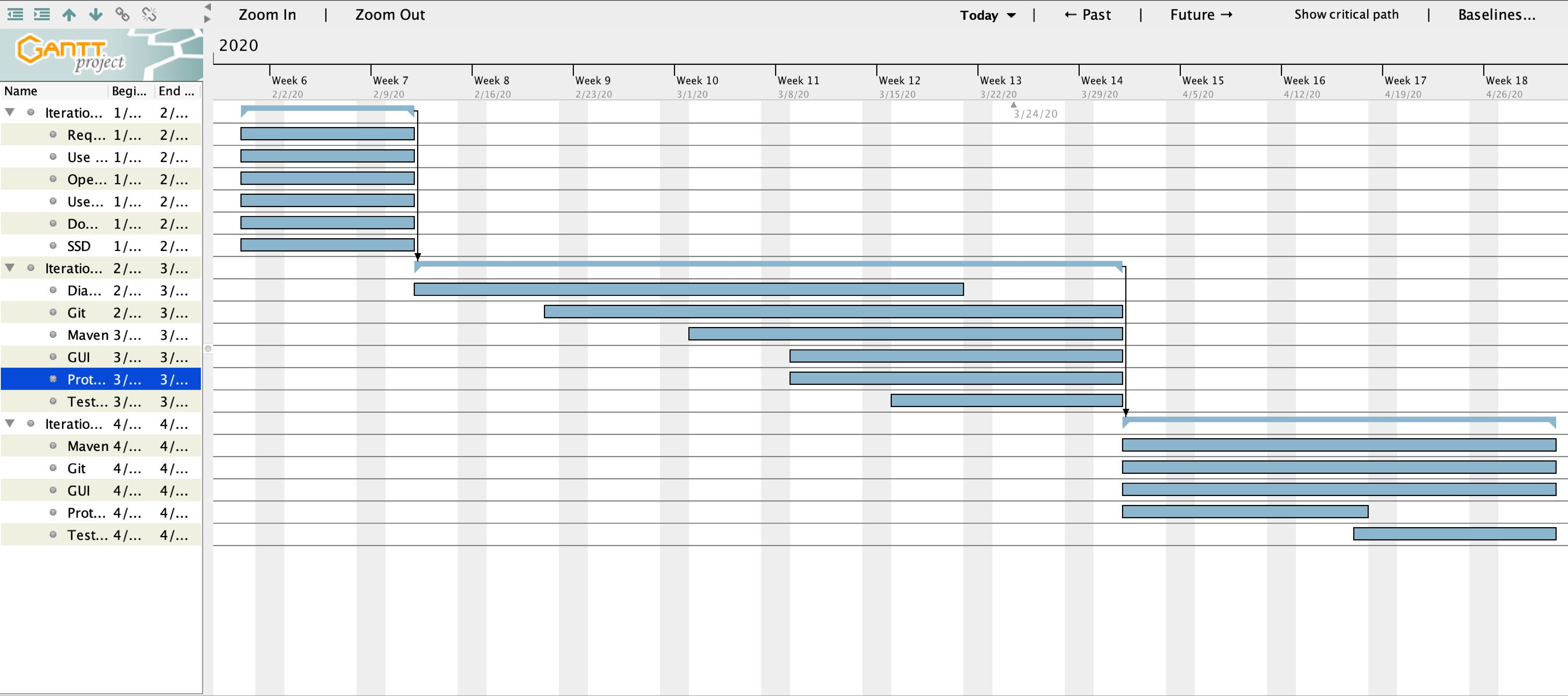




Gantt

Resources Chart



# GRASP

1. `getAppointmentDate`:
  - Location: Appointment
  - Principle/Pattern: Information Expert
    - Responsible for holding information for other classes; holds the time of the appointment for the notes and the billing information for the patient.
2. `getDescription / getName`:
  - Location: Diagnosis
  - Principle/Pattern: Information Expert
    - Responsible for holding information about diagnoses, these methods relay the information the class knows about the ailment of the patient, its name and description.
3. `createPrescription / orderTest`:
  - Location: Doctor
  - Principle/Pattern: Creator
    - Responsible for using the input of the doctor in order to create test and prescription objects
4. `updateHealthSheet`
  - Location: Doctor
  - Principle/Pattern: Controller
    - Responsible for using the information passed to the Doctor class to edit the patient's health sheet, which is outside of the doctor class.
5. `sendBillToRecipients`:
  - Location: Health Care Provider
  - Principle/Pattern: Controller
    - This operation is responsible for gathering information from the appointment and providing it to the necessary parties.
6. `editGraphProperties`:
  - Location: Health Care Provider
  - Principle/Pattern: Information Expert
    - This operation is responsible for changing the graph to represent the way it is to be displayed, which it knows.

7. getBillInfo:

- Location: Patient
- Principle/Pattern: Information Expert
  - Responsible for knowing the information regarding the bill the patient must pay, and potentially making that information available to the patient.

8. getPrescription:

- Location: Prescription
- Principle/Pattern: Information Expert
  - Responsible for knowing the information regarding the medication, number of refills, patient ID, and doctor ID along with the dosage. This information can be accessed by the doctor and patient.

9. storeDiagnosis / storePrescriptions:

- Location: Patient Diagnosis
- Principle/Pattern: Controller
  - Responsible for assigning an existing Diagnosis or Prescription to a Patient based on the input of the Doctor.

10. createAppointment / sendTestOrder:

- Location: Staff
- Principle/Pattern: Creator
  - Responsible for creating Appointments and Tests

11. getTestName / getInsuranceCode:

- Location: Test
- Principle/Pattern: Information Expert
  - Responsible for storing data necessary to be able to perform a test on a Patient in a lab.

12. sendTestOrder / confirmTest:

- Location: Test Order
- Principle/Pattern: Controller
  - Responsible for assigning an existing Test to a Patient based on input from a Staff member.

13. getTestResult:

