CS 3354 Software Engineering
Final Project Deliverable 2

NexMed

Team: Code Byters
Alex Chan-Nui,
Cory Harris, Sophia Kobzar,
Nahome Abraham, & Jeremy Pacheco

Delegation of tasks of project deliverable 2:

- Alex Chan-Nui
  - Question 5
  - Question 7
  - Question 8
- Cory Harris
  - Question 3
  - Question 8
- Jeremy Pacheco
  - Question 3
  - Question 8
- Nahome Abraham
  - Question 6
  - Question 8
- Sophia Kobzar
  - Question 1
  - Question 2
  - Question 4
  - Question 8
  - Question 10

Project Deliverable 1 content

CS 3354 Software Engineering
Final Project Deliverable 1

NexMed

Team: Code Byters
Alex Chan-Nui,
Cory Harris, Sophia Kobzar,
Nahome Abraham, & Jeremy Pacheco

NexMed is an app for doctors to use so patients can see their health records and for other doctors of the patient can see the records. Patients can also see their records if they need to; a patient could have forgotten when they needed to take their medicine.

- Long story short (Sophia's story)
    - My sister at an early age got a vaccine while she was taking medicine for arthritis. Because she got the vaccine, it stopped the treatment for her arthritis from working. Suppose there was better communication between my sister's doctors. My sister will not have to change to an arthritis medicine that is a lot more harmful to her body; that is why she starts with the first one in the first place. Thankfully now my sister is ok, and her arthritis is in remission.
- For a doctor to be connected to a patient
    - The doctor will send a patient a request to be added to their health team and then the patient will go into the email to accept it.
- Doctors will see anything medical that happened with the patient.
    - Vaccines, surgeries, or medication
- The doctor can only add the vaccines, surgeries, and medication etc.
    - If the patient finds an issue with their record (example: wrong medication) they should contact the doctor that was involved to change it.
- Patients can add notes if they wish.
    - Examples: Medication that did work or did not and surgery notes that patient wants to write.
- If the doctor becomes a patient to another doctor how to manage that?
    - The system will ask if the doctor need the doctor's interface or patient's
- If police, ask for records how to handle it?
    - If they have a warrant they can get the records of the patient. If they need it, they need to show it is needed. Find what we need to do to keep the records safe and what laws we must follow (HIPAA)
- Patients should be able to pay their bills on the NexMed website/app
- Patients will be able to message their doctor and vice versa on NexMed
- tasks delegated.
    - How will the interface look like for patients, doctors, and parents of minors? - Sophia Kobzar
        - Make the interface user friendly for all users.
        - Find good color schemes and models.
    - How would the setup be for doctor's accounts and patient's accounts? - Alex Chan-Nui
        - Patients can sign up like any other service. Doctors need to be check before account is live (so random people cannot have a doctors account)
    - Customer service – Nahome Abraham
        - What common issue could be possible that we could add to a help page (an FQA)
    - Patient's insurance needs to be connected to the account to help them find a doctor in their network. - Jeremy Pacheco
        - How will the billing look like?
        - How will the patient pay the doctor and how the doctor gets their money
    - Making our app able to connect with smart devices so patients and doctor can be notified when something is wrong - Cory Harris
        - Example: blood sugar levels, blood pressure levels etc.
        - When do people(patients and doctors) need to be notified when something isn't right?
        - How will patients add device to account & app?

Instructor feedback about NexMed: “A lovely topic!! Once complete, it will save a lot of time and effort for doctors and patients who are to meet for a sustained healthcare system. Please consider implementing it fully if you can. No pressure w.r.t. grade on implementation. In the final report, please make sure to include comparison with similar applications -if any-, make sure that you differentiate your design from those, and explicitly specify how.
Fair delegation of tasks.
Please share this feedback with your group members.
You are good to go. Have fun with the project and hope everyone enjoys the collaboration.”

Git repository URL: https://github.com/sophiakobzar/3354-codeByters.git

Delegation of tasks of project deliverable 1:

- Alex Chan-Nui
  - Question: six
- Cory Harris
  - Question: eight
  - Commit file to GIT repository
- Jeremy Pacheco
  - Question: nine
  - Commit file to GIT repository
- Nahome Abraham
  - Question: seven
- Sophia Kobzar
  - Questions: 1 & 3-5
  - Made the Git repository and committed to Git repository

Software process model that is employed in the project and why:

- The Waterfall model is what the group choose:
  - In the waterfall model everyone gets up to speed quickly, schedules are kept, there are no financial surprises, testing is done easily, and the outcome is clear; so, you get what you planned. The waterfall model can also be challenging to define needs, lacks flexibility, and longer delivery time.. Often the most criticized aspect of waterfall development; the customer does not know what they want. It is important that software developers can guide and advise clients effectively to avoid problems later.

