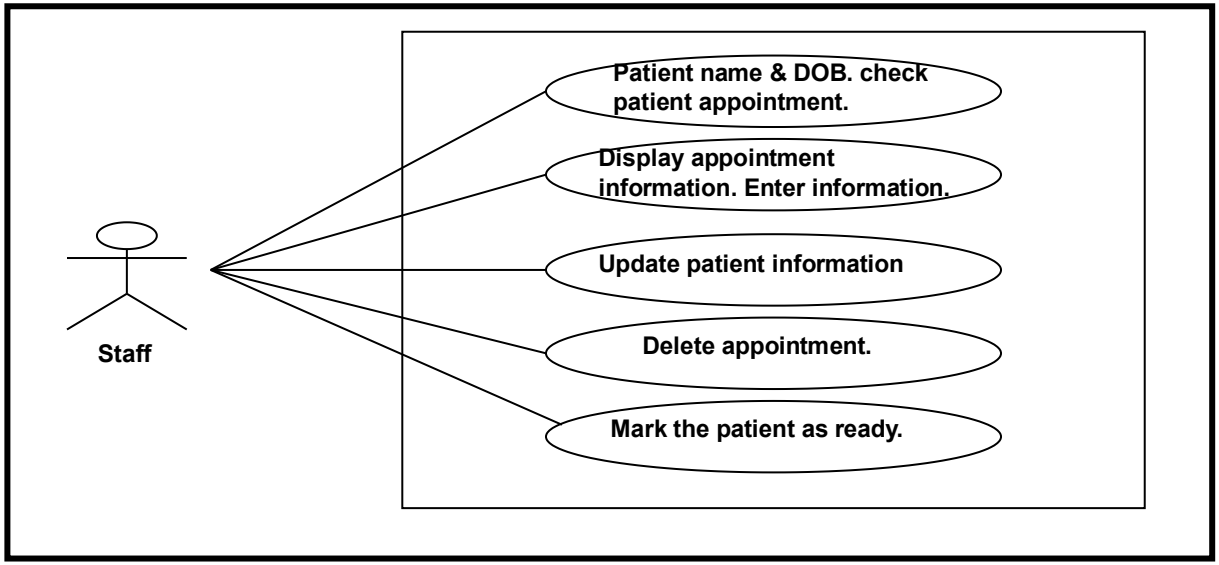
Test Cases:

Assumption: Patient Name - Author, Doctor Name - Dr. Smith, Patient Phone Number – 8976543210, Date of Birth – January 5th 2000, Appointment date and time – April 25th 2023 , 12:00 PM

T1: Author, Dr. Smith, 8976543210, January 5th 2000, April 25th 2023, 12:00 PM)// Valid

T2: (Author, Dr. Smith, 8976543290, January 5th 2000, April 25th 2023, 12:00 PM)// Invalid Patient phone number is wrong

Use case 2: Check-in Patient



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1.SQL command injection	Upadete patient infomation(input)	Attacker can retrive, modify or delete the sensitive infromation from database	Improper input validation	Critical	90%
2. Keylogging Atta	Check patient appiontment (output)	Attacker can use a keylogger to steal senstive information such as patient information	Staff may accidentially download malicious software program	Critical	90%

Assumptions:

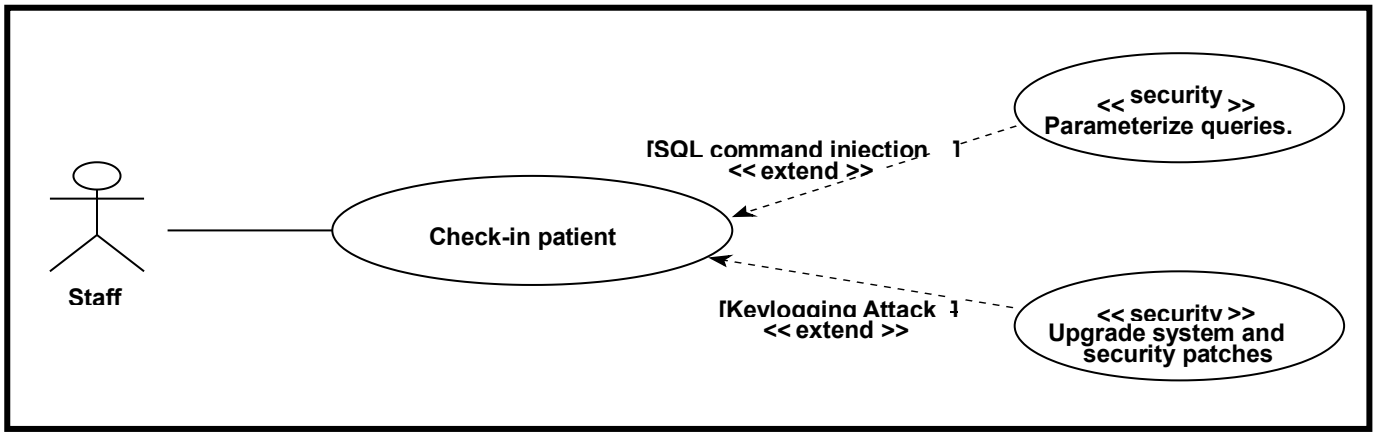
1.SQl command injection:

An attacker gains access to the system by exploiting a system vulnerability or by utilizing stolen credentials. By altering the patient information input fields, the attacker next attempts to introduce malicious SQL queries into the system.90% critical. SQL injection attacks are a prevalent and danger to SQL-based web applications. An attacker is very likely to attempt such an attack, and the repercussions could be serious, including the compromise of personal patient information.

2. Keylogging attack:

On their workstation, a staff worker unintentionally installs a dangerous software program that includes a keylogger. The attacker can then intercept sensitive data entered by the staff member, such as patient appointment information displayed on the screen.90% critical. Attackers frequently utilize keylogging attacks to steal sensitive information. The chances of a staff person inadvertently installing a malicious software program including a keylogger are significant, especially if the staff member does not follow adequate security protocols, and the repercussions might be serious, including the theft of confidential patient information.

Security Use Case:



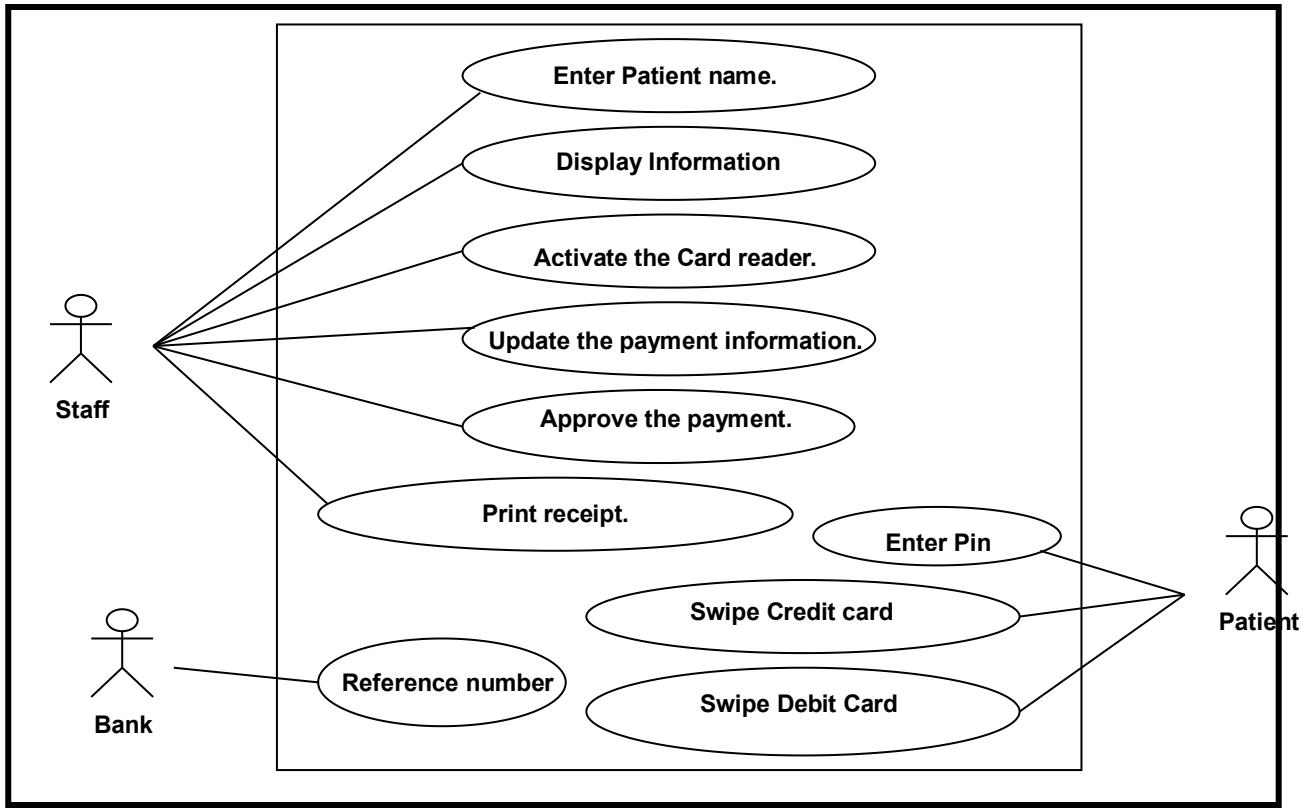
Test case:

Assumption: Patient Name - Author, Date of Birth – January 5th 2000, Appointment information(like number) – 2345A, Patient Address – 3124 16th St unit 123B Austin Texas 73301, Phone Number - 8976543210, SSN - 123456789, Pharmacy and Health Insurance – CVS and DIA123

T1: Author, January 5th 2000, 2345A, 3124 16th St unit 123B Austin Texas 73301, 8976543210, 123456789, CVS and DIA123 //Valid

T2: (Author, January 5th 2000, 2345A, 3124 16th St unit 123B Austin Texas 73301, 8976543210, 123456789, CVS and)//Invalid Patient doesn't have Health Insurance.

Use case 3: Check-out patient.



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1.DoS Attack	Approve Payment (Output)	Attacker can get the card data	Network segmatation	Critical	100%
2.Disclose credit card	swipe card(input)	The card information sent to bank can be disclosed	The system does not encrypt credit card output	Critical	90%

Assumption:

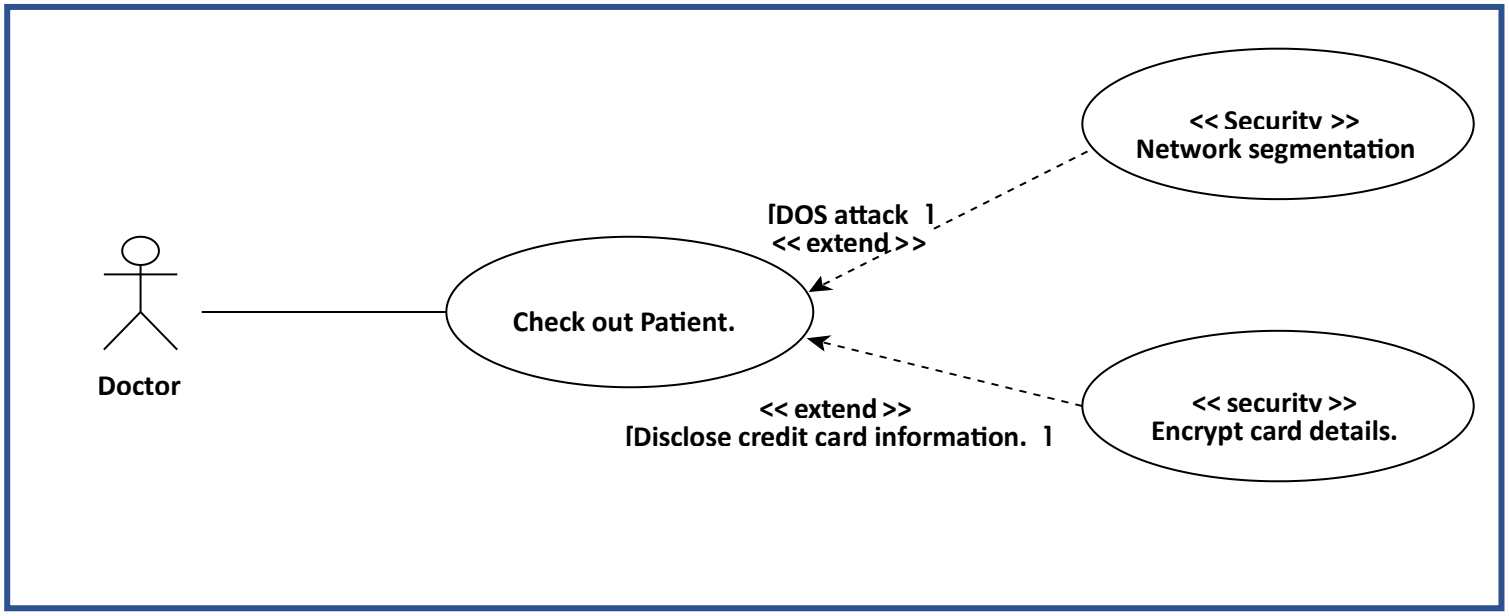
1.Dos Attack:

An attacker may attempt to overwhelm the system by flooding it with a large number of requests, rendering it unable to process genuine payment requests. Critical (100%) - Given the importance of the payment process, attackers are likely to try to disrupt it.

2. Disclose credit card information:

An attacker could capture credit card data being transmitted from the card reader to the system or the bank and use it fraudulently.90% critical - Credit card information is a valuable target for attackers, and it is at risk of being intercepted if the system does not adequately encrypt the data.

Security Use Case:



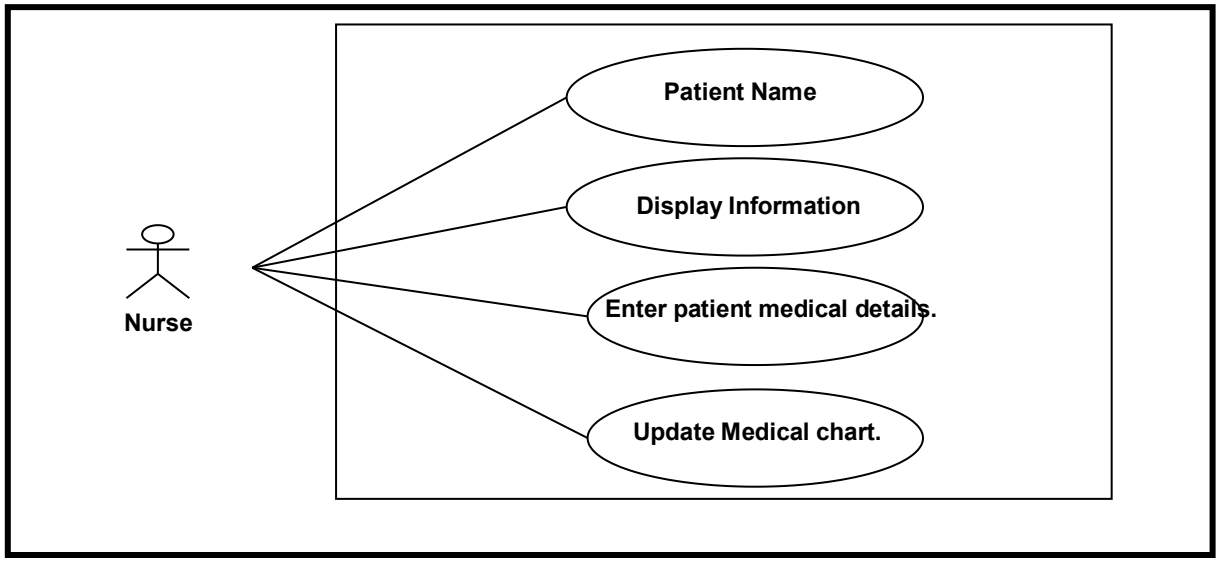
Test case:

Assumption: Patient Name - Author, Medical fee - $450 , Activates Card Reader - Activate, Credit card - 1234, Debit card- 9999, PIN - 5678, Approve payment - Yes, Reference Number - 9876, Update Payment - Done, Print receipt – Print.