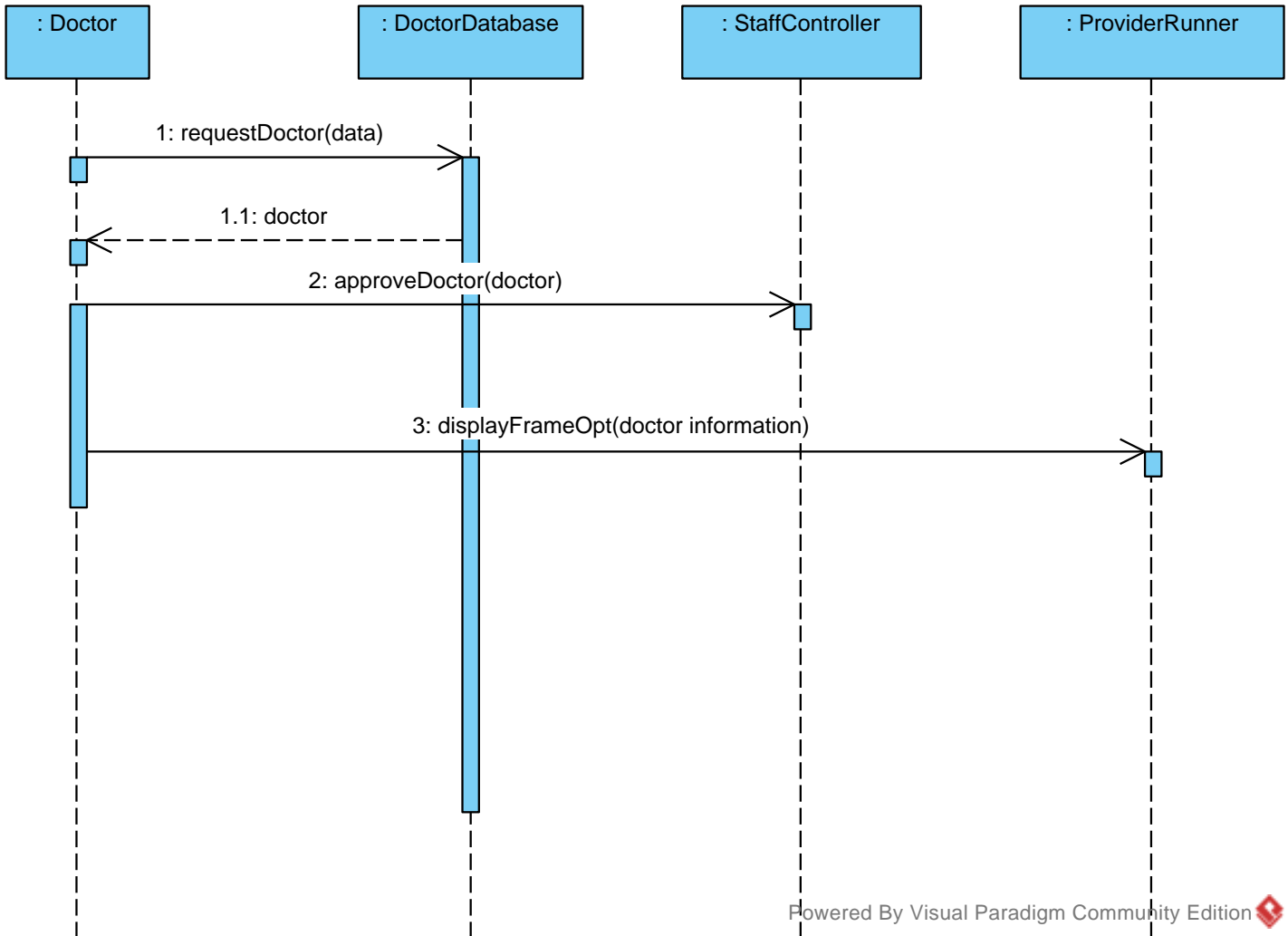
- Location: Test Result
- Principle/Pattern: Controller
  - Responsible for taking input from the Test which was performed and relaying the information to the Patient.

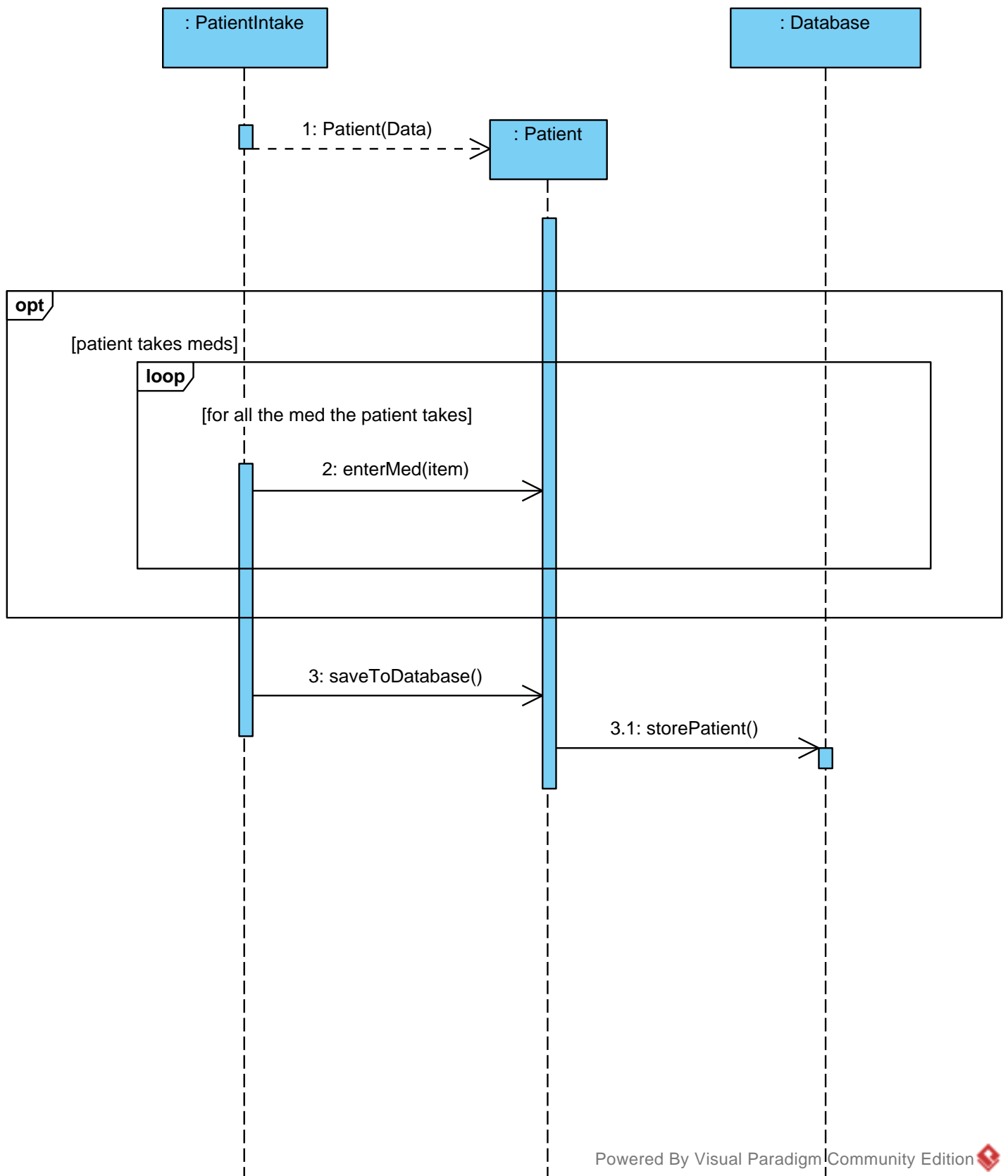
14. viewSchedule:

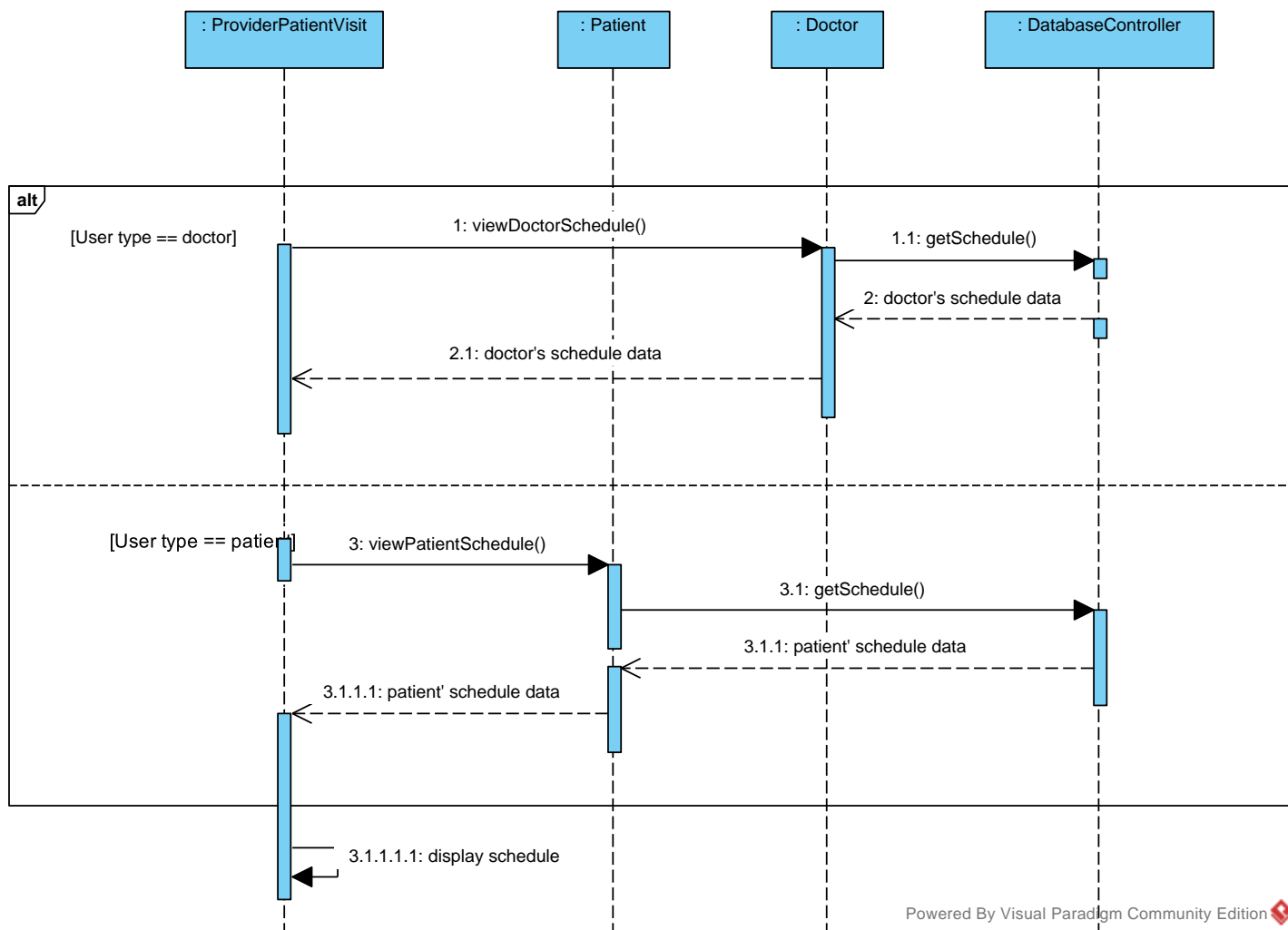
- Location: User
- Principle/Pattern: Low Coupling, High Cohesion, Polymorphism, Information Expert
  - Useful for polymorphism, allows different types of users to view their own personalized schedule, being either Doctor, Patient, or Staff

15. SqlConnection():

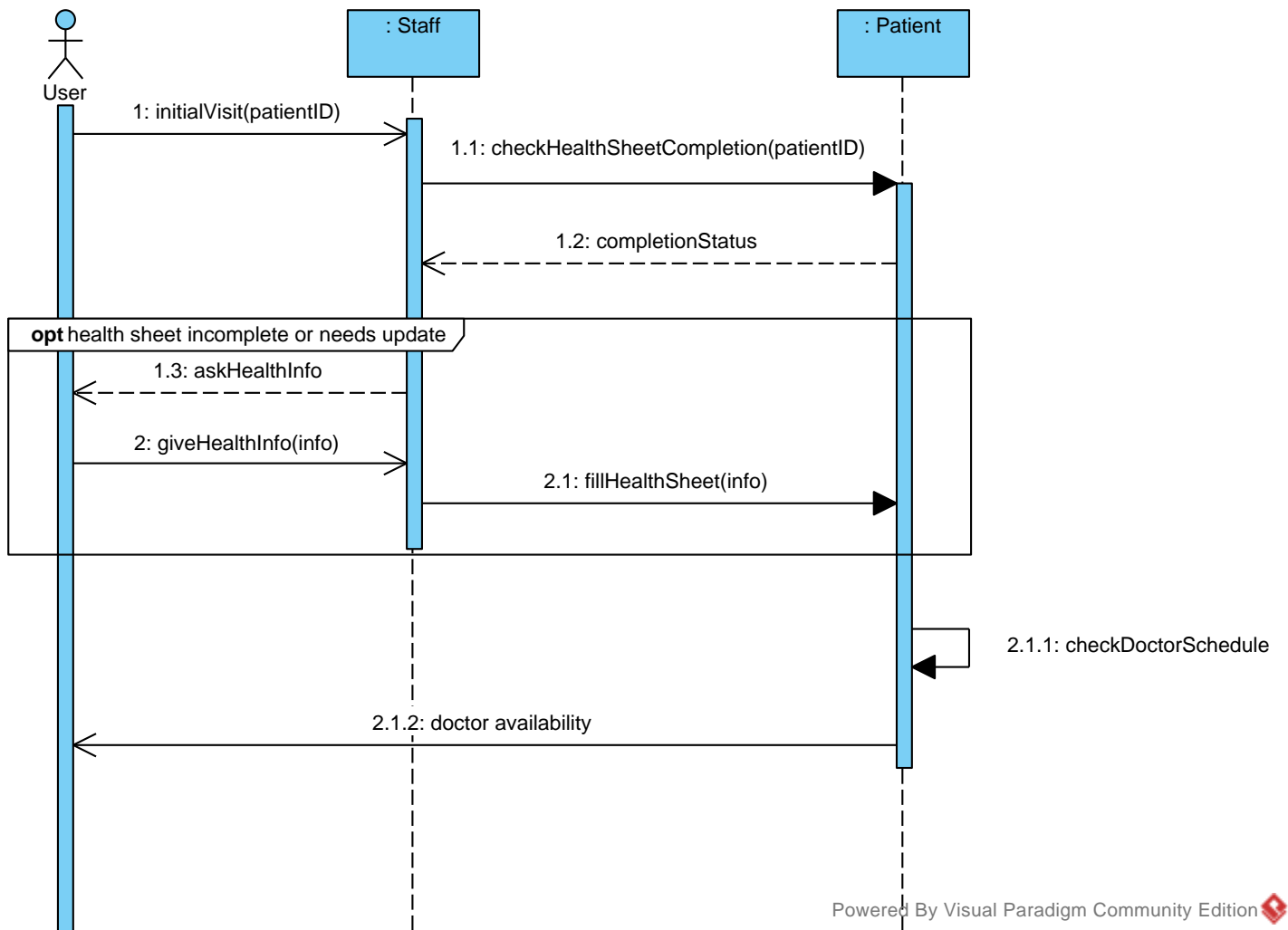
- Location: SqlConnection
- Principle/Pattern: Low Coupling, Controller
  - Receives information from a database and initializes data in other classes