Software Requirements:

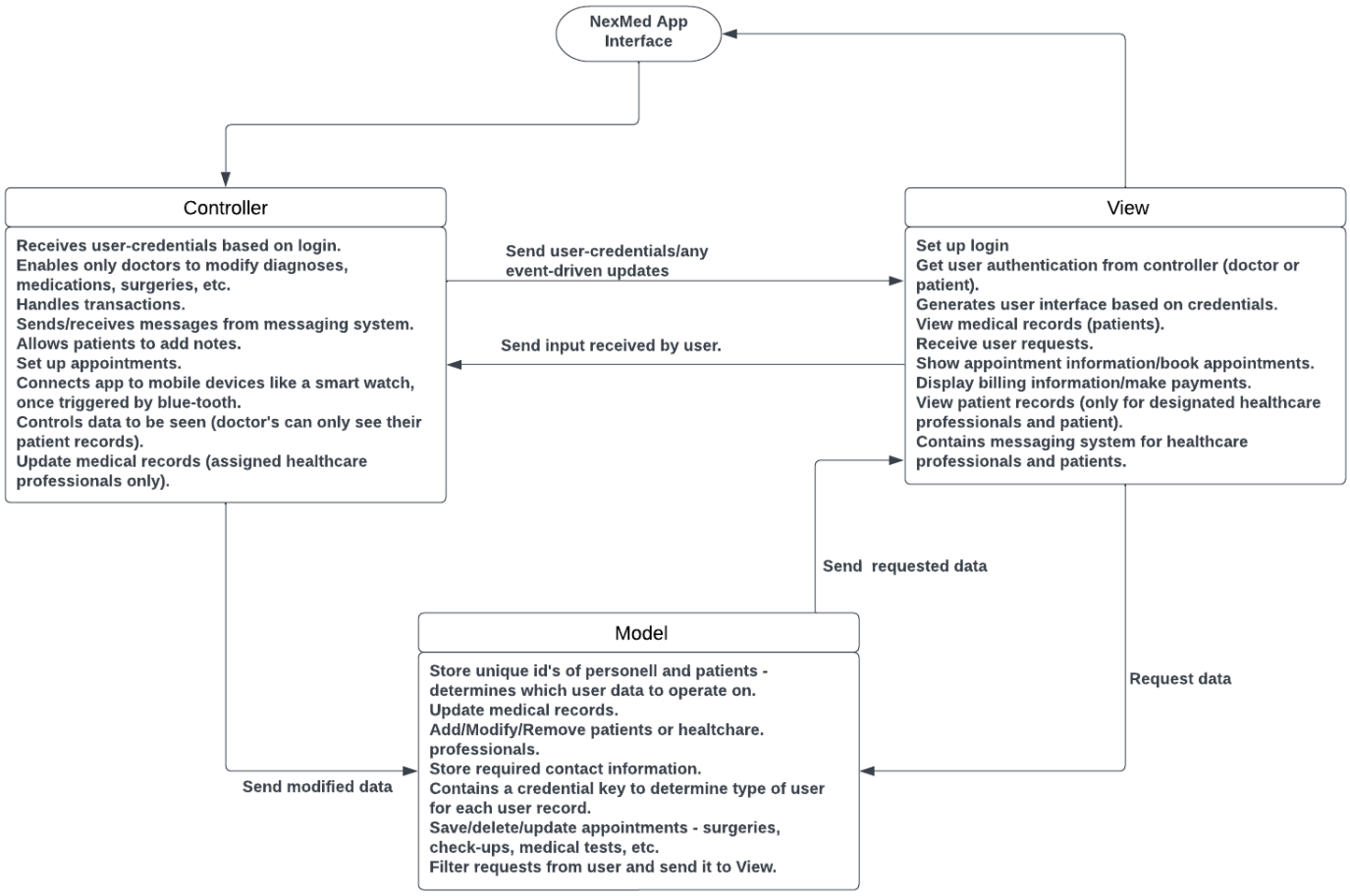
- Functional requirements
  - Doctors should be able to connect to patients’ profile.
  - Patients and their doctors should see the patient’s profile. Doctors should be able to see their patients only.
  - Users(doctors, patients, patient’s guardian if patient minor) should be able to sign into their account. If a doctor becomes a patient the use interface will be different.
  - Minors can see their profile and their guardian can to until they separate. Guardians and minor patients can separate when the minor patient turns 18 years old, or minor is emancipated.
  - Patient’s profile (medications, surgeries, allergies, etc.) should be displayed on their account in a user-friendly manner.