T1: Author, $450, Activate, 1234, 9999, 5678, Yes, 9876, Done, Print//Valid

T2: Author, $450, Activate, , 9999, 5678, Yes, 9876, Done, Print//Invalid (Credit card Decline)

Use case 4: Record visit



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1. Cross-Site Scripting Attack	Enter patient madical details (input)	Attacker gains access to the system while updating the information.	lack of input validation and output encoding communication	Critical	90%
2. DoS Attack	Upadte patient medical chart(output)	There is no limit on the number of request that can be made to server which result in loss of medical chart & information.	System failure	Moderate	80%

Assumption:

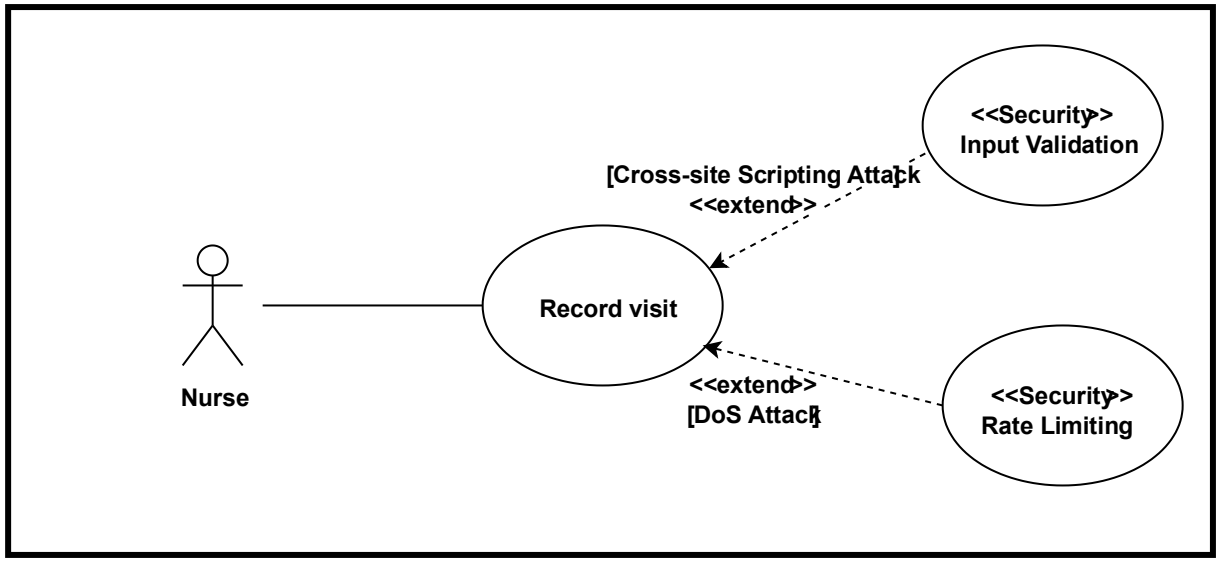
1. cross- site Scripting attack:

An attacker gains access to the system by introducing malicious code into the patient medical data input fields. Taking use of system flaws.90% critical Because the system may lack sufficient input validation and output encoding, attackers may be able to introduce malicious scripts and obtain access to the system while updating patient data.

2. Dos attack:

An attacker floods the system with requests, overloading the server and rendering it unable to react to genuine requests from nurses attempting to update patient charts. The attacker may carry out the attack via a variety of ways, such as botnets or other automated tools, or by exploiting flaws in the network or server infrastructure.60% Moderate The system may lack sufficient mechanisms to limit the number of queries that can be made to the server, resulting in system failure and loss of patient medical chart data.

Security Use Case:



Test cases:

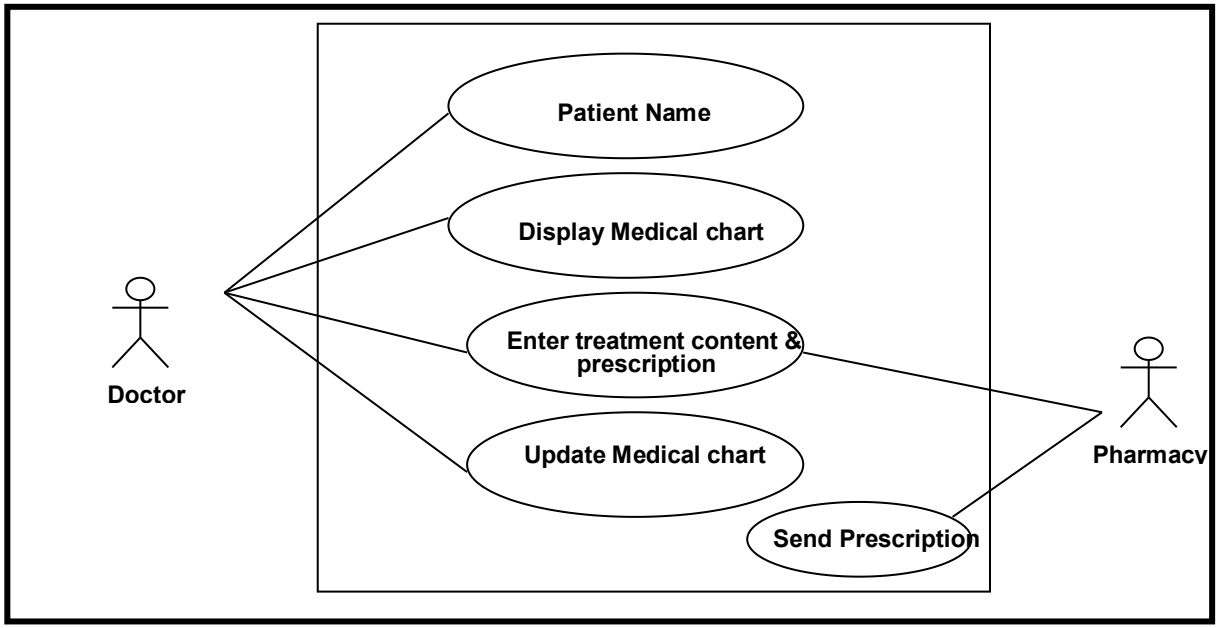
Assumption: Patient Name - Author, Patient Weight – 80Kg, Height 5.7Ft, Blood Pressure -120/80 mm Hg,

Temperature – 100.4 F, Reason – Feaver.

T1: Author, 80Kg, 120/80 mm Hg, 100.4 F, Feaver//Valid

T2: Xyz, 80Kg, 120/80 mm Hg, 100.4 F, Feaver// Invalid patient name

Use case 5: Treat Patient



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1.Tampered	Enter treatment content and Pharmacy(input)	Attacker access to the pharmacy system and modifies prescription	Modify the records and change the settings of the patient account	Moderate	80%
2. Man-in the middle	Send Prescription (Output)	If third party manages to insert themselves between healthcare and pharmancy. Victim can featch the patient perscription information and modify the prescride medice and patient information	Eavesdropping attack	Critical	90%

Assumption:

1.Tampered:

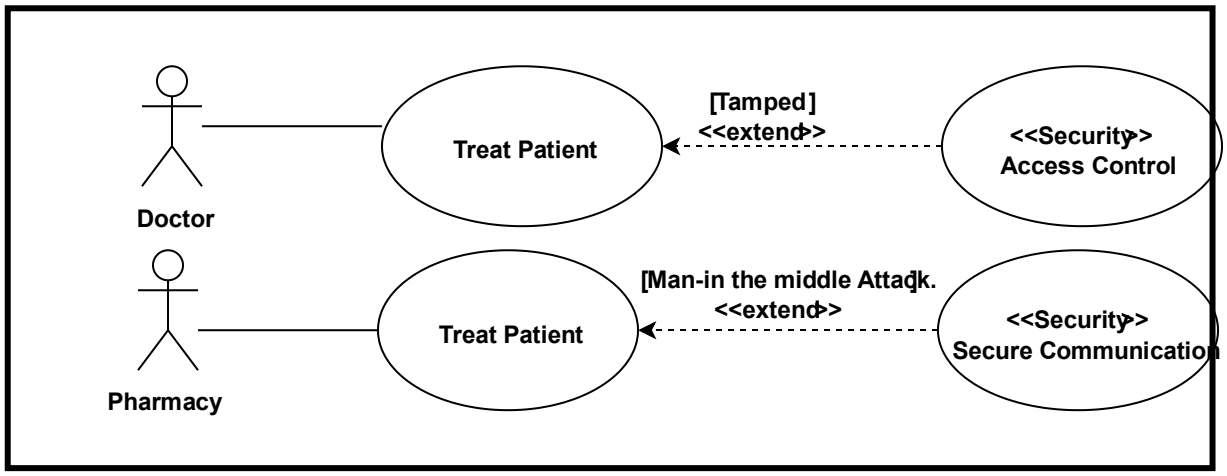
An attacker gains access to the system and adjusts the prescription or the patient account settings, resulting in wrong medicine or dosage. The vulnerability could be caused by a lack of appropriate authentication or access control procedures. Such an attack can have serious effects, including impairment to the patient's health or even death. 70% is

considered moderate. This is because, while different security mechanisms may be in place to prevent unauthorized access, if an attacker manages to circumvent these protections, they may be able to modify the system.

2.Man- in- the- middle:

An attacker intercepts and changes communication between a healthcare provider and a pharmacy, which may contain prescription information or patient information. The vulnerability can develop as a result of insufficient encryption or a lack of effective security measures in the system. The repercussions of such an assault can be disastrous, resulting in improper treatment or the disclosure of vital patient information. The likelihood is high (90%).This is because an attacker can easily intercept unencrypted communication routes, such as email or unsecured Wi-Fi, and modify the prescription without being detected.

Security Use Case:



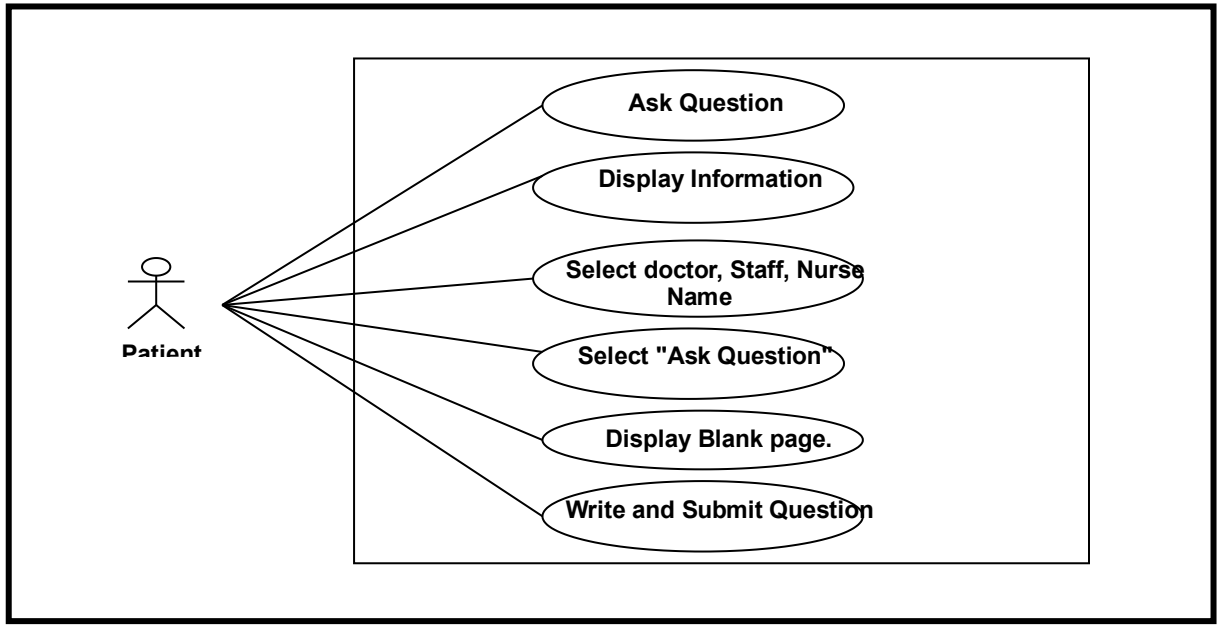
Test Case:

Assumption: Patient Name – Author, Medical Chart(Reason of last visit & medicines) – Feaver & Avril 4mg, Prescription (update if required) – Dolo 650

T1: (Author, Feaver & Avril 4mg, Dolo 650) // Valid

T2: (Author, Stomach Pain & Avril 4mg, Dolo 650) // Invalid Medical chart reason for last visit.

Use case 6: Ask Question



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1.Cross-site scripting Attack	Write and submit question (input)	Attacker can inject malicious code to the blank page in the system web application	It allows users credinational or perform unauthorized action on	Critical	90%
2.Tampered	Write and submit question (input)	Attacker could intercept and tamper with senstive information submitted by patient during the question submission process.	Lack of encryption	Critical	90%

Assumption:

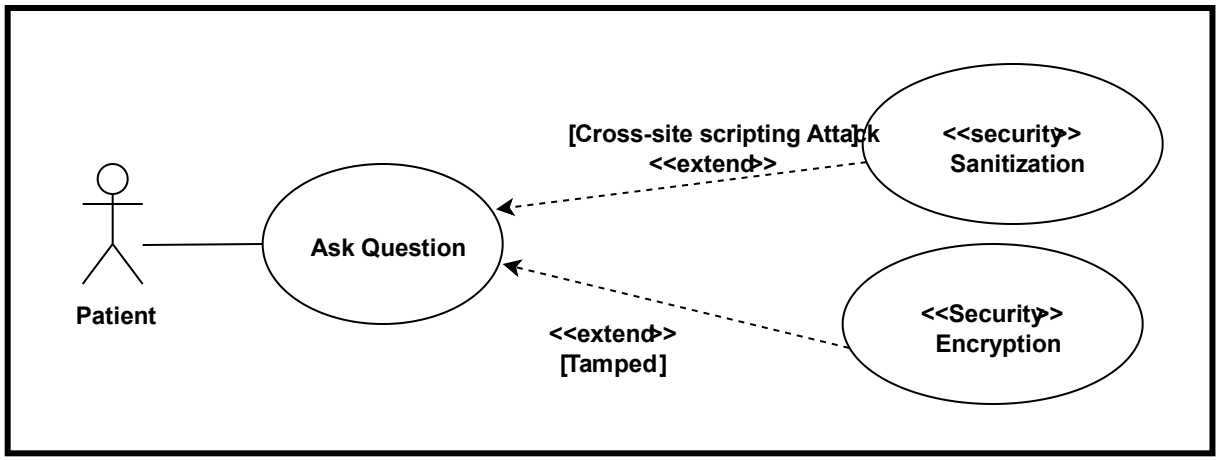
1. Cross- site scripting attack:

The online application does not validate user input adequately, allowing attackers to inject malicious code into the blank page where patients submit their inquiries. Critical 90% Cross-site scripting attacks are a prevalent sort of online application security weakness, and the system may be targeted by attackers looking to exploit this vulnerability.

2. Tampered:

Because the communication between the patient's device and the system is not securely protected, attackers can intercept and manipulate sensitive information given by the patient during the question submission process. 90% is critical Because the system does not use encryption for sensitive information, it is more open to tampering by attackers who intercept and edit the patient's supplied queries. Attackers looking to steal sensitive data or introduce harmful content onto the system may find this appealing.

Security Use Case:



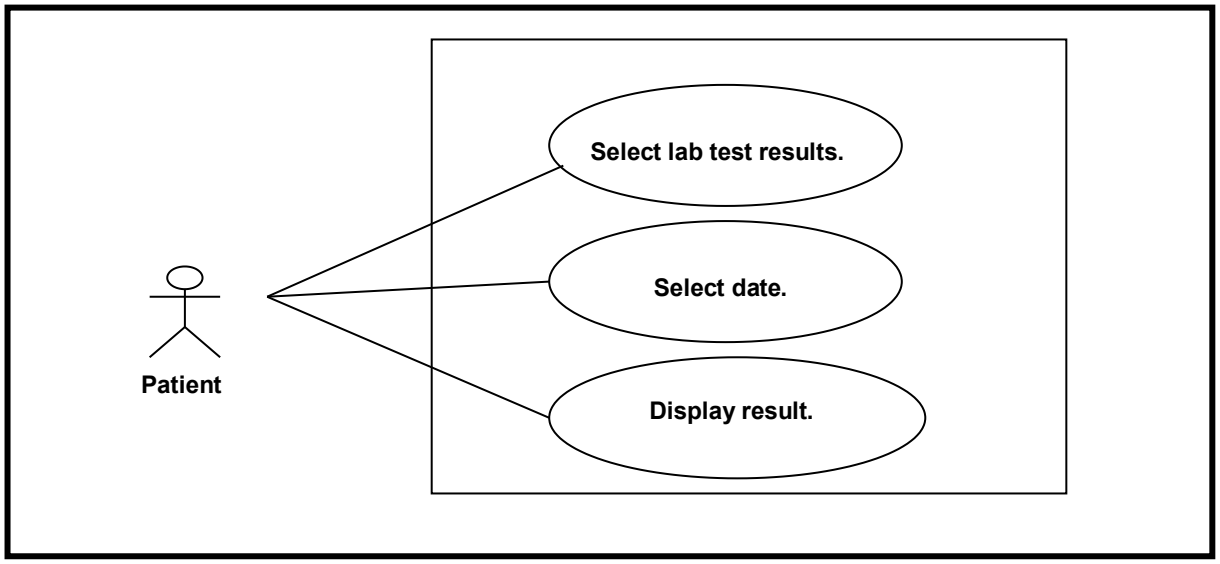
Test Cases:

Assumption: Doctor Name: Dr. Smith, Nurse Name: Tonny, Staff Name: TJ

T1: (Dr. Smith, Tonny, TJ)// Valid

T2: (Dr. Xyz, Tonny, TJ)// Invalid doctor name

Use case 7: View lab test results.