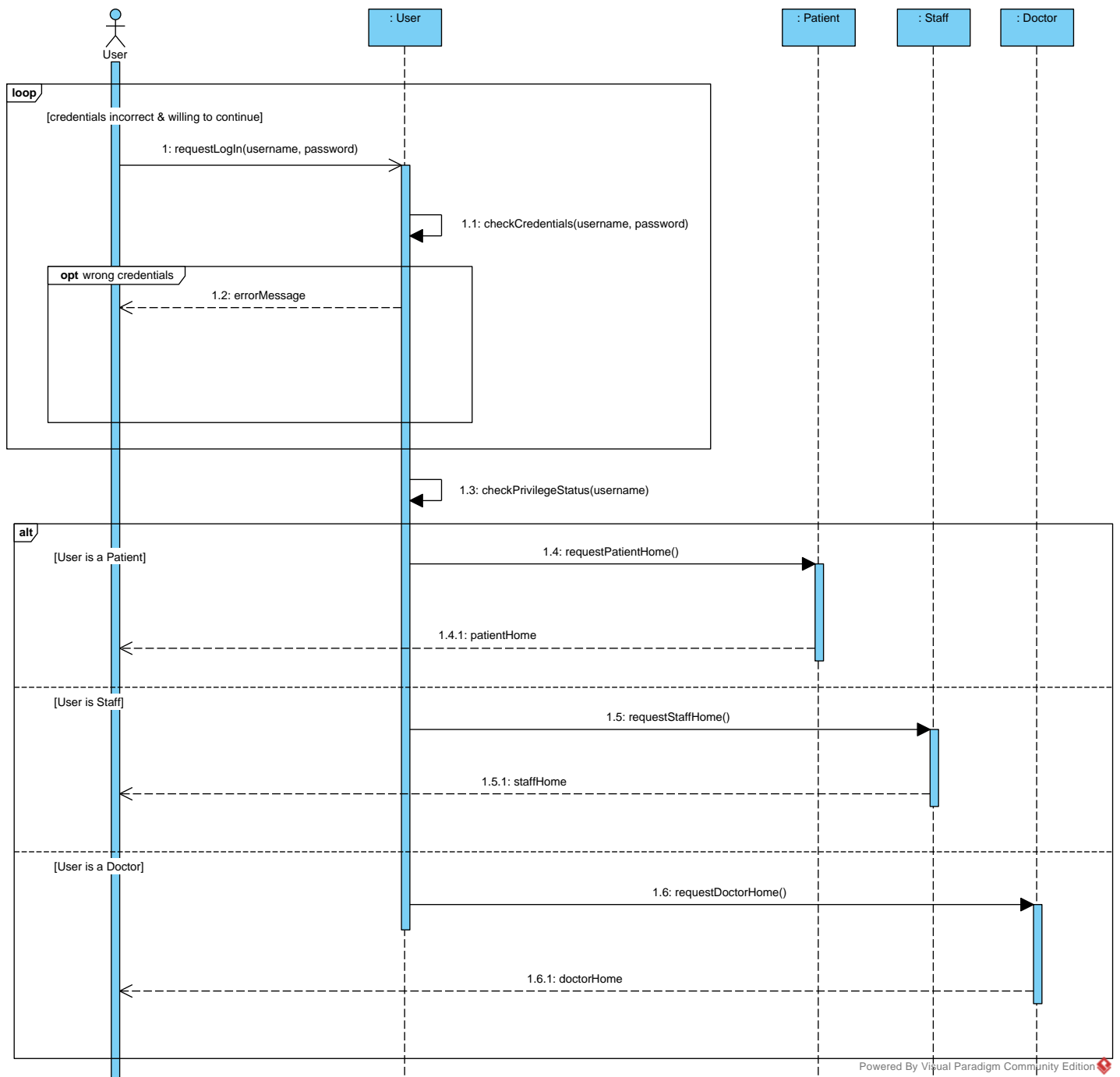


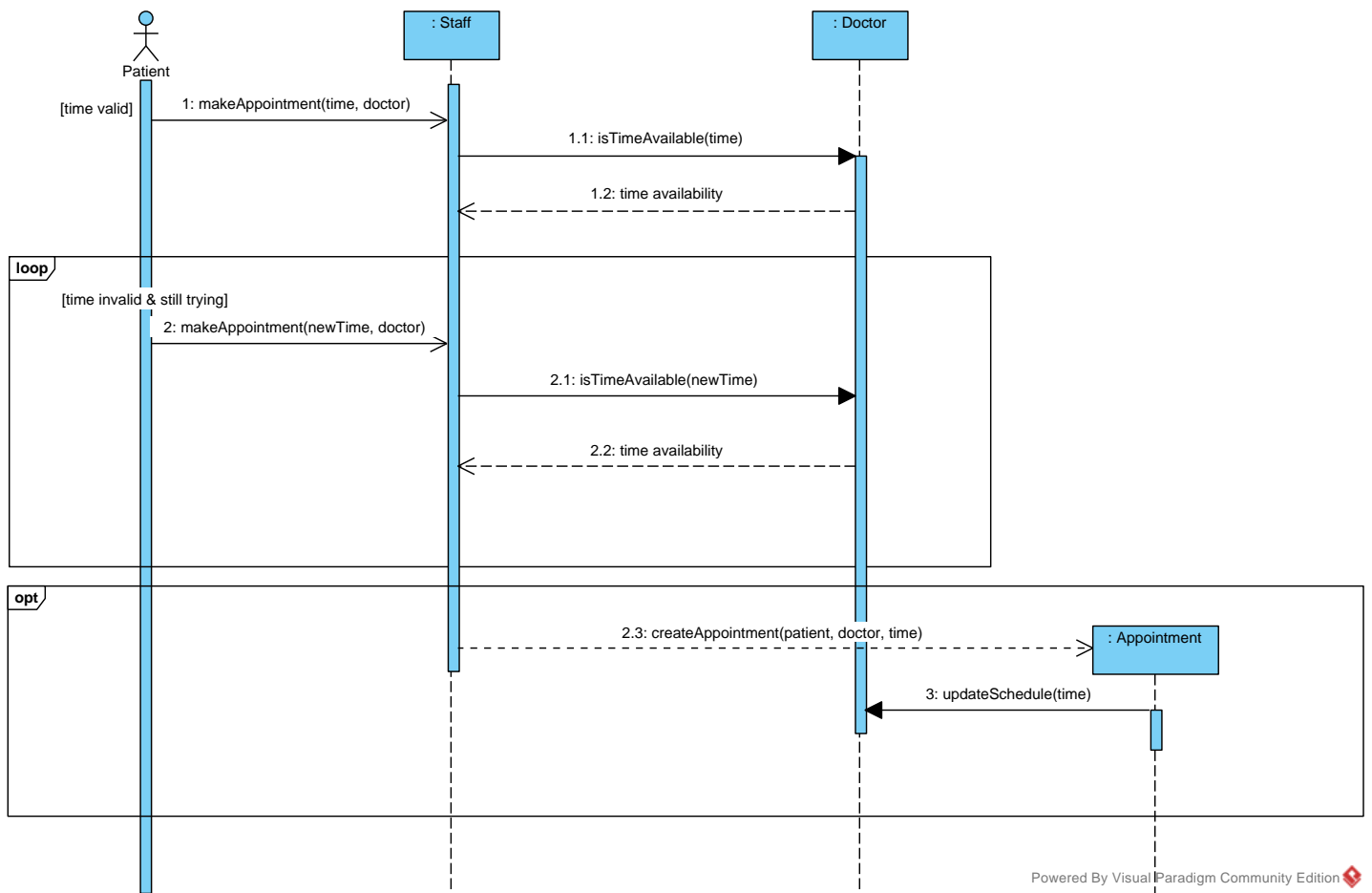