- The information about the patient's medication that is needed:
  - Information of the patient (Name, DOB, home address, contact information)
  - If the medication is currently taken.
  - Size of pills (if applicable could be liquid)
  - size of bottle (if applicable could be pills)
  - Dose size (number of pills or liquid dose)
  - When should the patient take the medicine?
  - Anything notes the patient needs to know (side effects, only take medicine after/before eating, etc.)
  - Which pharmacy was the medicine sent to?
- The information about the patient's surgery that is needed:
  - When was the surgery (time and date)
  - Where was the surgery (which hospital)
  - What is the surgery the patient had?
  - What did the surgeon do?
  - Contact information of the surgeon team

- Non-functional requirements
  - Product requirement
    - Efficiency requirements
      - Performance requirements
        - For read requests max 100 milliseconds latency
        - For write requests(create or update data) we want to keep under 300 milliseconds latency.
      - Space requirements
        - There should be enough space in the server for patient data. Also, copy need to be made of the data.
        - We will utilize AWS infrastructure to run frontend, backend and database services.
    - Dependability requirements
      - No matter the time or day the patient's data should be reachable.
    - Security requirements
      - Only appropriate staff members should be able to see the patient's data.
    - Usability requirements
      - Any laptop or Mac can access the website.
      - IOS and android can download the app.

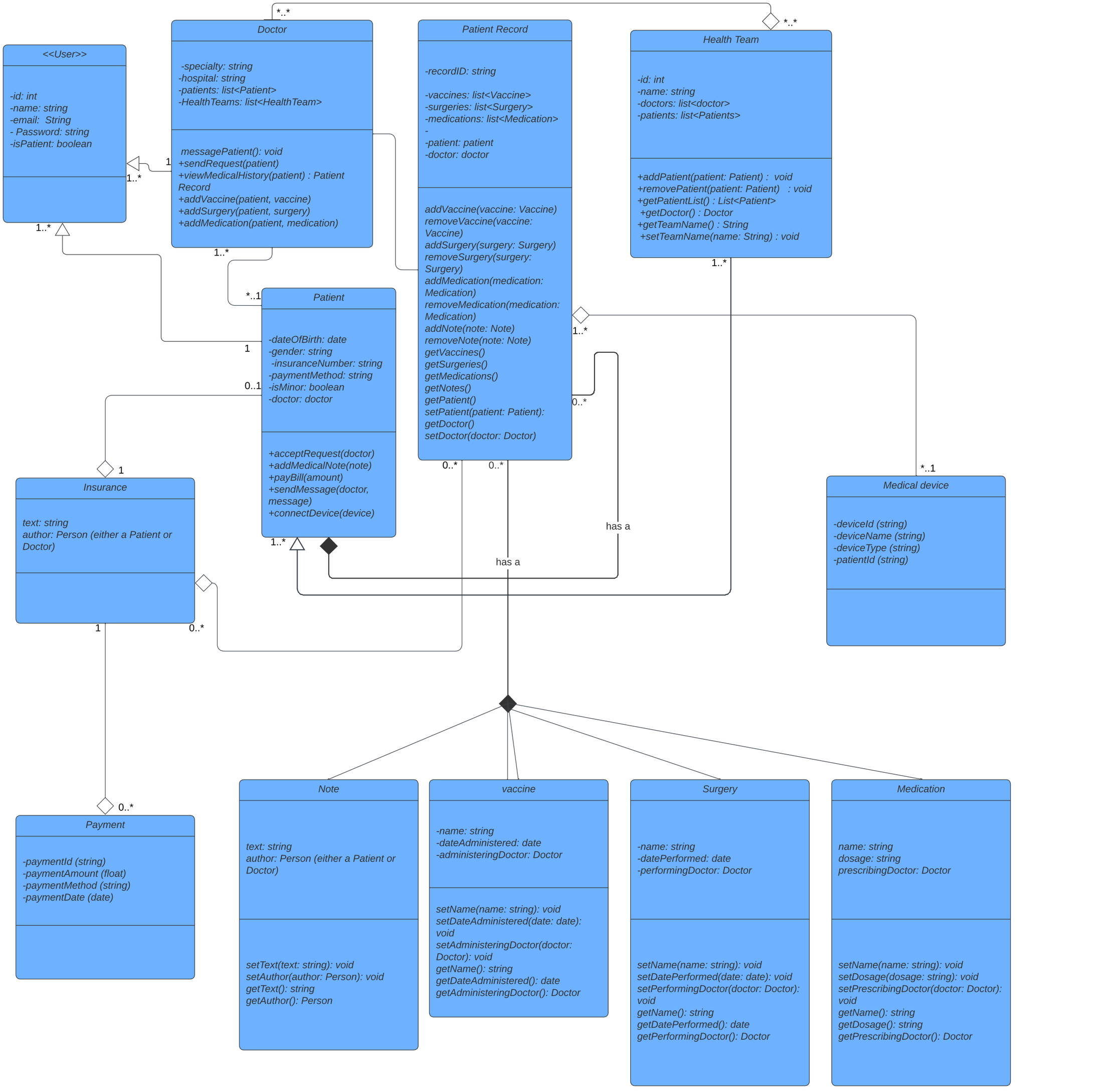
  - Organizational requirements
    - Environmental requirements
      - The servers should be kept in a clean cool room.
    - Operational requirements
      - Someone needs to be hired to maintain the server room.
    - Development requirements
      - Developers need to be hired to do regular updates.