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1.Information disclosure	Display result(output)	Lab test results display to the patient which leads to incorrect diagnoses or treatment	System may not have adequate safety	Critical	90%
2.Session hijackin	Display result(output)	Attackers can Hijack user session by stealing session cookies or using other methods to take over	It allow attackers to impersonate the users and perform action on	Critical	90%

Assumption:

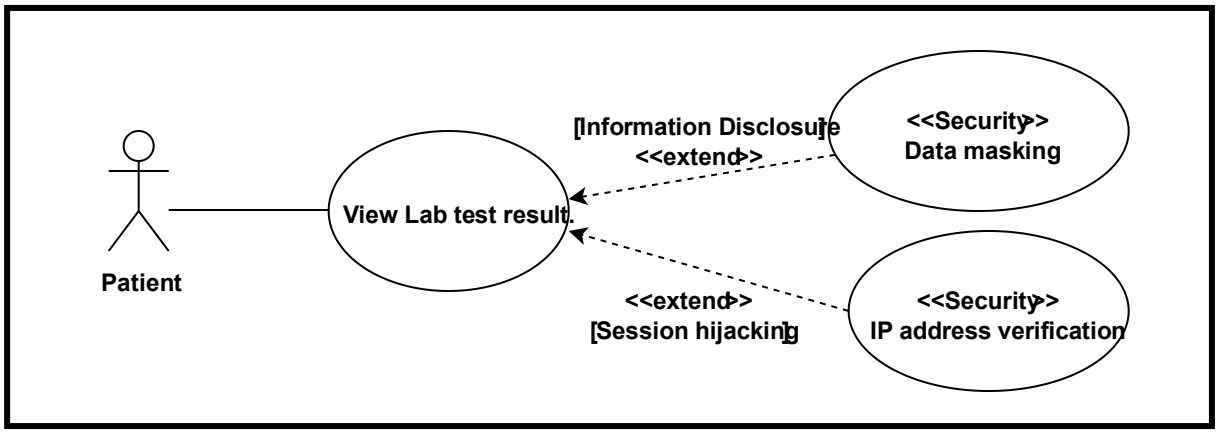
1. Information disclosure:

The system lacks adequate security safeguards to safeguard sensitive patient data such as lab test results. Attackers may gain access to and read the patient's lab test results resulting in inaccurate diagnoses or treatment. Critical (90%), since knowledge exposure is a serious threat with serious consequences for the patient.

2. Session hijacking:

Session management in the system may be exposed to attacks such as stealing session cookies or other techniques of hijacking user sessions. Attackers might take over the patient's session, allowing them to pose as the patient and do acts on their behalf, such as reading lab test results. Critical (90%), since session hijacking is a serious concern that could jeopardize the patient's privacy and security.

Security Use Case:



Test cases:

Assumption: Lab test result - Lab, Dates patient had tested – February 6th 2020 & January 21th 2022 &

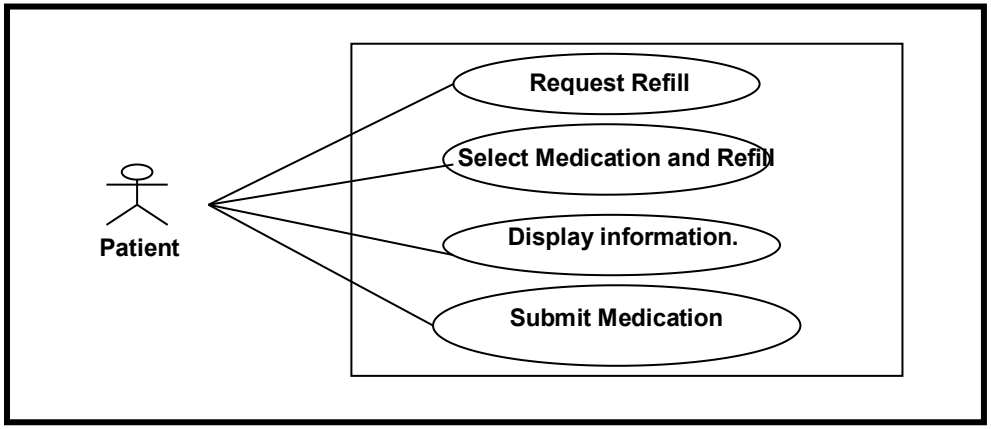
March 1st 2023 & March 26th 2023, Test result – Display.

T1: (Lab, February 6th 2020 & January 21th 2022 & March 1st 2023& March 26th 2023, Display)// Valid

T2: (Lab, December 7th 2019 & January 21th 2022 & March 1st 2023& March 26th 2023, Display)// Invalid

Selected Date to see the results.

Use case 8: Request Refill



Threats:

Threat	Security assets	Description	Vulnerability	Consequences	Likelihood
1.Data leakage	Patient data	If the patient select medication to refill. If it content any senstive information then it may result in data leakage.if the data is not properly protected	Identify theft, loss of medication information	Critical	90%
2.SQL injection	Availability of the system	If the patient select medication to refill and if contains any query or command attacker may be able to malicious code into the filed	Improper inputvalidation, and dynamic queries, poorely configured system	Critical	90%

Assumption:

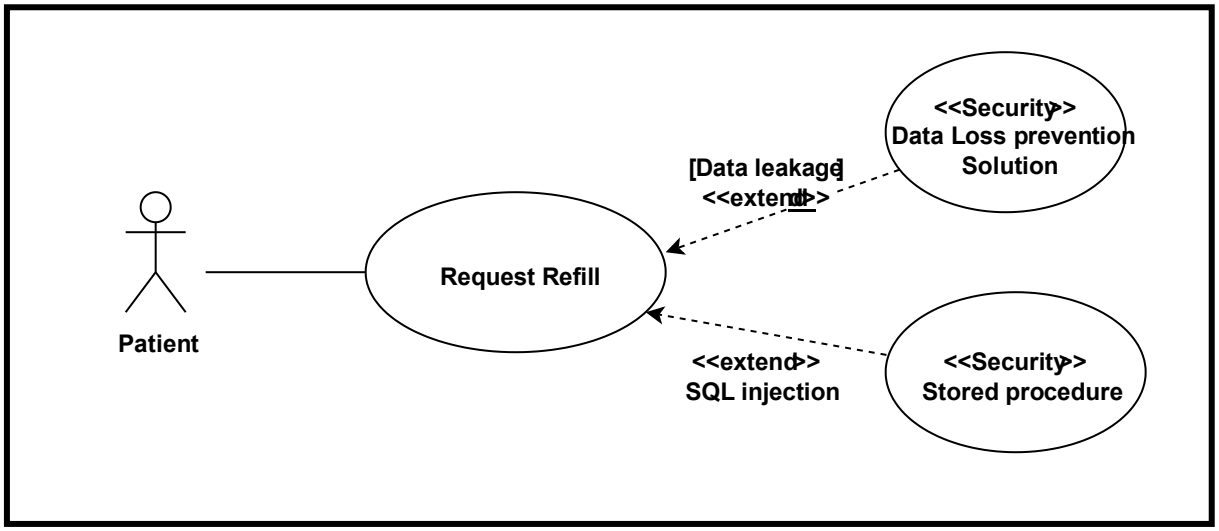
1. Data leakage:

The patient chooses a drug that contains sensitive information, such as personal identifying information. or a medical history, to be refilled. The system lacks sufficient encryption and access controls to safeguard sensitive information from data leakage. Data leaking occurs when an attacker gains access to the system and intercepts the request. Critical 90% This is because if sensitive information about the patient's medicine is not securely protected, it can lead to identity theft and loss of medication. information, both of which can have major ramifications for the patient's health.

2. SQL injection:

An attacker sends a refill request with a malicious SQL query or instruction. Because the system lacks sufficient input validation and dynamic query management, the malicious query is allowed to run. The attacker gains system access and injects malicious SQL code, resulting in a denial of service or other illegal actions. Critical To be refilled, the patient selects a medicine that contains sensitive information, such as personal identifying information or a medical history. The system lacks adequate encryption and access controls to protect sensitive data from data leakage. When an attacker gains access to the system and intercepts the request, data leakage happens. Critical 90% This is due to the fact that if sensitive information about a patient's medication is not securely maintained, It can result in identity theft and the loss of medication information, both of which can have serious consequences for the patient's health.

Security Uses Case:



Test case:

Assumption: Request Refills – Select, Medications – 1. Avril 4mg 2. Dolo 650, Select refill – 1, 2 or (both),

Submit – Yes.

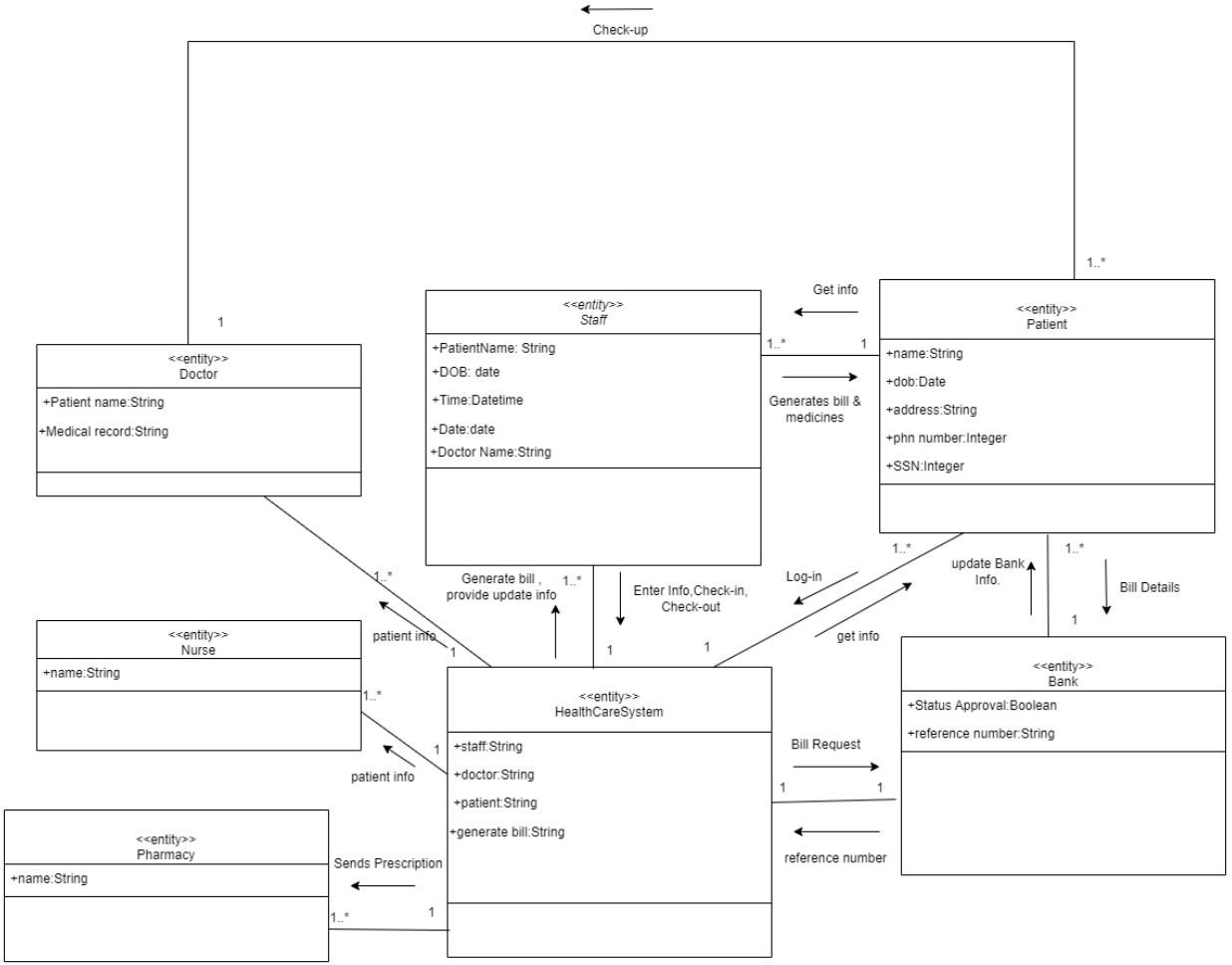
T1: (Select, 1. Avril 4mg 2. Dolo 650, Both, Yes)// Valid

T2: (Select, 3. TusQ D, Both, Yes)//Invalid Medication selected.

Software Security Class Diagram

Develop the class diagram that shows the entity and interface classes for application use cases and security use cases specified in (1) and (2). Indicate security classes among the classes with a stereotype. Apply threat modeling to the class diagram to identify additional threats.

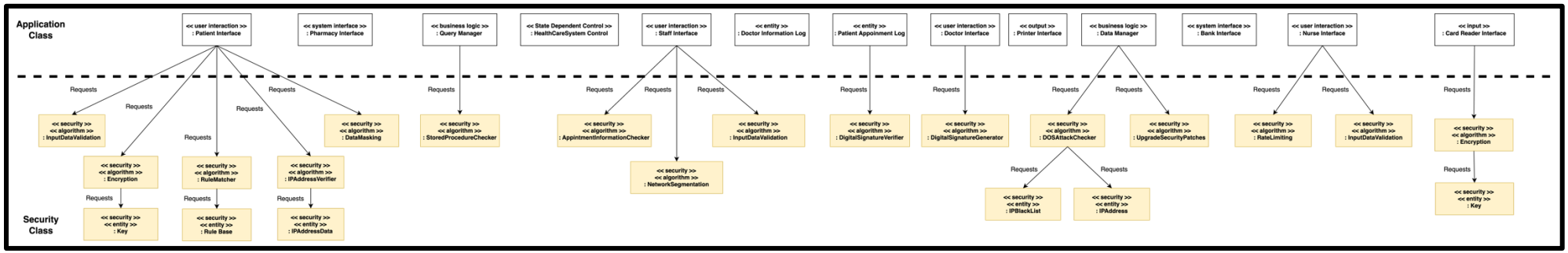
Class Diagram:



Assumption:

1. Here, the healthcare system is viewed as a centralized entity that can be contacted by any other system entities.
2. Staff members have been trained to handle sensitive patient data and appointment details of the patient.
3. Nurses have access to patient medical records and are responsible for updating them after every visit.
4. Doctors have access to patient medical records and are responsible for reviewing and diagnosing patient health issues.
5. Pharmacies have access to patient medication records and are responsible for dispensing medications to patients.
6. Patients have access to their medical records and can request appointments, medication refills, and ask health-related questions.
7. Banks are used for processing patient payments for medical services, and the healthcare system needs to be integrated with the bank's payment gateway.

Application and Security class:



Use case 1: Make Appointment

Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Man in the middle attack	System Patient Information	An attacker can intercept the communication between the system and the staff or patient, allowing them to manipulate or steal sensitive information.	Unsecured network communication, lack of encryption, unverified software	Critical	90%

Assumption:

man-in-the-middle attack in a healthcare system as it is an illegal and unethical activity that can result in serious harm to patients and violate their privacy rights. It is important to ensure that healthcare systems are secure and protected against such attacks to maintain the confidentiality, integrity, and availability of patient data. Instead, it is recommended to focus on implementing strong security measures, such as encryption, access control, and monitoring, to prevent and detect any potential security breaches in healthcare systems.