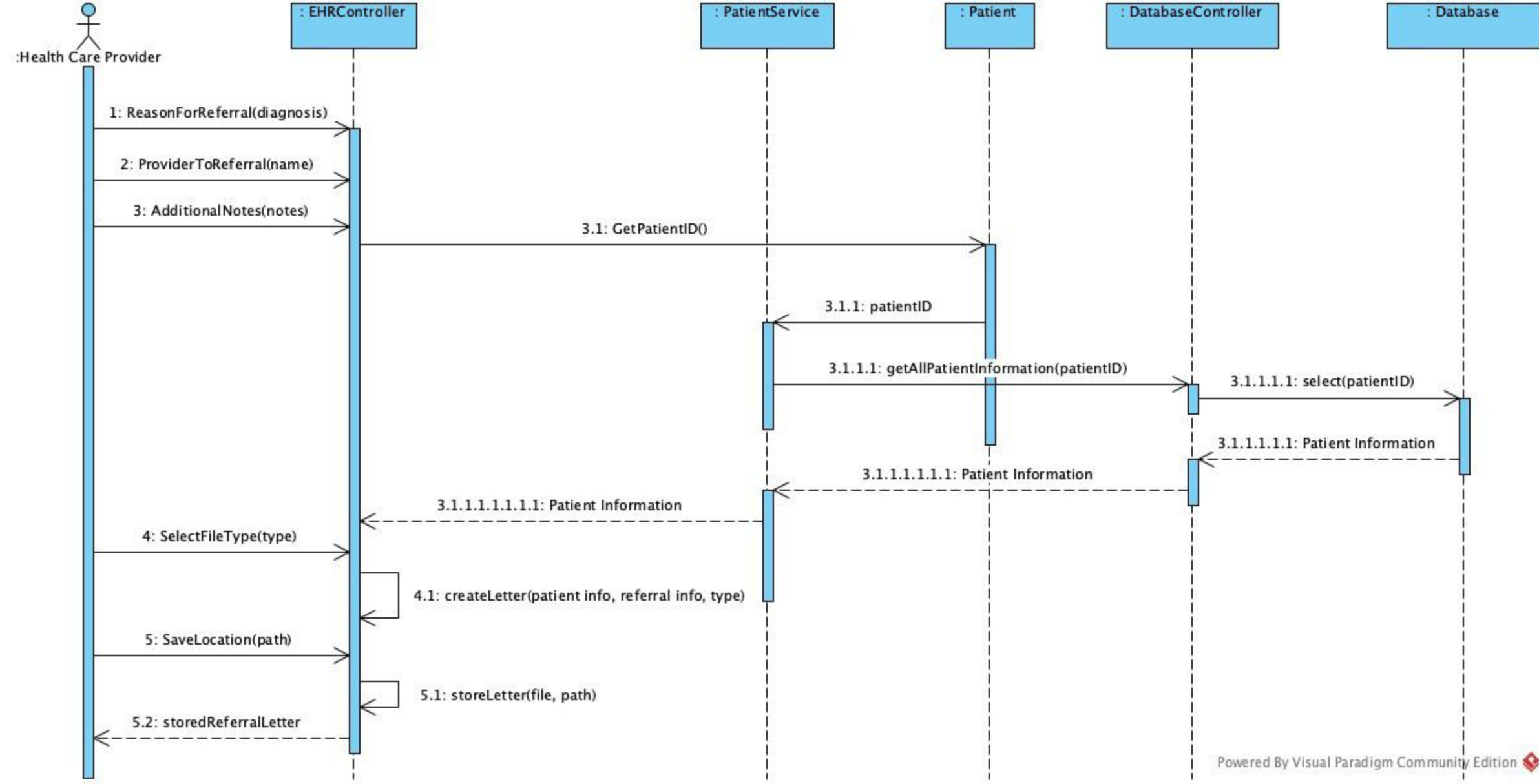


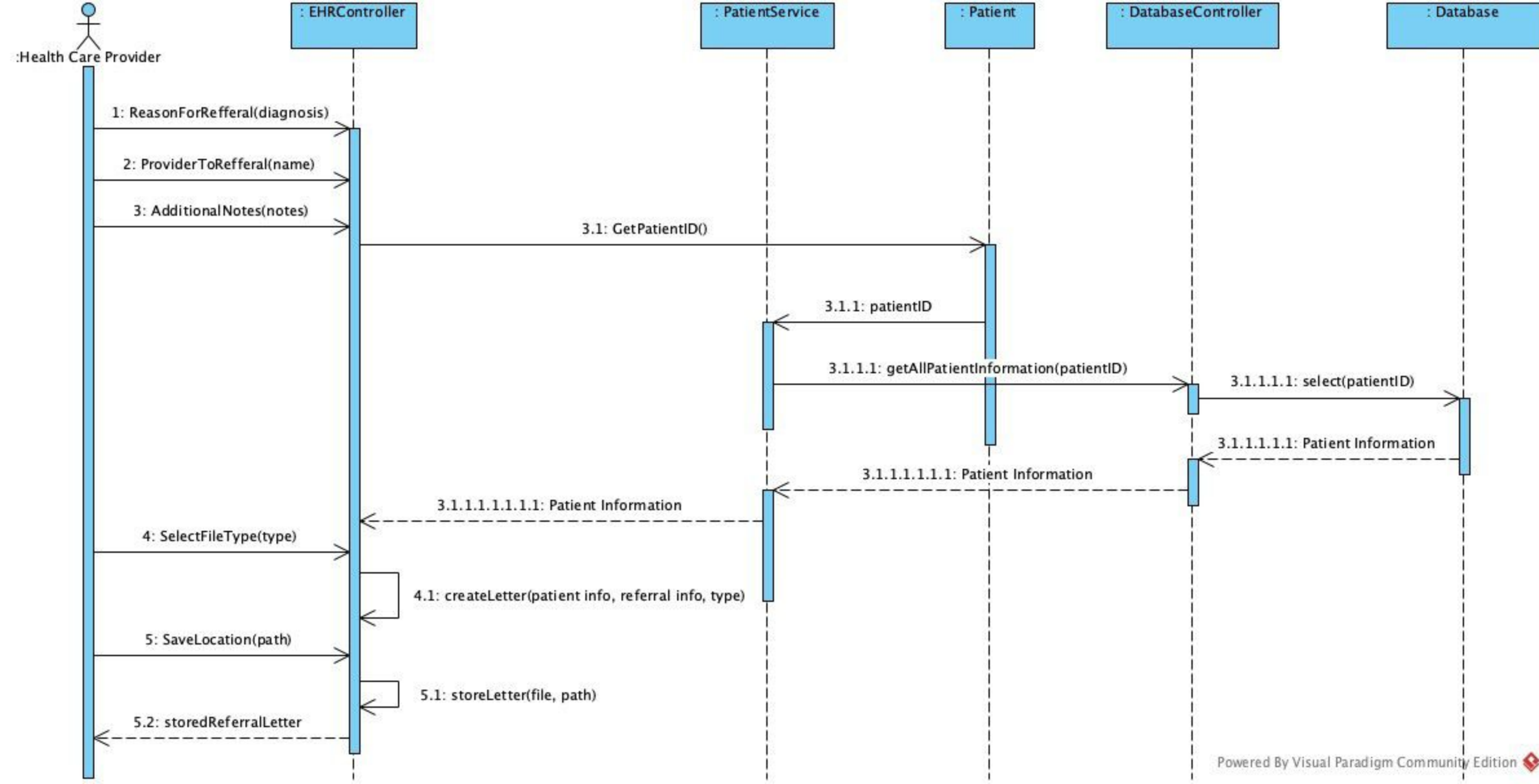