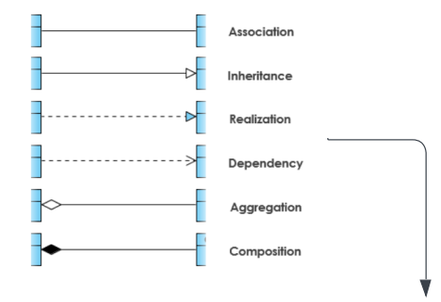
- External requirements
  - Regulatory requirements
    - Random people should not be able to access the server room.
  - Ethical requirements
    - Doctors should not be able to share the patient's data with unapproved sources.
  - Legislative requirements
    - Accounting requirements
      - Keep track of the current clients and getting new ones.
    - safety/security requirements
      - Only appropriate members can access the data room.
      - Apps also must follow HIPPA laws.



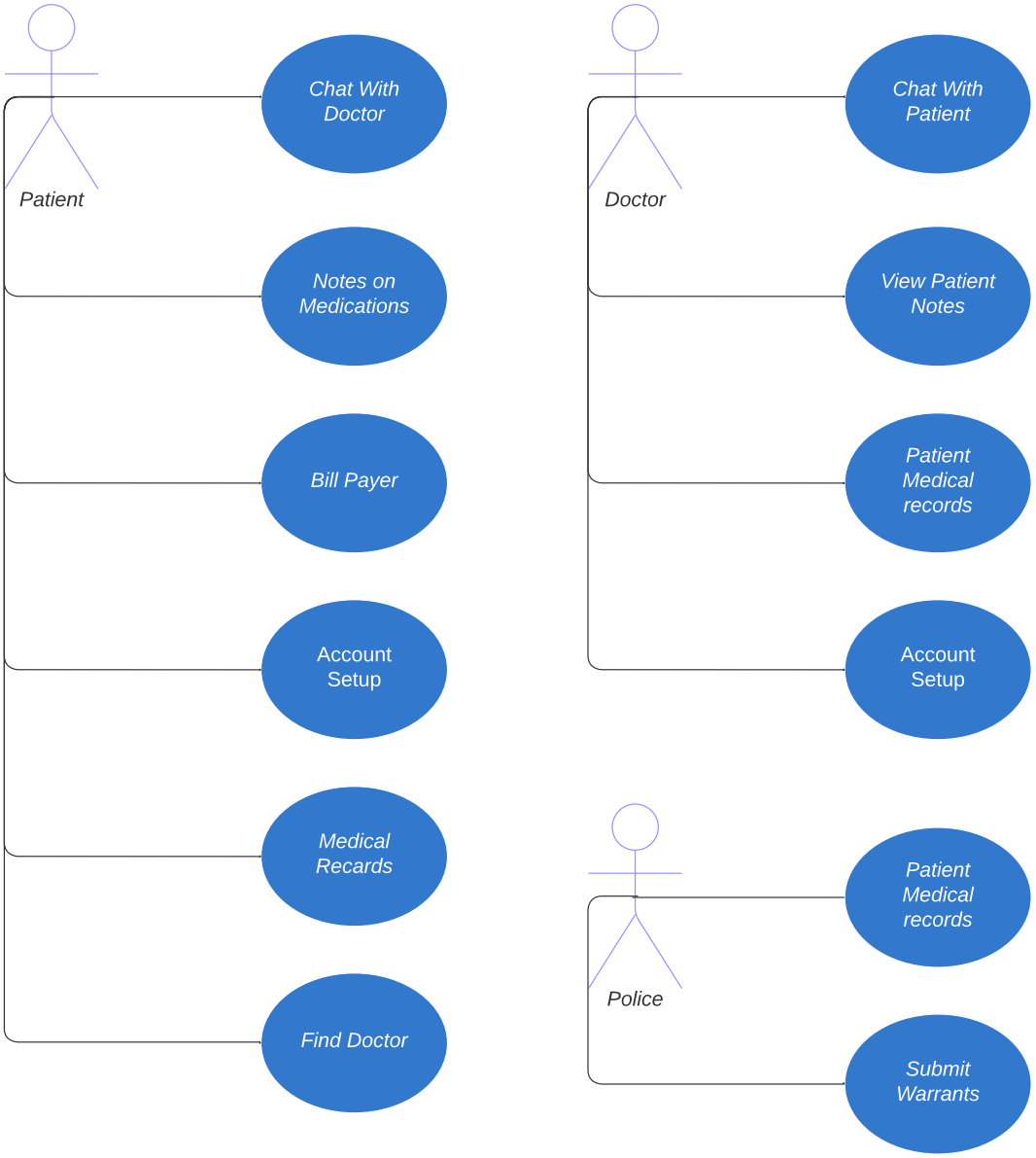
ClassDiagram_NexMed

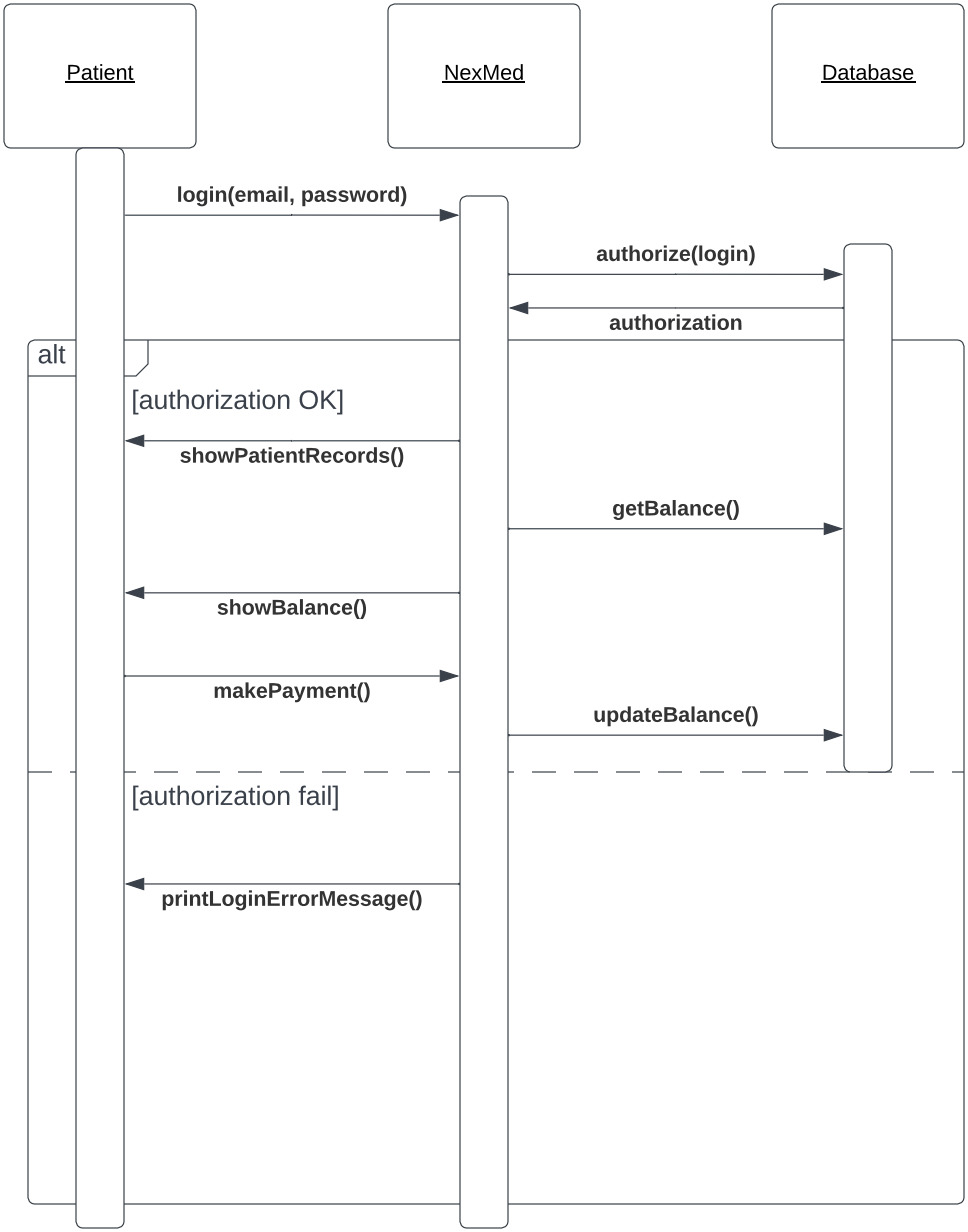
cory harris | March 24, 2023

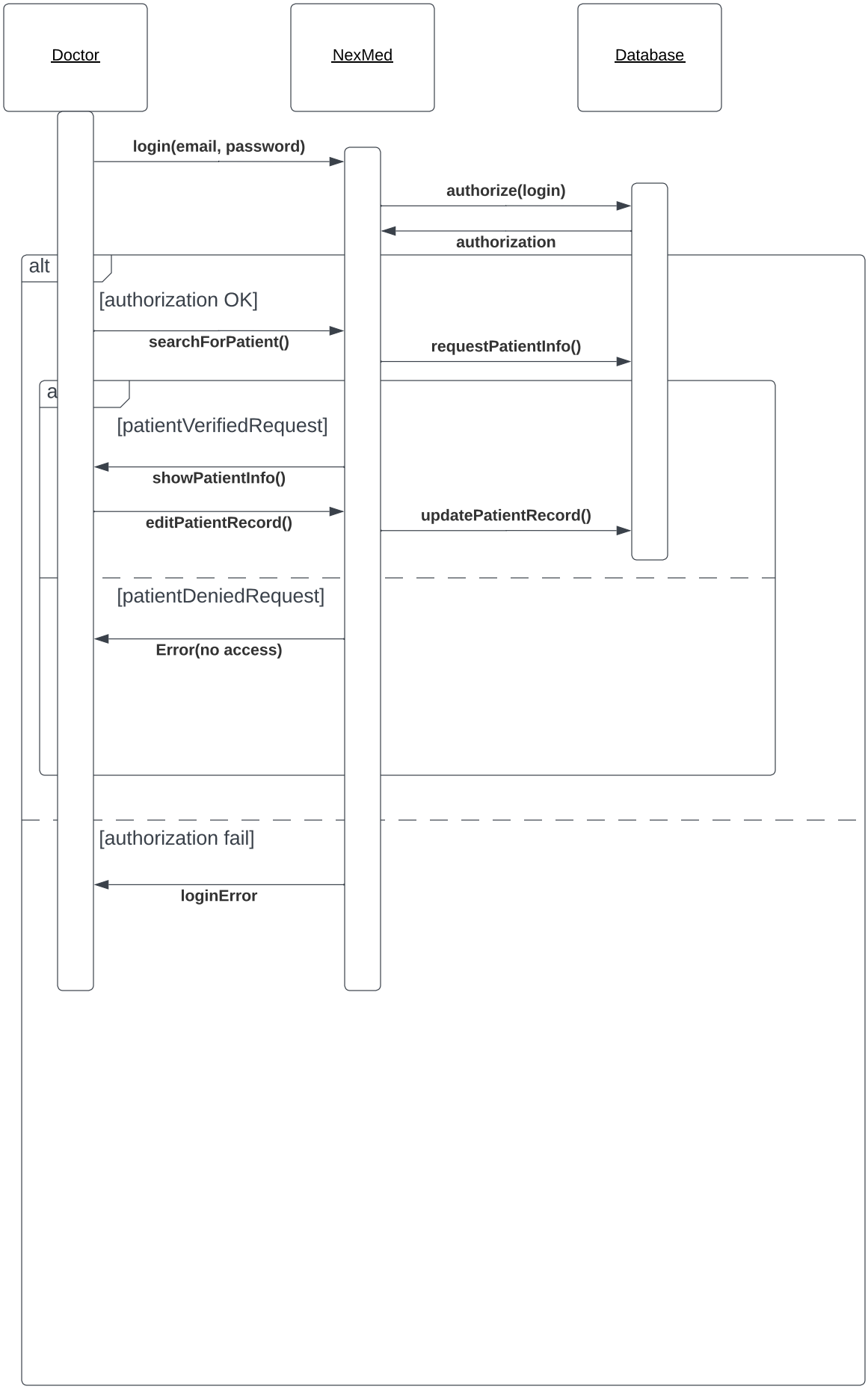