Use Case 2: Check-in patient

Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Insider threat	Patient information (name, address, phone number, SSN, pharmacy, and health insurance)	A staff member abuses their privileges to access or modify patient information for personal gain or malicious purposes.	Malicious intent, weak access controls	Critical	90%

Assumption:

Insider Threat in a healthcare system as it is an illegal and unethical activity that can result in serious harm to patients and violate their privacy rights. The healthcare system should have strong security measures in place to prevent insider threats, including proper access controls, regular security training for staff, and monitoring of system access and activity. It is also important to conduct thorough background checks and screening of staff before granting them access to sensitive patient data. Fostering a culture of security and accountability can help reduce the risk of insider threats and maintain the integrity and confidentiality of patient data.

Use Case 3: Check out patient

Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Malware Attack	Payment Processing System, Patient's Financial Information	An attacker installs malware on the payment processing system, stealing patient financial information.	Vulnerabilities in the payment processing system, lack of security controls, poor security practices.	Critical	90%

Assumption:

Malware Attack on a healthcare system is that malware could be introduced through an infected email attachment, a malicious website, or a USB device inserted into a computer or server within the healthcare system. Once the malware has infected the system, it could spread to other computers and servers through the network. The malware could be designed to steal sensitive patient data, modify, or delete patient records, or disrupt the operations of the healthcare system. The attackers behind the malware could be motivated by financial gain, political objectives, or personal vendettas. The malware could be designed to evade detection by anti-virus software or other security measures, and it could be difficult to remove once it has infected the system.

Use Case 4: Record Visit

Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Data Loss	Patient Data, Medical Chart	The system loses patient data or medical records due to hardware failure, software bugs, or human error.	Insufficient Backup and Recovery Measures	Moderate	80%

Assumption:

Data Loss healthcare system stores sensitive and confidential patient data electronically, such as medical records, personal information, and payment details. This data may be vulnerable to loss due to technical failures or human errors, such as accidental deletion, hardware failure, or software corruption. Additionally, the system may be targeted by malicious actors seeking to steal or destroy patient data, such as hackers or insider threats. The consequences of data loss in healthcare can be severe, including compromising patient privacy, impeding patient care, and damaging the reputation of healthcare providers. Therefore, it is essential for healthcare systems to implement robust data backup and recovery measures and to prioritize data security to protect against data loss.

Use Case 5: Treat Patient

Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Unauthorized Access	Patient Data, Medical Chart	An attacker gains unauthorized access to the system and obtains patient data or modifies medical records.	Weak Authentication and Access Controls	Critical	100%

Assumption:

Unauthorized Access to the healthcare system is when an attacker gains access to sensitive patient data without proper authorization. This can occur when there are weak passwords, unsecured network connections, or when an authorized user shares their login credentials with an unauthorized user. Once inside the system, the attacker can potentially access and steal confidential patient information such as medical records, personal information, and billing data. The attacker can also modify or delete patient records, causing disruption in patient care, misdiagnosis, and potential harm to the patient. The attacker may also install backdoors or other malicious software to allow them to maintain access to the system in the future. The impact of unauthorized access can be severe and can lead to legal action, reputational damage, and loss of patient trust.

Use Case 6: Ask Questions

Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Phishing Attack	System Security	Patient is directed to a fraudulent website or email, tricked into revealing personal information.	Social Engineering	Critical	100%

Assumption:

Phishing Attack to Healthcare employees receive emails containing phishing links disguised as legitimate requests to access or update patient information. These emails may also contain malware attachments that can infect the healthcare system if opened by employees.

Use Case 7: View Lab Test Results

Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Data interception	Lab Test Results	An attacker intercepts the patient's lab test results while they are being transmitted from the system to the patient's device.	Lack of encryption, Weak network security protocols.	Critical	90%

Assumption:

Data Interception in Health care systems communicates a lot of sensitive and confidential data, including personal information and medical records, between different parties such as doctors, nurses, staff, and patients. Therefore, there is a risk of data interception during transmission, either through unauthorized access to network communication channels or through the exploitation of system vulnerabilities. Attackers can intercept data by using various methods, including man-in-the-middle attacks, packet sniffing, or eavesdropping on wireless networks. The intercepted data can be used for identity theft, financial fraud, or other malicious activities.

Use Case 8: Request Refill

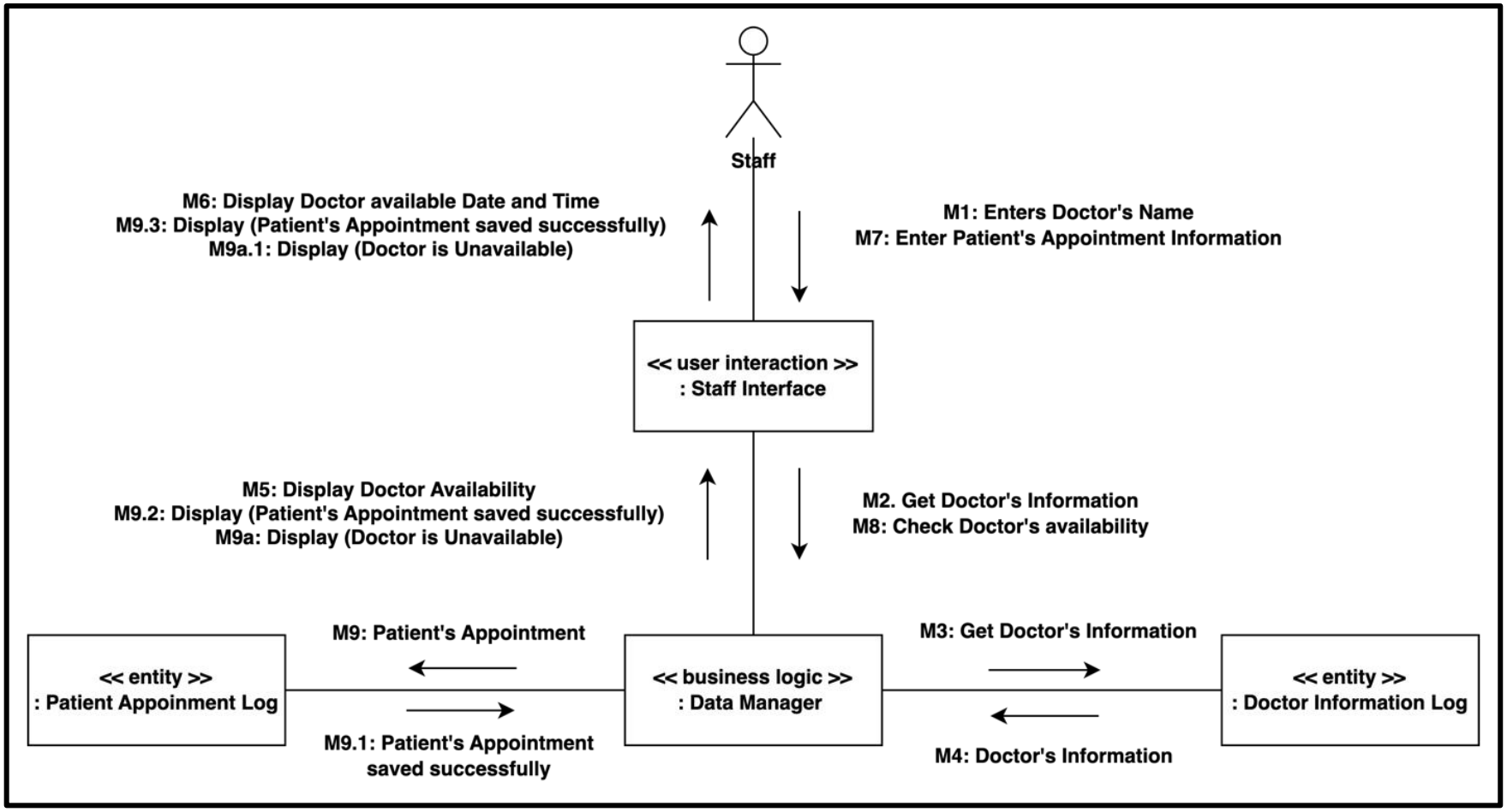
Threat Name	Security Assets	Description	Vulnerability	Consequences	Likelihood
Denial of Service (DoS)	Refill request functionality	An attacker floods the refill request functionality with requests, making it unavailable to patients.	Lack of capacity planning, absence of rate limiting	Critical	90%

Assumption: DOS attack on the healthcare system for the "Request Refill" use case could be that an attacker could overload the system with an excessive amount of refill requests, causing the system to become unavailable to patients or healthcare providers trying to access the system.

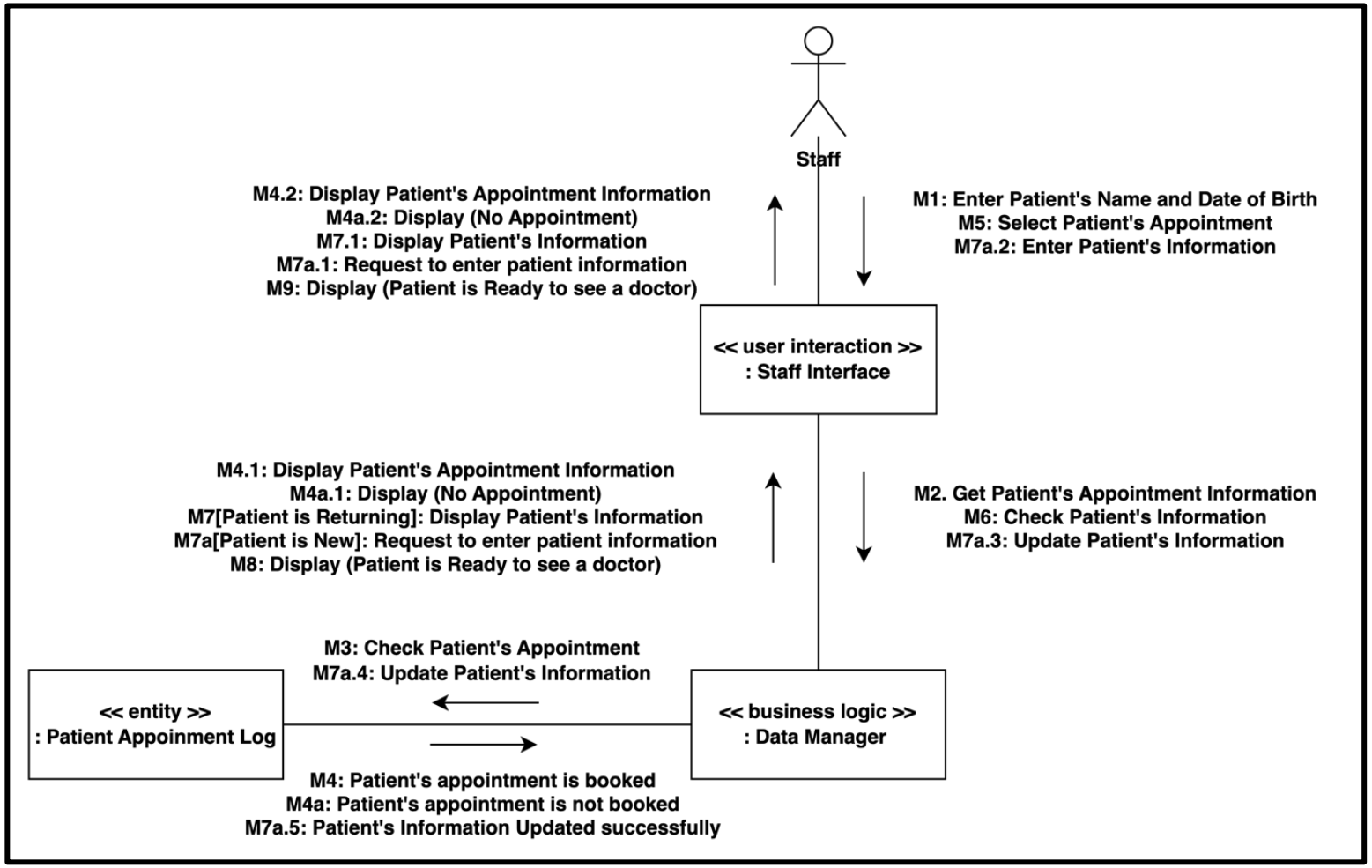
Application and Security Communication diagram

Develop the class diagram that shows the entity and interface classes for application use cases and security use cases specified in (1) and (2). Indicate security classes among the classes with a stereotype. Apply threat modeling to the class diagram to identify additional threats.