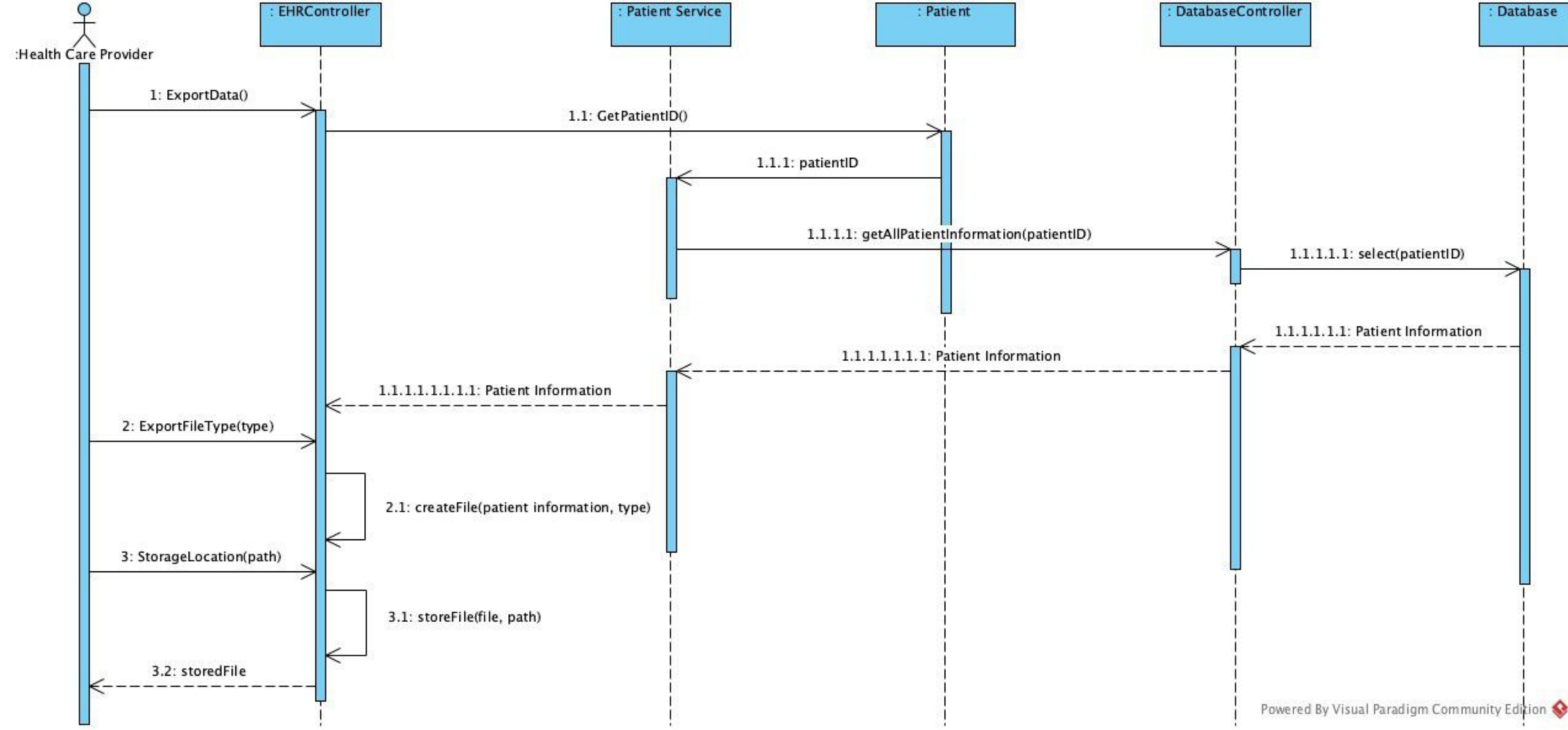


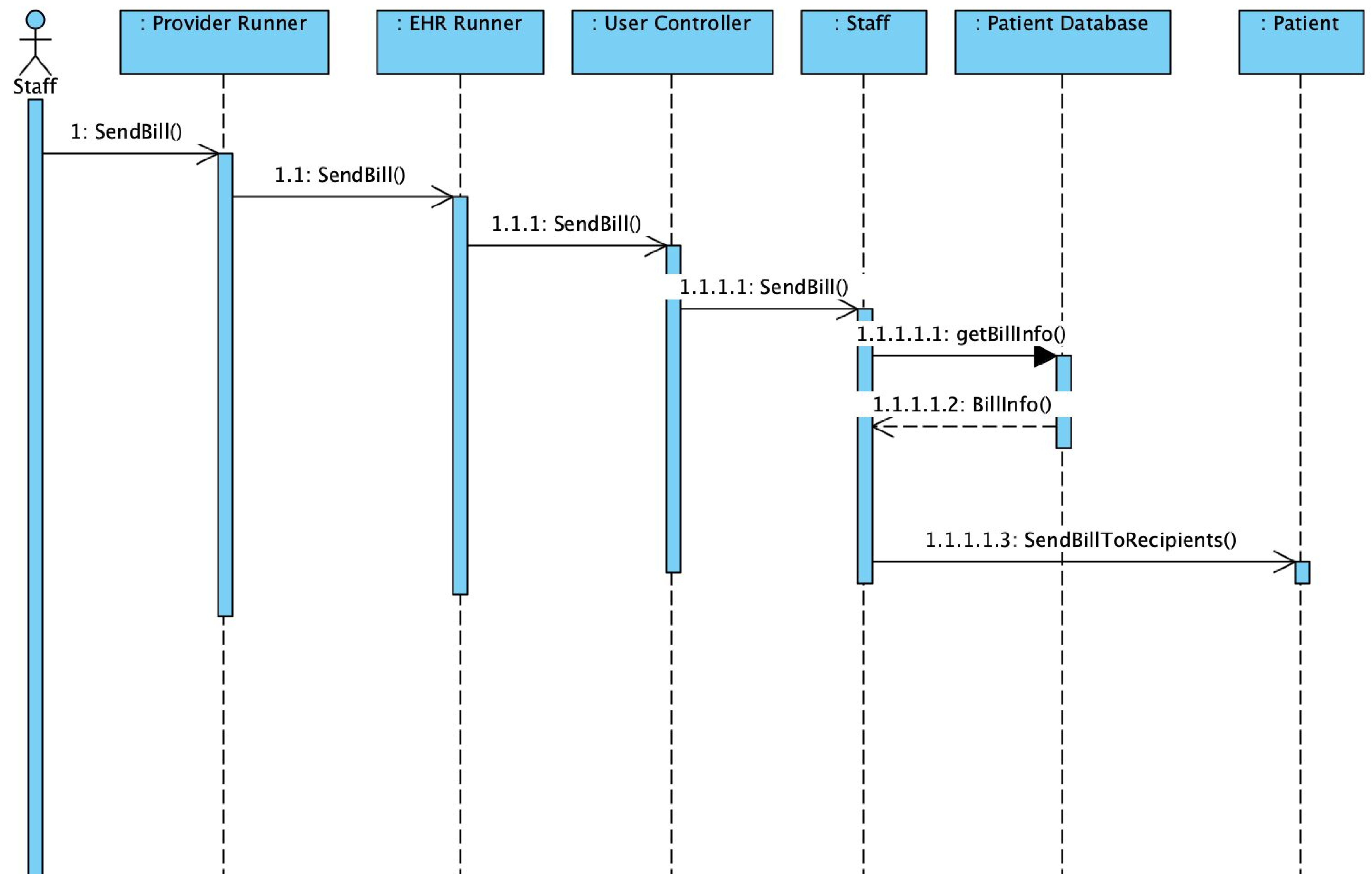