SOURCE:https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/


Patient
Chat With Doctor
Notes on Medications
Bill Payer
Account Setup
Medical Recards
Find Doctor
Doctor
Chat With Patient
View Patient Notes
Patient Medical records
Account Setup
Police
Patient Medical records
Submit Warrants


Patient
NexMed
Database
login(email, password)
authorize(login)
authorization
alt
[authorization OK]
showPatientRecords()
getBalance()
showBalance()
makePayment()
updateBalance()
[authorization fail]
printLoginErrorMessage()


Doctor
NexMed
Database
login(email, password)
authorize(login)
authorization
alt
[authorization OK]
searchForPatient()
requestPatientInfo()
a
[patientVerifiedRequest]
showPatientInfo()
editPatientRecord()
updatePatientRecord()
[patientDeniedRequest]
Error(no access)
[authorization fail]
loginError

Back to deliverable 2

Project Scheduling, Cost, Effort and Pricing Estimation, Project duration and staffing for NexMed:

3.1
Our project deals with a lot of full-stack development and is an extensive mobile application. Due to this, our estimated start date would be May 1st, and our expected end date would be June 30th. We like a work-life balance, so weekends will not be included in our schedule. We expect to be working anywhere from 6-10 hours a day on this project, with an average of 8 hours a day. 3.2
We used the FP method for cost estimation of the application itself.