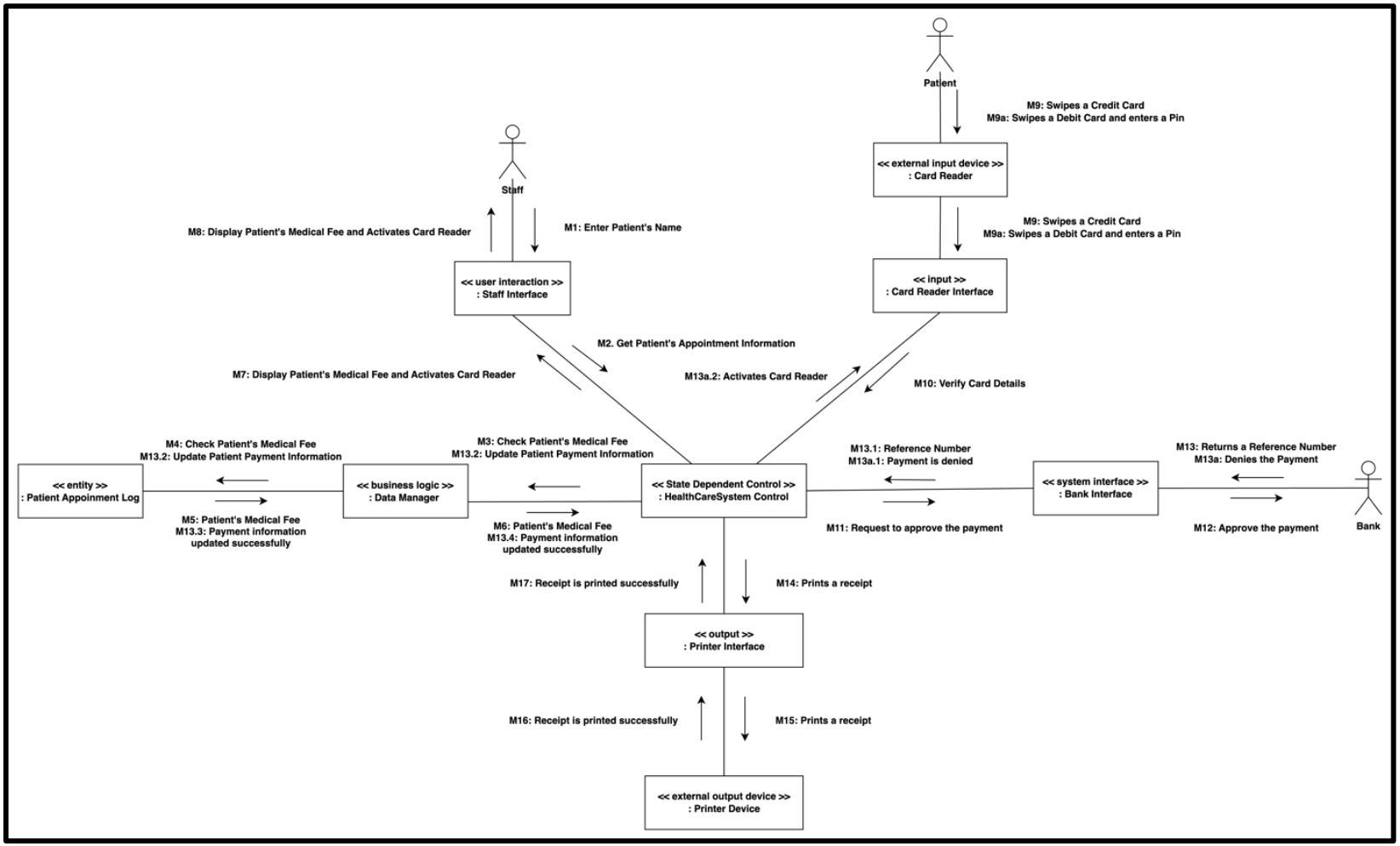
Use Case 1 – Make Appointment



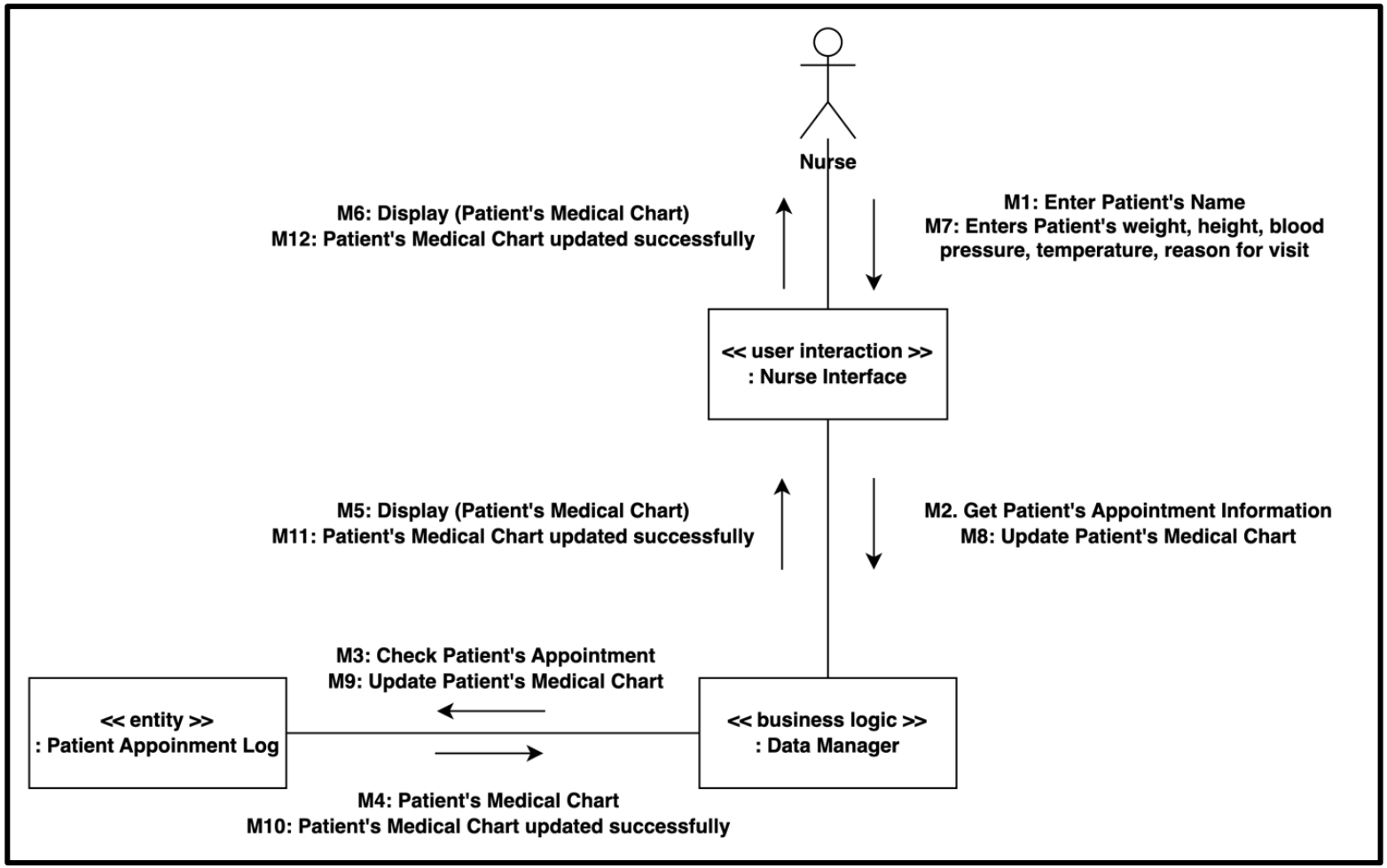
Case 2 – Check-in patient



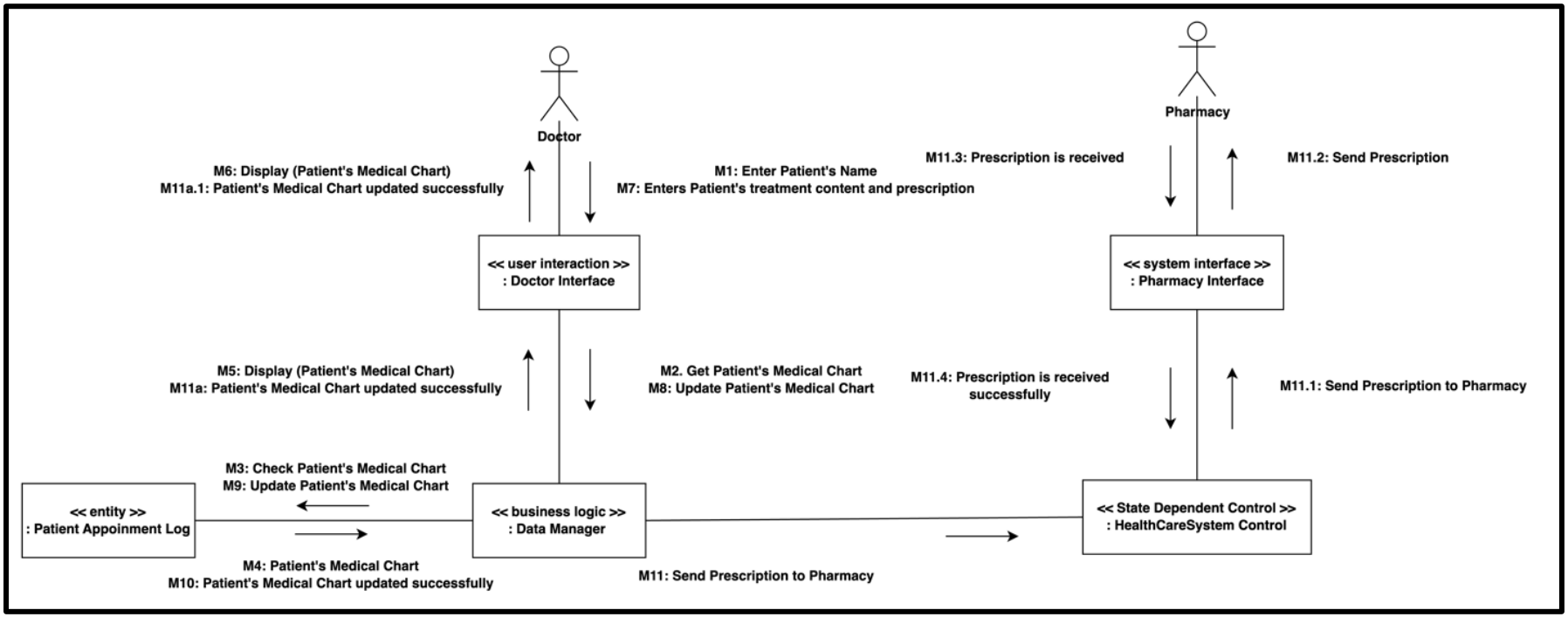
Case 3 – Check out patient



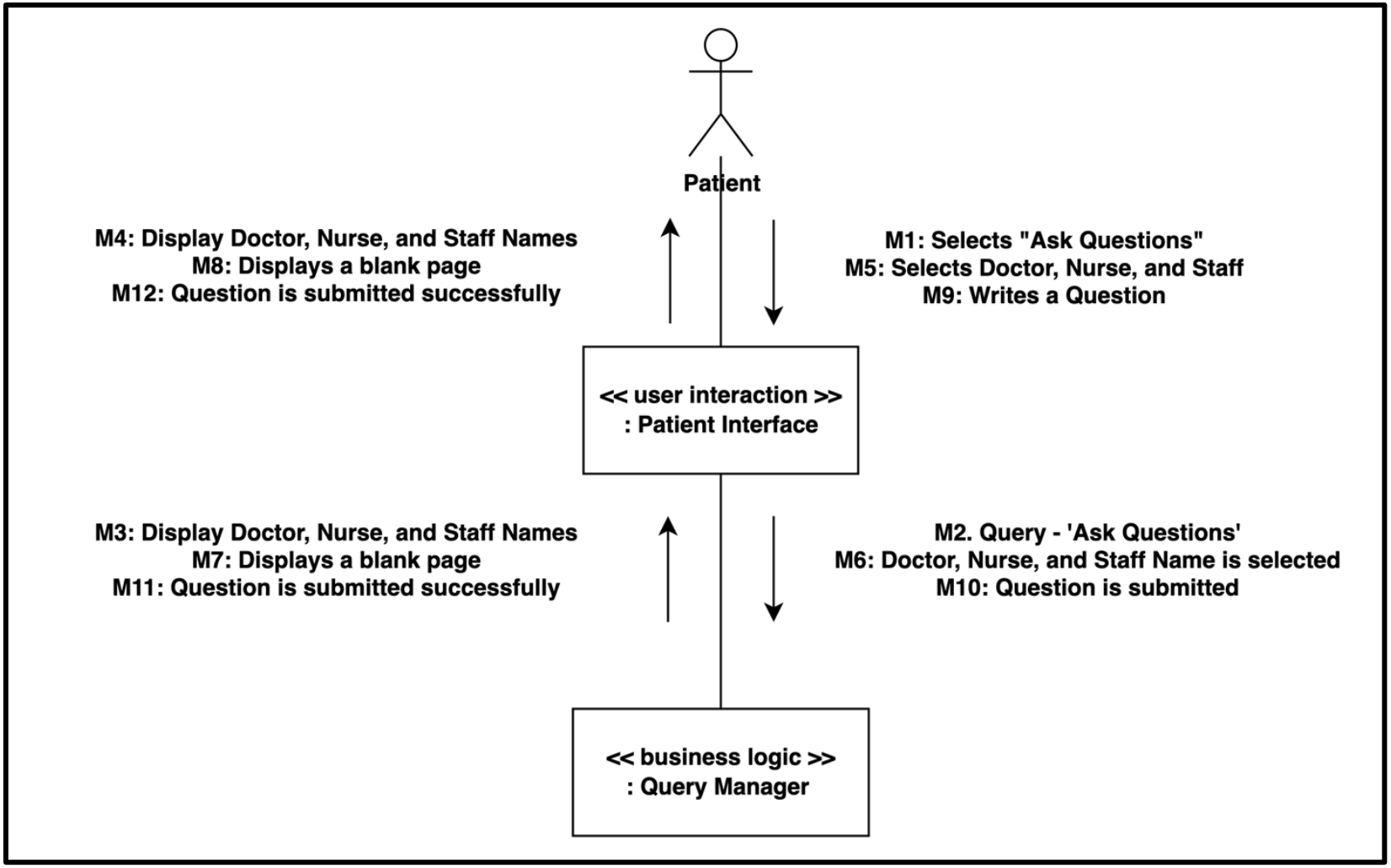
Case 4 – Record Visit



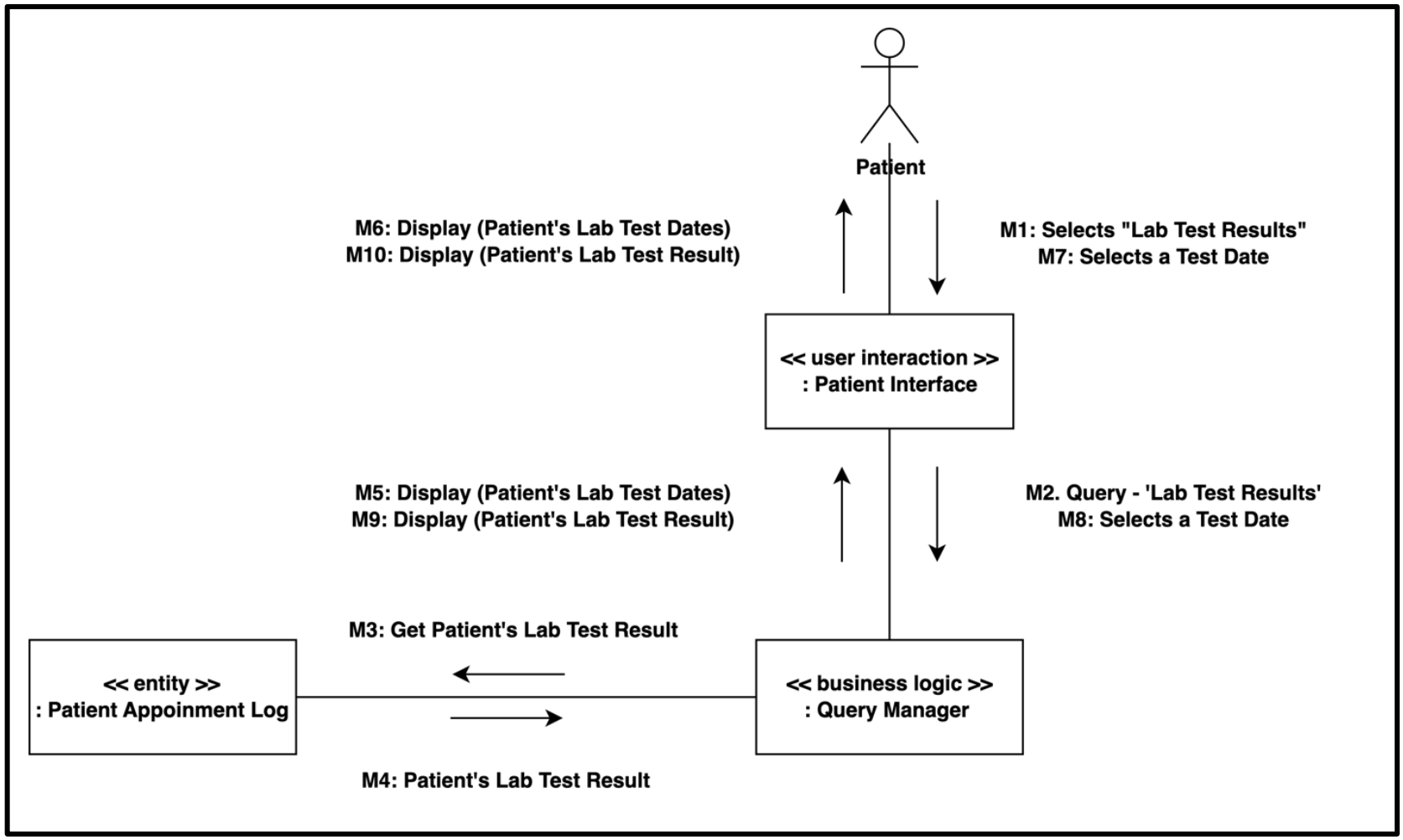
Case 5 – Treat Patient



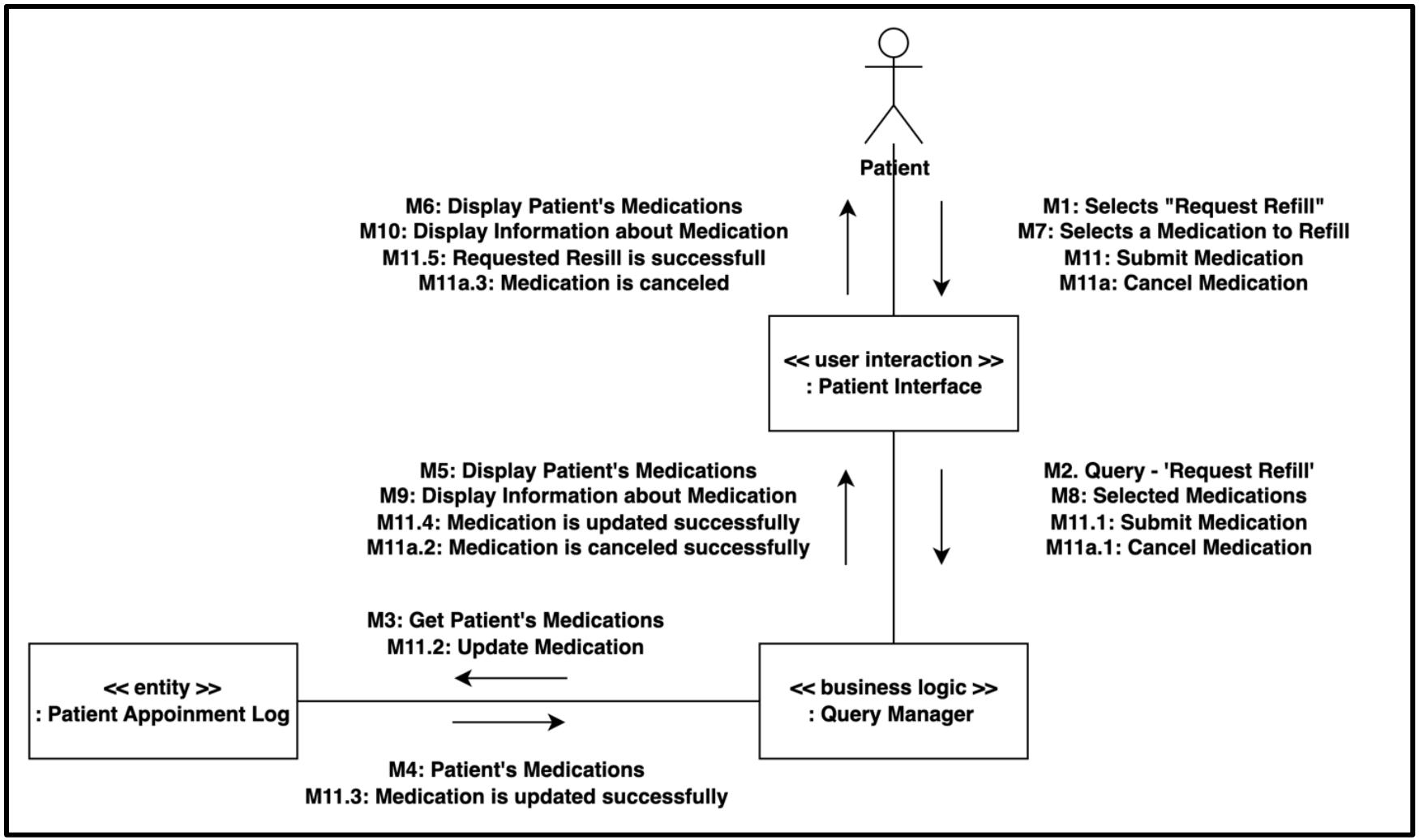
Case 6 – Ask Questions



Case 7 – View Lab Test Results

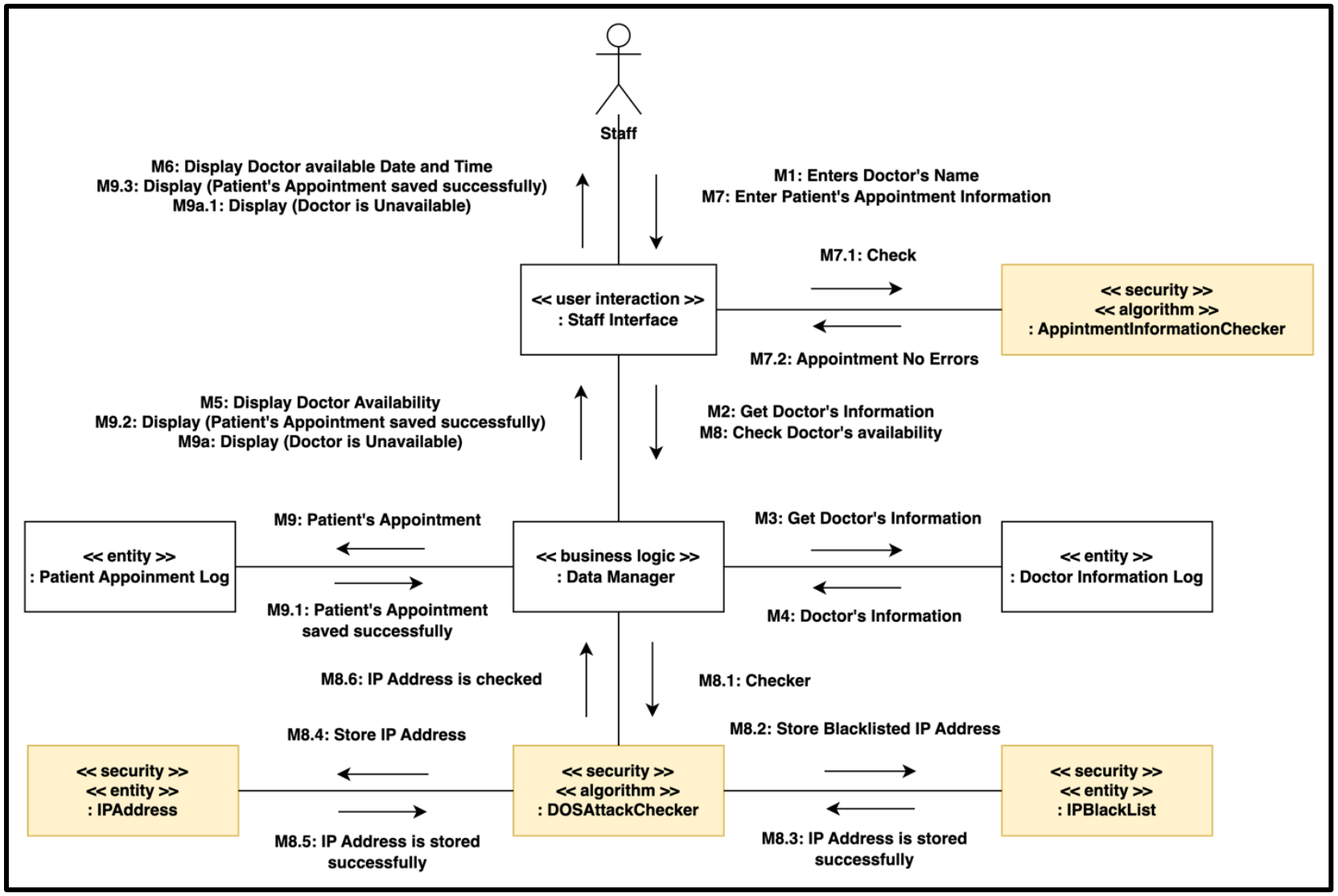


Case 8 – Request Refills



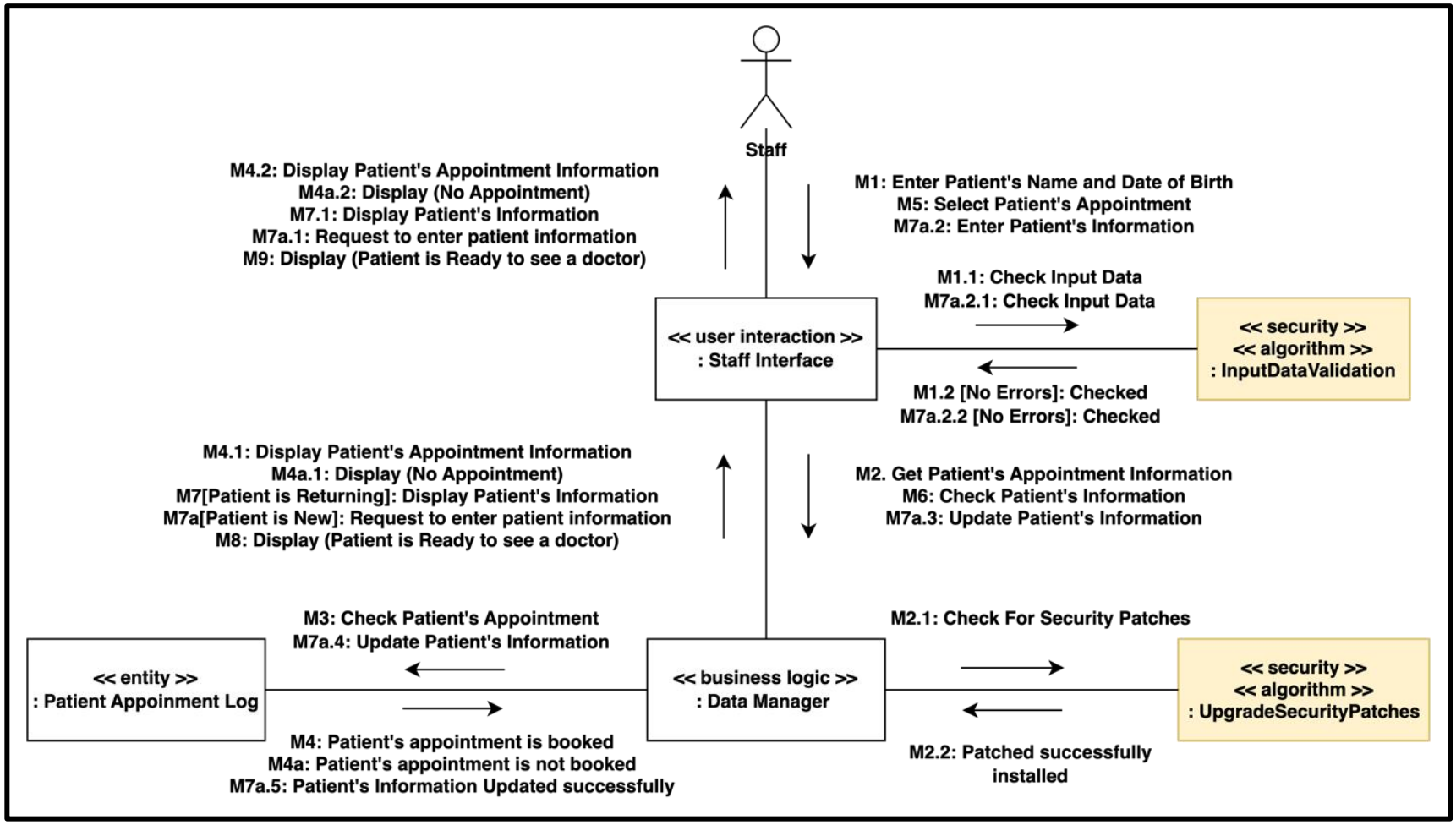
Secure Communication Diagram

Case 1 – Make Appointment