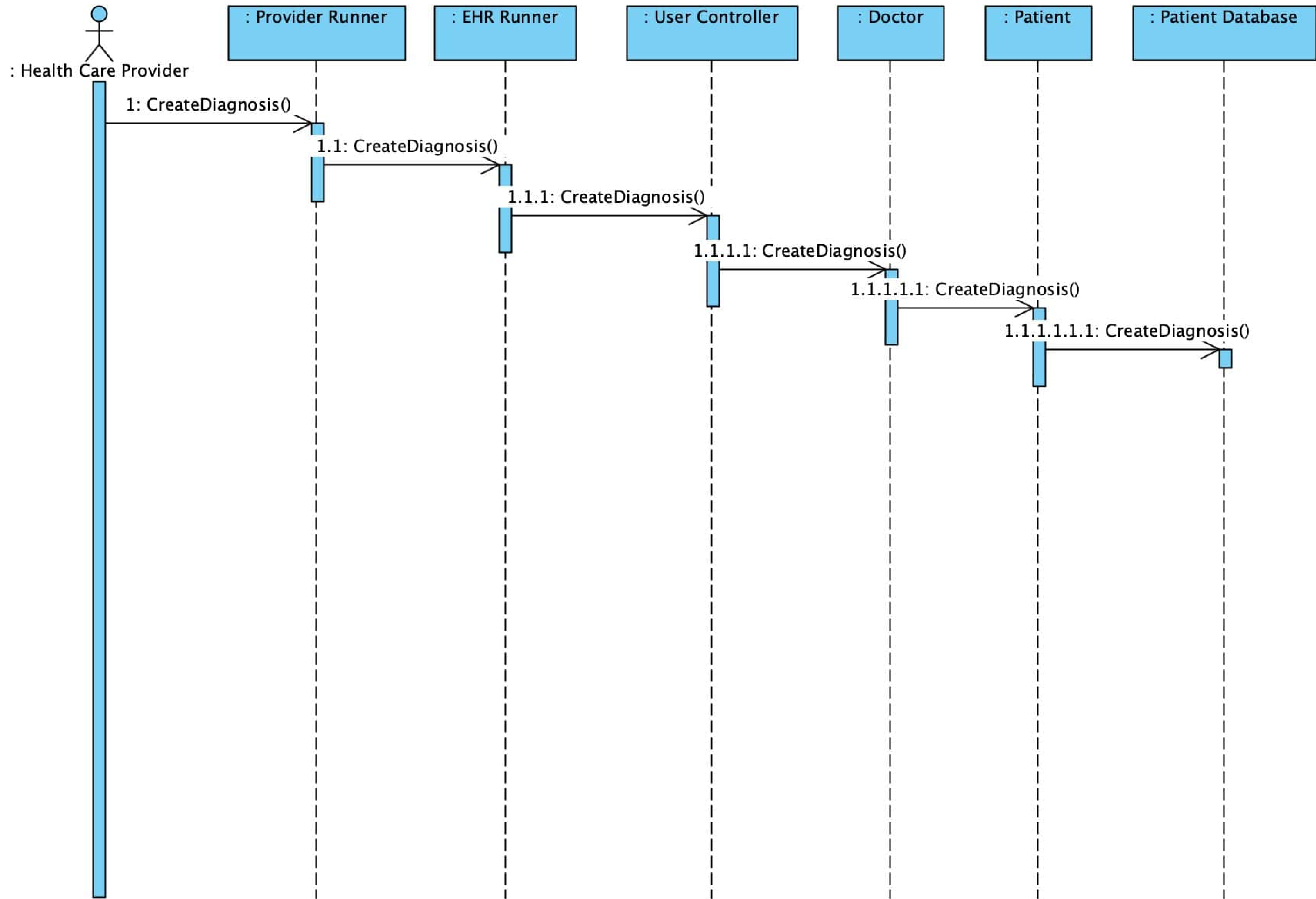




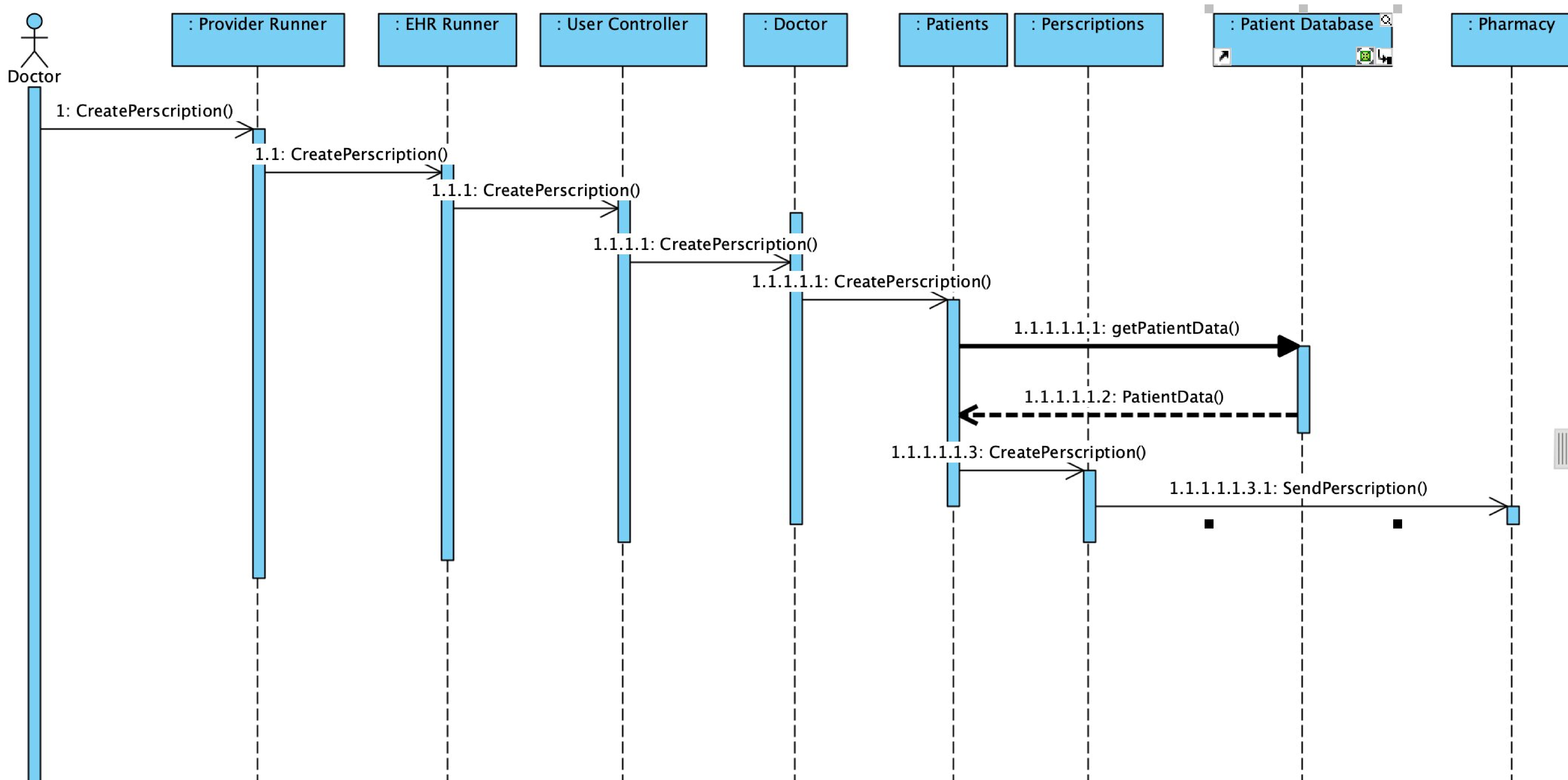