1B. Function Point (FP) Method:

1. Determine function category count.

	Function category	Count	Complexity			Count*Complexity
			Simple	Average	Complex	
1	Number of user input	15	3	4	6	
2	Number of user output	20	4	5	7	
3	Number of user queries	30	3	4	6	
4	Number of data files and relational table	20	7	10	15	
5	Number of external interfaces	5	5	7	10	
					GFP	

2. Determine complexity.

	Function category	Count	Complexity			Count*Complexity
			Simple	Average	Complex	
1	Number of user input	15	3	4	6	45
2	Number of user output	20	4	5	7	100
3	Number of user queries	30	3	4	6	120
4	Number of data files and relational table	20	7	10	15	200
5	Number of external interfaces	5	5	7	10	35
					GFP	

3. Compute gross function point (GFP)

	Function category	Count	Complexity			Count*Complexity
			Simple	Average	Complex	
1	Number of user input	15	3	4	6	45
2	Number of user output	20	4	5	7	100
3	Number of user queries	30	3	4	6	120
4	Number of data files and relational table	20	7	10	15	200
5	Number of external interfaces	5	5	7	10	35
					GFP	500

4. Determine processing complexity (PC)
To determine the PC, you need to answer the following questions:
1) Does the system require reliable backup and recovery? 4
2) Are data communications required? 4
3) Are there distributed processing funcϴons? 5
4) Is performance critical? 5
5) Will the system run in an existing, heavily utilized operational environment? 4

6) Does the system require online data entry? 5
7) Does online data entry require the input transaction to be built over multiple screens or operations? 4
8) Are the master files updated online? 4
9) Are the inputs, outputs, files, or inquiries complex? 3
10) Is the internal processing complex? 3
11) Is the code designed to be reusable? 3
12) Are conversion and installation included in the design? 3
13) Is the system designed for multiple installations in different organizations? 5
14) Is the application designed to facilitate change and ease of use by the user? 3

5. Compute processing complexity adjustment (PCA)
    - .65+.01*(4*5 +5*4 +5*3)=1.2 PCA
6. Compute function point (FP) using the formula: FP = GFP × PCA
    - 500*1.2=600

Effort = $\frac{FP}{productivity}$ 600/40 15

35 week estimation

Team size: 5

Project duration: 15/5 = 3 weeks

3.3

If this app becomes a success, we expect this app to generate a decent amount of user-interaction, with about a million per day. We expect to be using about 2 servers, which costs around $300 a month. The majority of User data will be stored with our clients pre existing system architecture (Hospitals IT department). This allows us to avoid storage issues related to HIPAA requirements, as well as offload Cybersecurity costs to the client.

3.4

For Software costs, there will be significant overhead for cloud resources. Some of this can be offset with the clients existing resources, however in the case that they do not have on Prem or cloud sources to store data, our app would require at least 50TB of Archival storage, 200TB of general access storage and another 100TB fast access storage per month. We primarily would recommend this to be hosted via a 3rd party cloud provider with good security such as Google Cloud, AWS, or Microsoft and alternatively IBM cloud compute services. The example pricing for these would be as follows. Based on AWS exclusive system the requirements instance would be use based for example if all 200TB of standard data is being used the price to maintain would be 4,600$ per month. Archival would be around 200$ a month and for the Fast access storage the price would be $2300 per month. This example pricing is based on the S3 pricing page for AWS https://aws.amazon.com/s3/pricing/

3.5

As for the software developers, we'll have about 3 front-end (12.5k each /month), 3 back-end (12.5k each/month), and 2 full-stacks (16.5k each/month). The total costs for software developers' part will be $108k a month. We'll pay the implementation success manager $6k/month. The implementation manager will be responsible for the successful implementation

of the application on the Hospital(client) site. They will direct the training plan which will include tutorial and training videos and FAQs hosted on our platform and built by the developers.