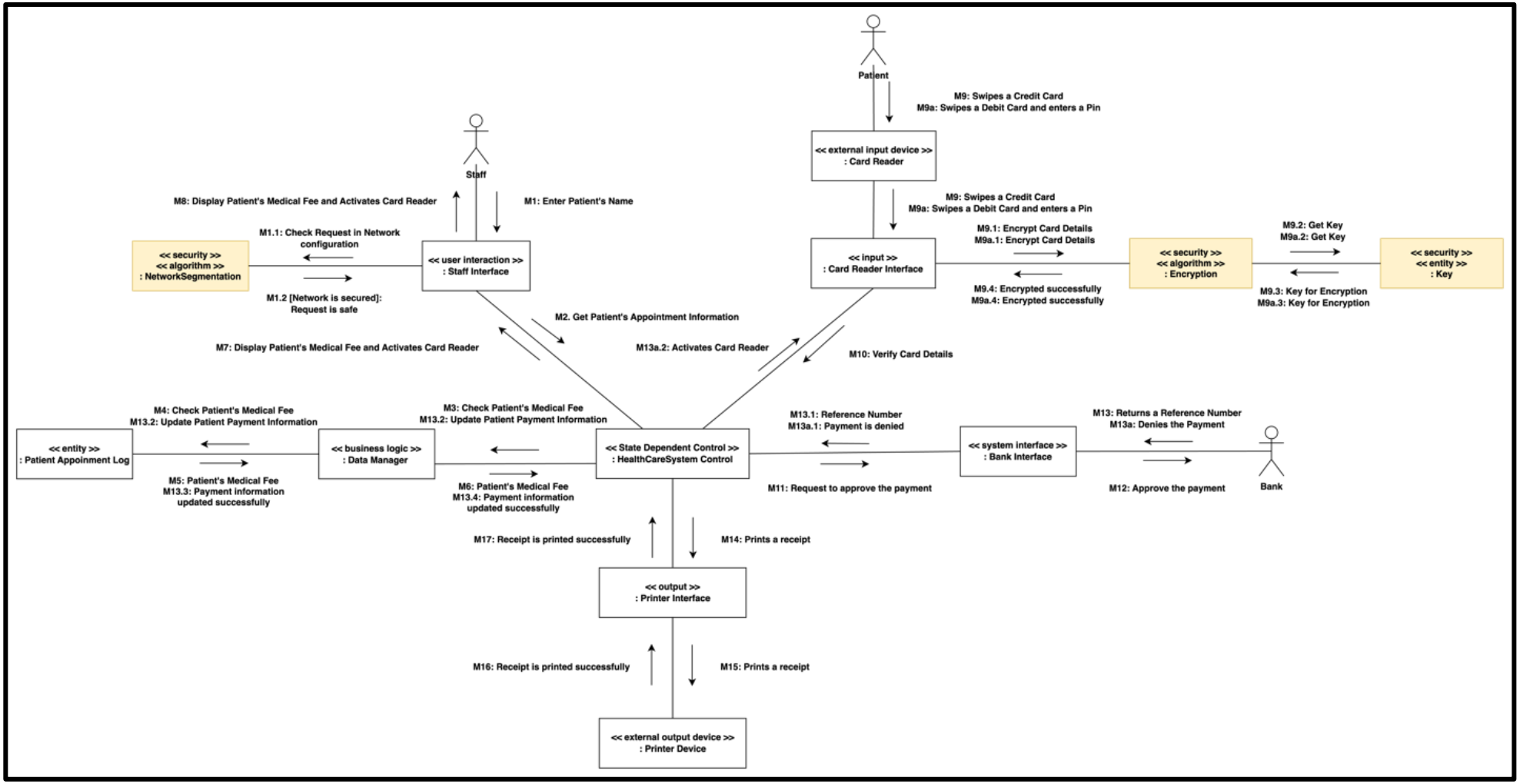
Assumption – We have considered AppointInformationChecker security algorithm on order to avoid Information disclosure and we have applied DOSAttackChecker algorithm to reduce DOS attacks.

Case 2 – Check-in Patient



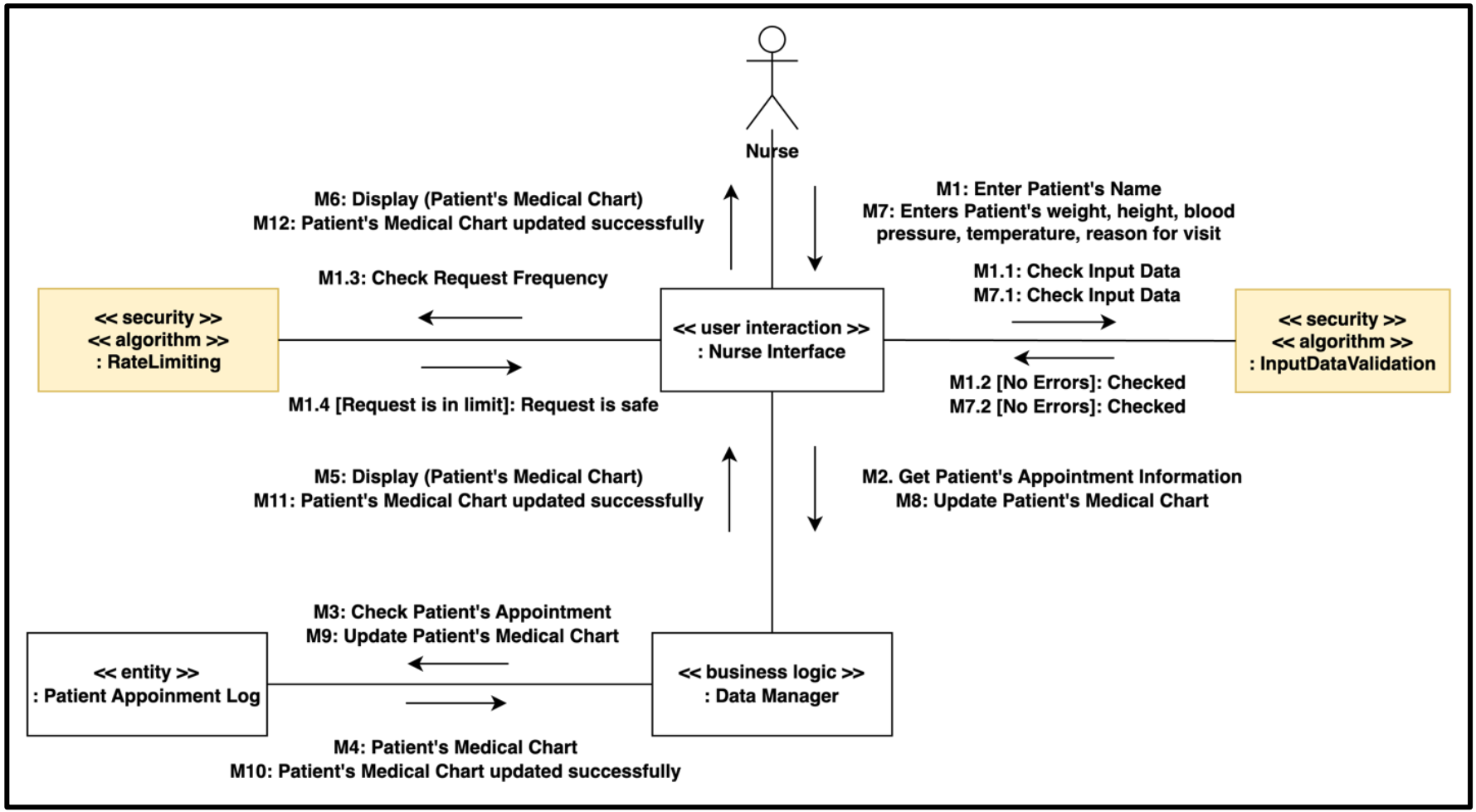
Assumption – To avoid SQLInjection attack, we need to validate each input which is entered by user. So, for this we have considered InputDataValidation sercure algorithm in order to validate user input data. If we update our system on regular basis, then there might some chance to prevent our system from Keylogging attacks.

Case 3 – Check out patient



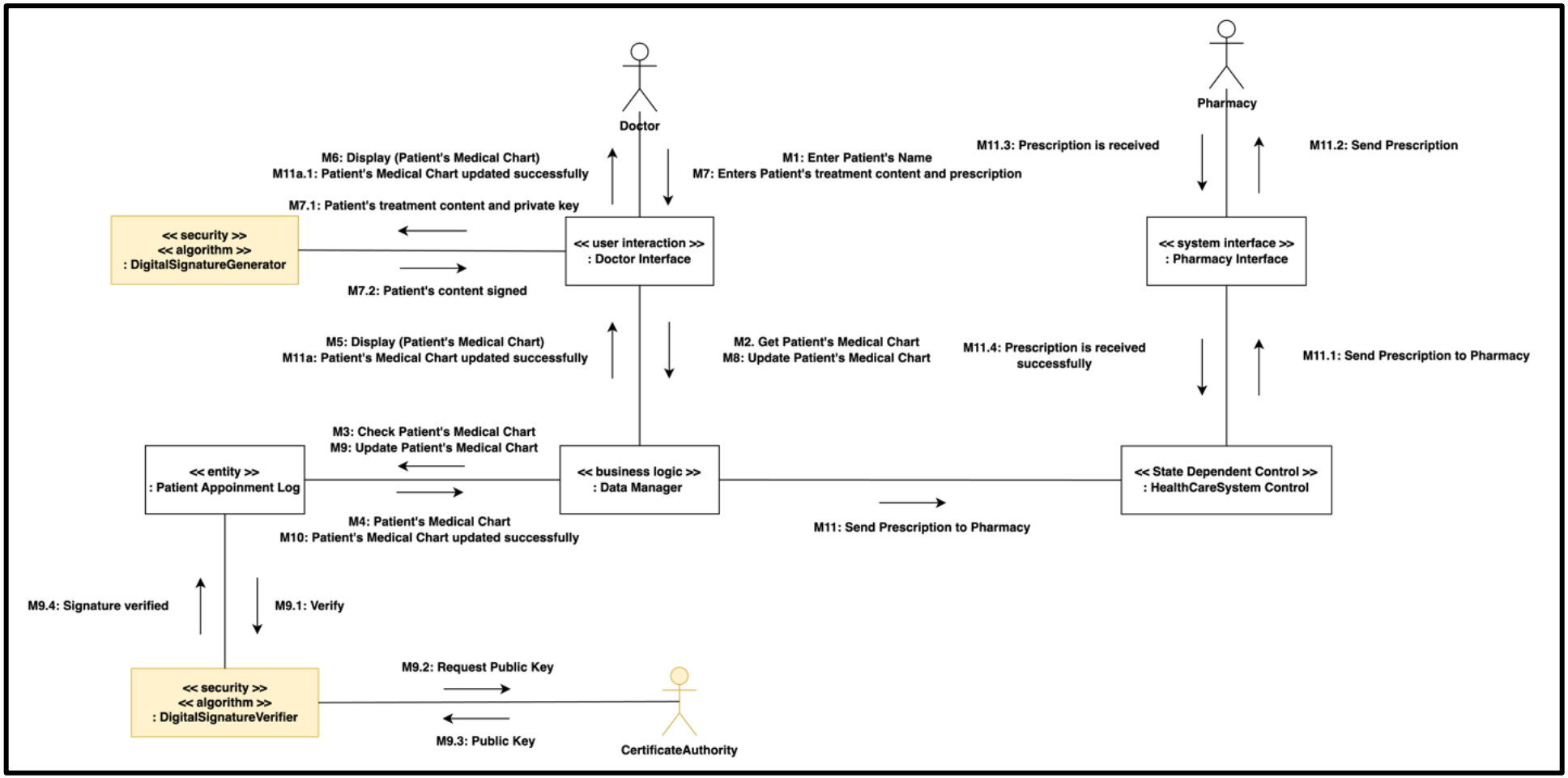
Assumption – To prevent data disclosure of credit/debit card, there is one best practice, encrypt your card details and then share that encrypted value to bank for further verification. And, to avoid DOS attack, we have considered, NetworkSegmentation algorithm which helps us to divide our network in multiple sub-networks where we can easily control request flow and can control DOS Attack

Case 4 – Record Visit



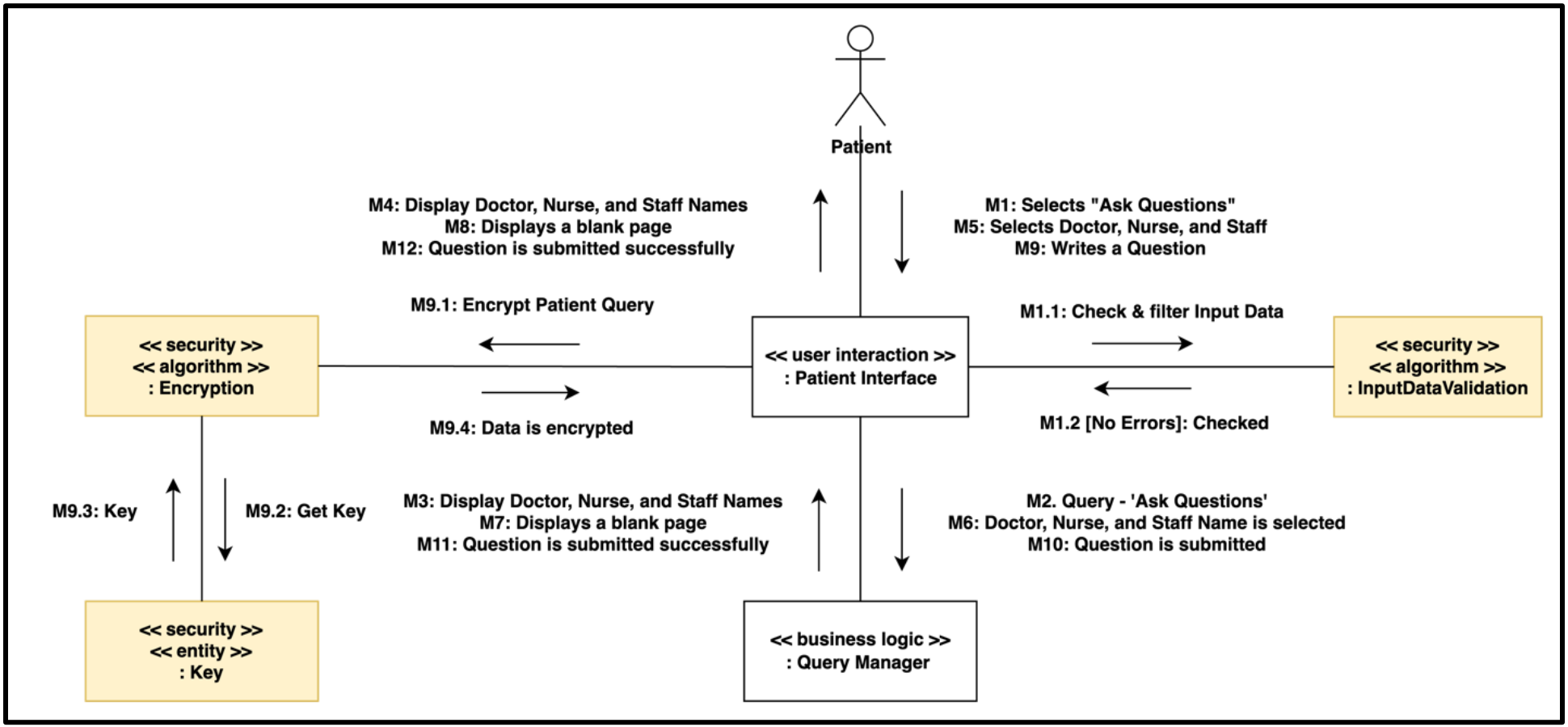
Assumption – To avoid Data Tampering, we are validating user data to avoid any scripts or unwanted code, we are validating use inputs and applied ‘Ratelimiting’ algorithm which helps us to control and helps us to track on valid user request which prevents our system from various attacks.

Case 5 – Treat patient



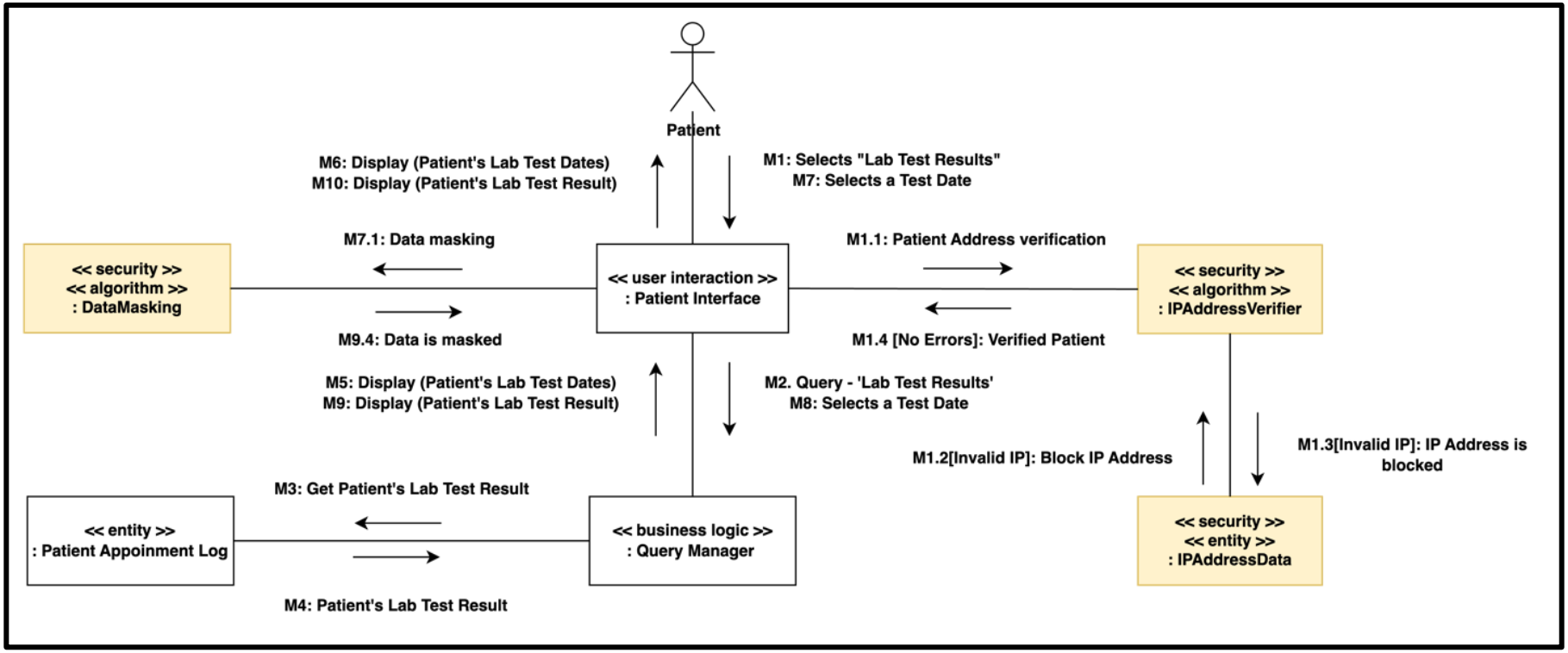
Assumption – In order to check user identity, we have used DigitalSignatureGenerator and DigitalSignatureVerifier algorithm.

Case 6 – Ask Questions



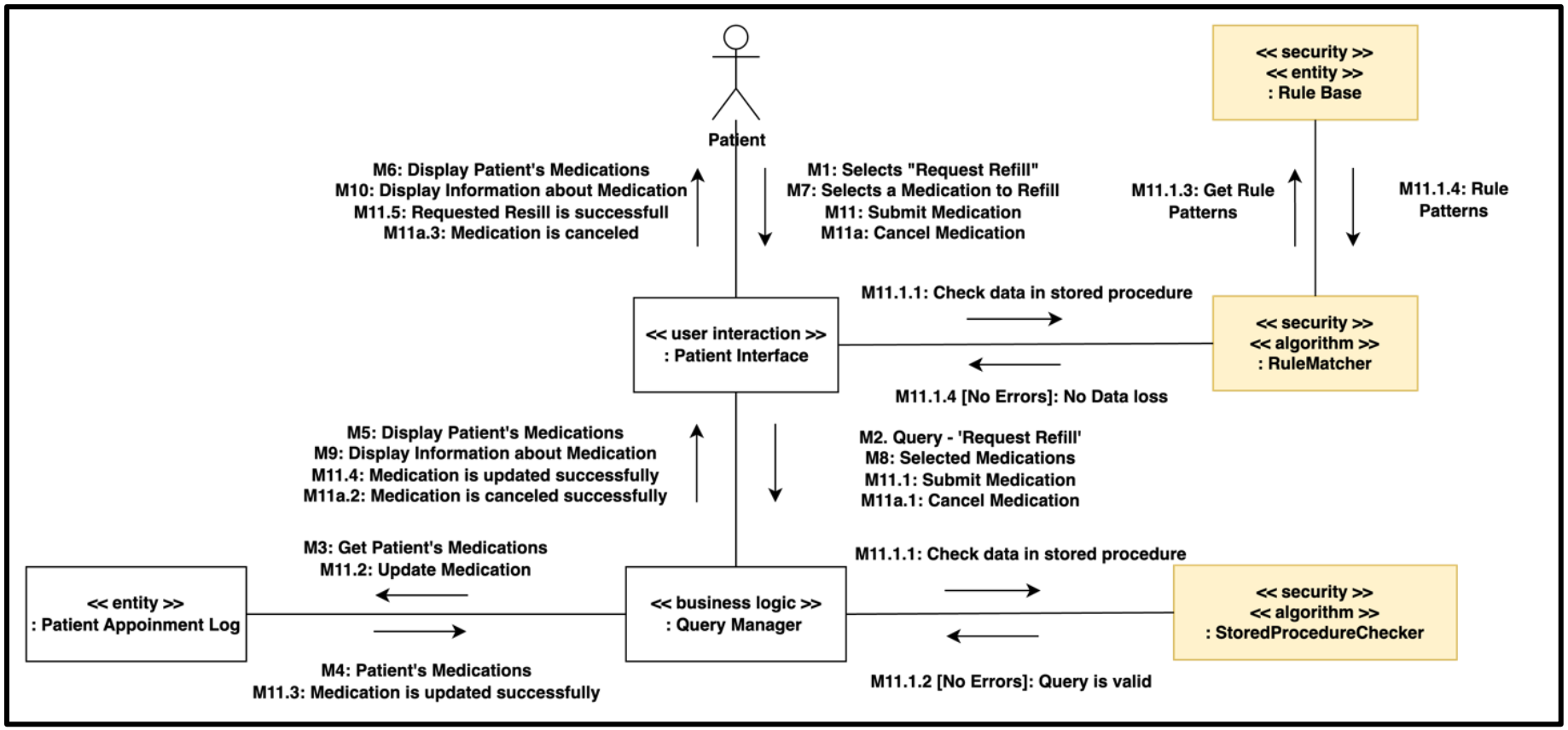
Assumption – We have used Encryption technique to store patient's inputs in secure manner. And, at the same time, we are validating user input to prevent possible SQL Injection attacks.