A test plan for NexMed software:

- Following tests, we created:
    - Test display of sign in page.
        - In real world would be a pop up and test the pop up
        - We used a function that would print out the “sign in page” which is just string and I tested to make sure what is printing on the console is what is needed for the page.
    - Test credentials of doctors to make sure if correct credentials given the sign in is allowed.
        - We have a 2-dimensional array that holds correct credentials of the doctor and a for-loop that checks if the given username and password is correct.
        - In the real world the client servers would do check if the username and password is correct. The opposite should be tested to if incorrect password is given give an error and do not allow to sign in.
    - Test credentials of patients to make sure if correct credentials given the sign in is allowed.
        - We have a 2-dimensional array that holds correct credentials of the patient and a for-loop that checks if the given username and password is correct.
        - In the real world the client servers would do check if the username and password is correct. The opposite should be tested to if incorrect password is given give error and do not allow to sign in.
    - Test the data of patient’s data is properly display.
        - To test to make sure the user is seeing what they should be seeing.
        - In the real world our doctor’s data should hold a lot more than what our demo holds but for testing purposes it’s enough to show the idea.
    - Test the data of doctor’s data is properly display.
        - To test to make sure the user is seeing what they should be seeing.
        - In the real world our patient’s data should hold a lot more than what our demo holds but for testing purposes it’s enough to show the idea.

Comparison of NexMed with similar designs:

There are multiple competitors in this space. One of the largest apps already being used by many doctors all over the United States is Healow. Healow is similar in many ways to NexMed but also different in key areas. Major differences between NexMed and Healow are records, doctor communication and scheduling appointments. In the Healow app you can schedule appointments with doctors just through the app but in NexMed you will need to contact the doctor directly. This may seem like a downside but one of the major goals of NexMed is to encourage open communication between the patient and their doctor. One of the ways the NexMed goes in a different direction than Healow is a direct communication line between the patient and doctor. Like a text message thread, the doctor and

patient can communicate any time troubles come up and even just for general advice. Unlike in Healow where patients need to schedule video appointments or an in-person appointment to get help which can take days and sometimes even weeks for the video appointments and it can be even worse for a regular in-person appointment.
Records are good for both the doctors and the patient to have access to. The patient might need to go to a specialist or need to send those records to a work or school. While the doctor might need to know the medications that the patient is on as well as the procedures, they have undergone so that they know what the best course of action is for a patient to take. In NexMed the doctor and patient will both have access to the patient's medical records, vaccination records, and surgical records. Unlike in Healow, they only show basic Health records such as work that was done like blood tests with the patient's family physician. With greater access to the patient's records in NexMed both the doctor and the patient will benefit. This will allow the patient to ensure the records are up to date and give the doctor more information to make better informed decisions about their patients.

Conclusion:

For this project, our group is developing a healthcare application that will allow anyone to access their medical records, request refills for prescriptions, set up doctors appointments, etc. We plan to make the application address the needs of doctors and patients, including a doctor using the app as a patient as well. We based our project off of the likes of "Healow", another healthcare application. We named our application NexMed and believe it will be the next best thing in healthcare as it is a valuable and reliable app for managing medical operations.

In our project, we implemented the waterfall model as it keeps the developers up to speed, keeps schedules set, testing is done easily, and the goal is set and clear. However, the waterfall model lacks flexibility and can have a longer delivery time. Some of the functional requirements of NexMed will be that a doctor must be able to access their patients' profiles and only their patients, also doctors will view a user interface when in a user role. The interface shall be user-friendly. Parents/guardians will be able to view their children's profiles. Some nonfunctional requirements include that there be enough space in the server for all patient data, the app must be dependable and reliable any time any day, and also the app must be available on IOS and android.

Overall, we are satisfied with the work that we have done and believe that we have created a solid plan for NexMed, our healthcare app. We recognize that there may be further adjustments needed as we move forward with development, but we are confident that we have laid a strong foundation for success.

References:

eClinicalWorks, “Televisits,” *eClinicalWorks*, 09-Jan-2023. [Online]. Available: https://www.eclinicalworks.com/products-services/patient-engagement/televisits/. [Accessed: 21-Apr-2023].

“Healow software,” *SelectHub raquo*. [Online]. Available: https://www.selecthub.com/p/ehr-software/healow/. [Accessed: 21-Apr-2023].

K. Houseman, “New EMR mobile app: Healow,” *Healthcare Revenue Cycle Management Solutions*, 15-Feb-2013. [Online]. Available: https://www.revelemd.com/blog/bid/63022/New-EMR-Mobile-App-healow. [Accessed: 21-Apr-2023].

“S3,” *Amazon*, 2002. [Online]. Available: https://aws.amazon.com/s3/pricing/. [Accessed: 21-Apr-2023].

“With healow app your health is in your hands,” *healow*. [Online]. Available: https://healow.com/app.html. [Accessed: 21-Apr-2023].