Case 7 – View Lab Test Results



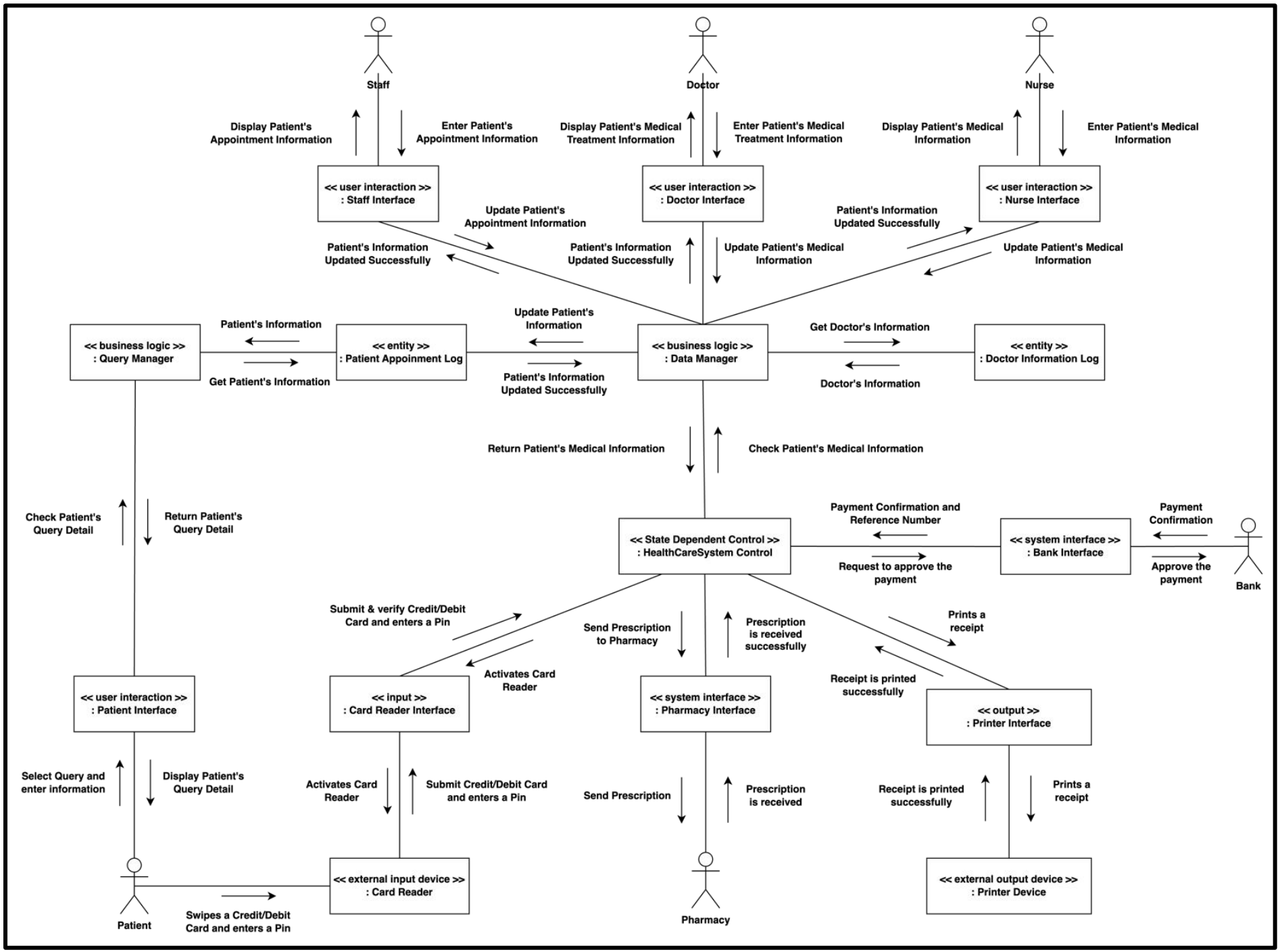
Assumption – We are applying a Datemasking technique to prevent user's sensitive information. We have used 'IPAddressverifier' to validate user's identity.

Case 8 – Request Refill

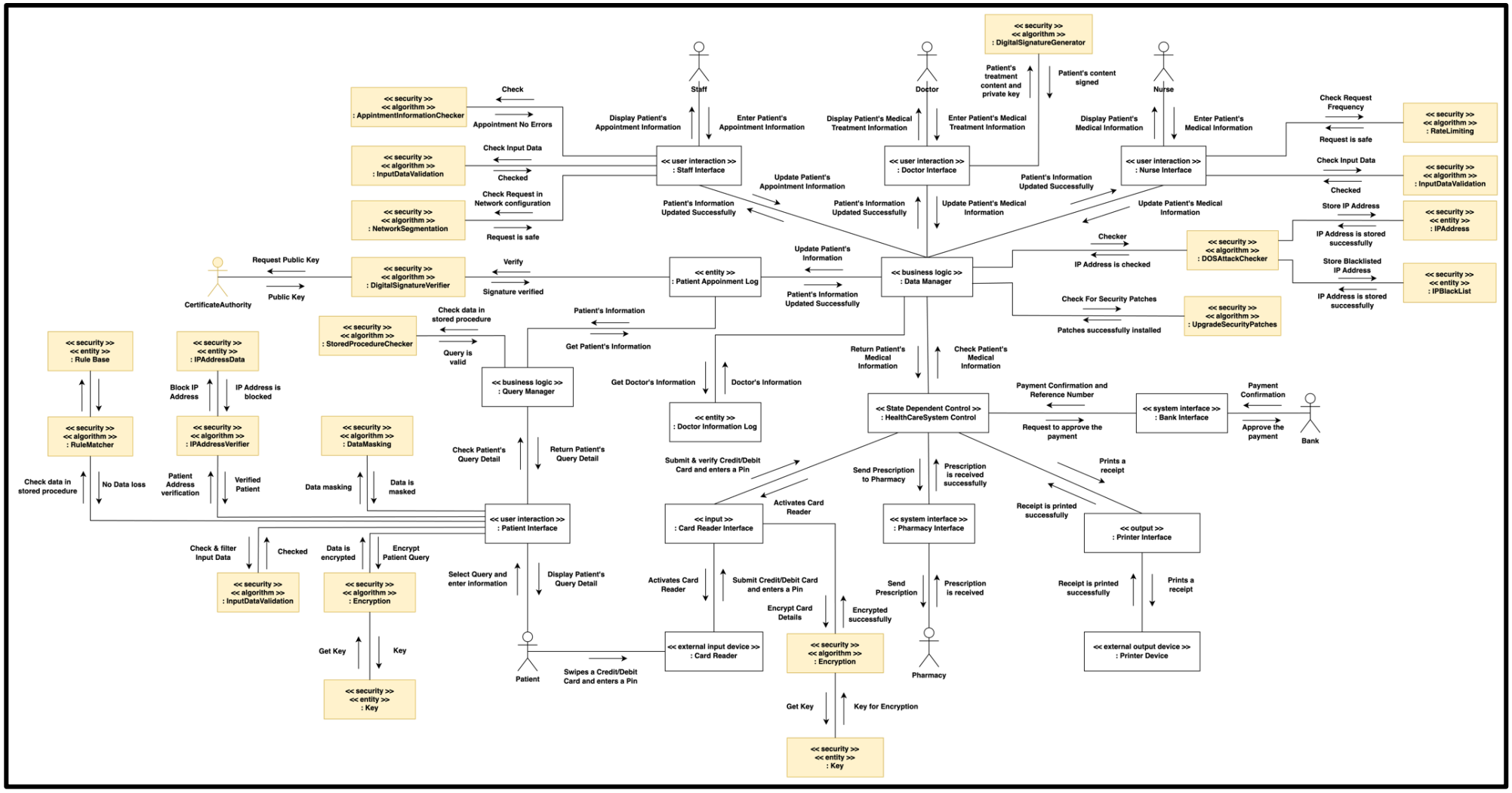


Assumption – We have considered 'RuleMatcher' algorithm to validate user's input with our system's predefined datasets which user is expected to enter. If user is entering any other information, then we are rejecting his/her request.

Integrated Communication Diagram



Integrated Secure communication Diagram

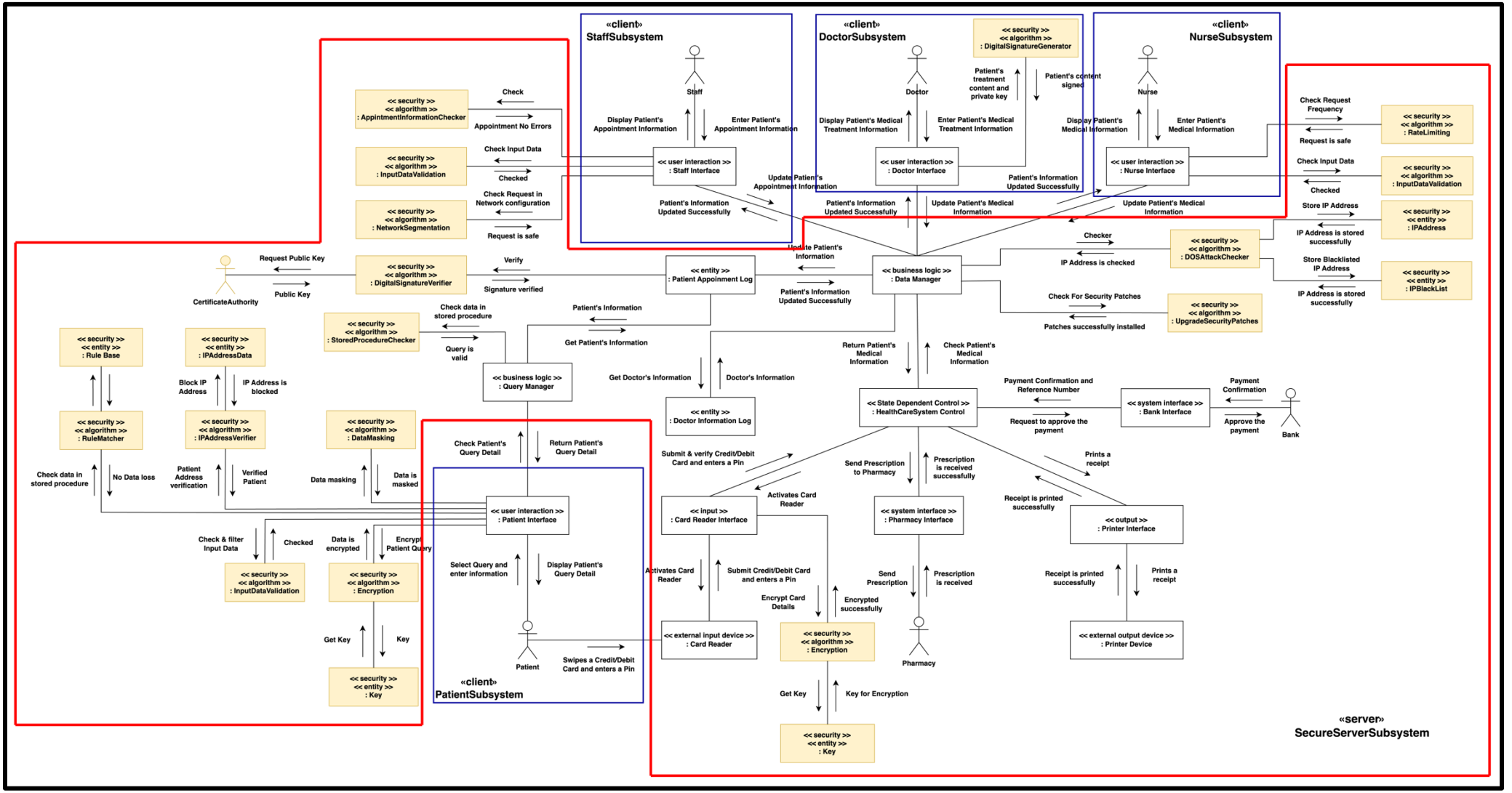


Assumption –

We have segregated our system in 3 parts – one top side, you can see all hospital workers and their respected security algorithm and application classes. On left side, you can see all application and secure classes related to patient. And right-bottom side, you can see all external system/outputs like card reader, bank, pharmacy and printer along with their security classes.

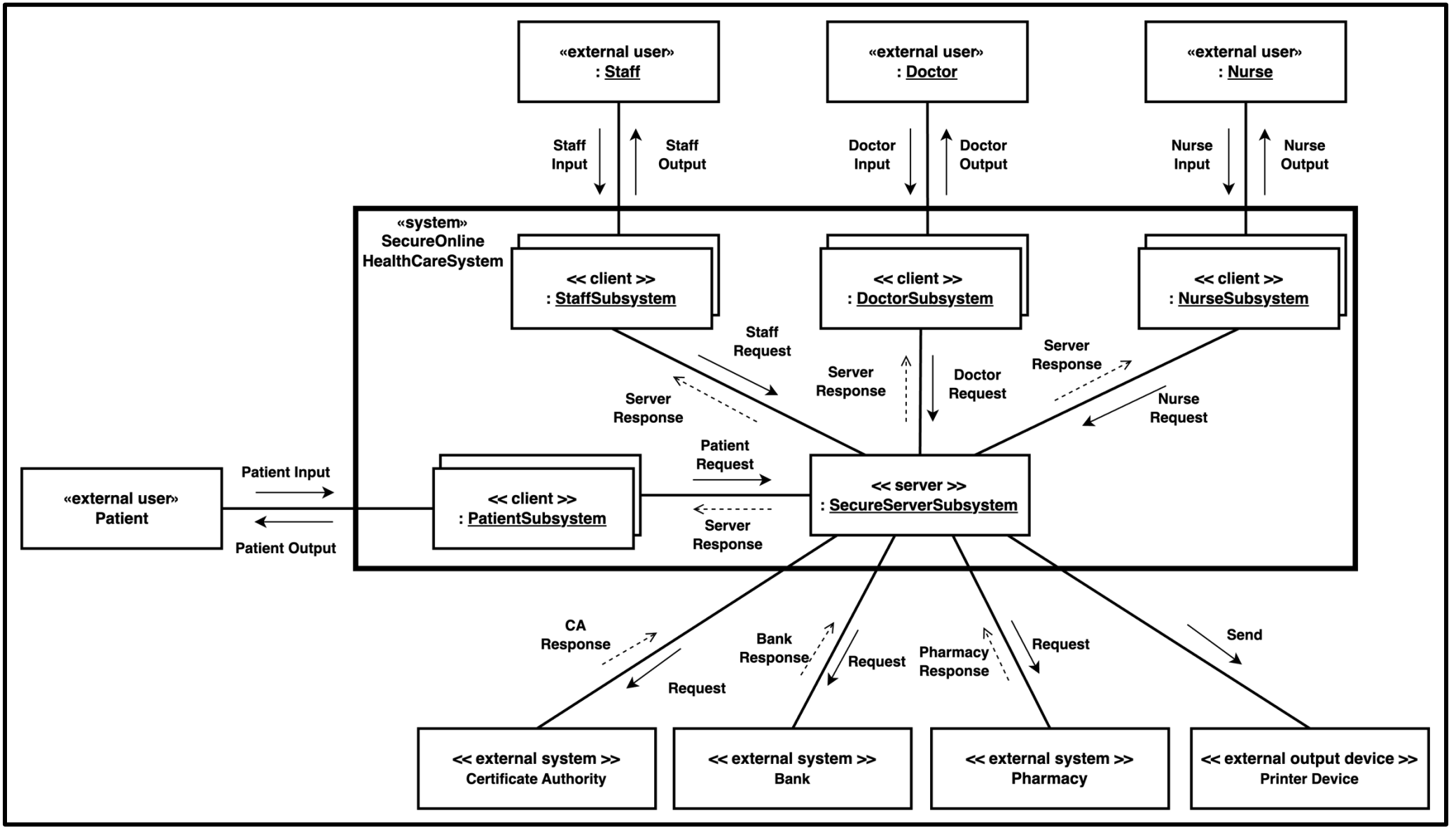
Software Architecture

Client and Server subsystem in integrated communication diagram



Assumption – We have divided our architecture in 5 parts- 4 client architecture (highlighted with blue color) and 1 server architecture (highlighted with red color)

Client-Server Architecture



Assumption – We have shown different client and server subsystem with their connection and request-response message along with their system boundaries. We have covered almost all security issues and their mitigation in earlier question, so we didn't have any strong security issue to add in this question. So, we have just considered only Software architecture diagram without any additional threat mitigation.