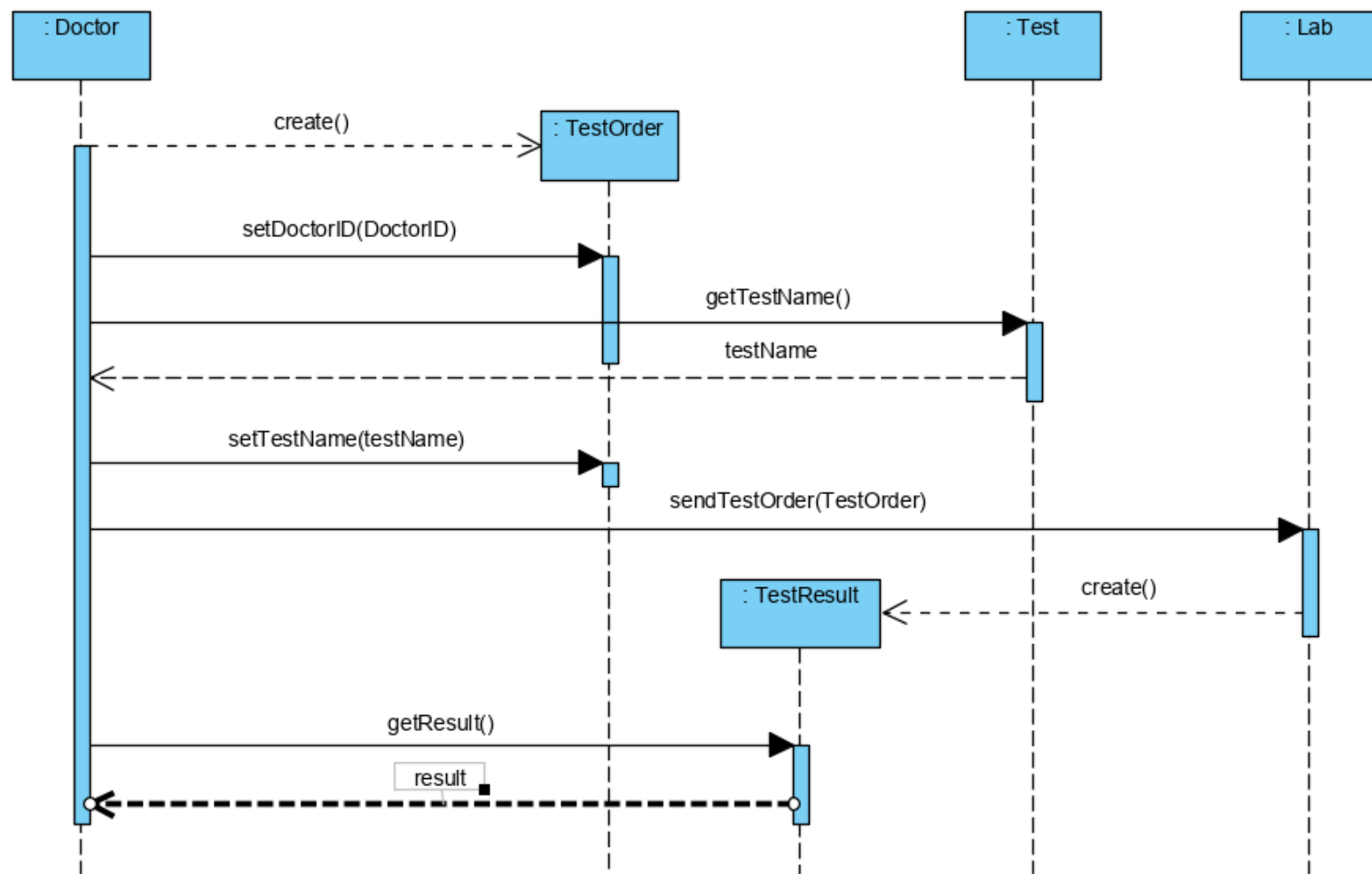


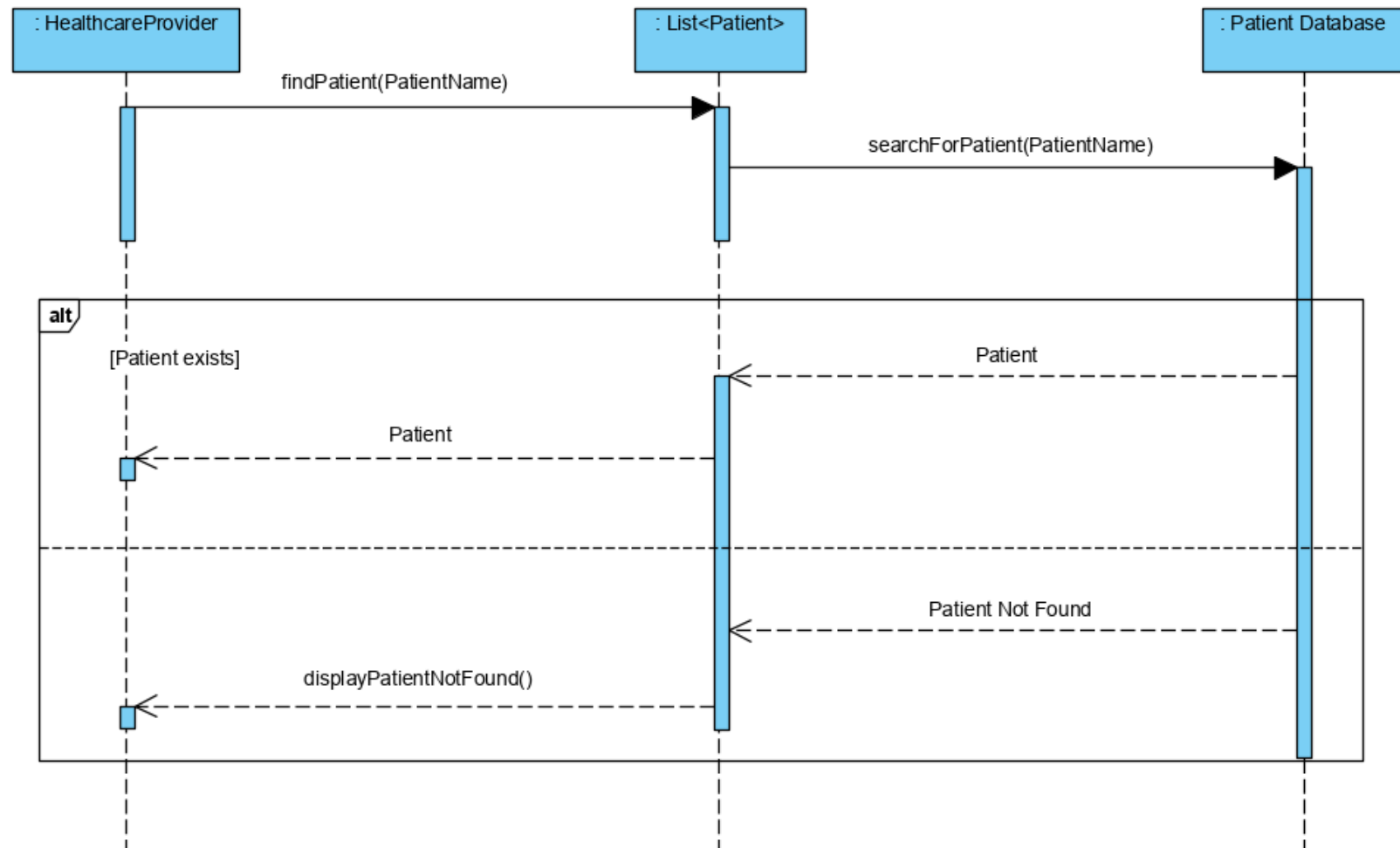
**sd** Diagnosis

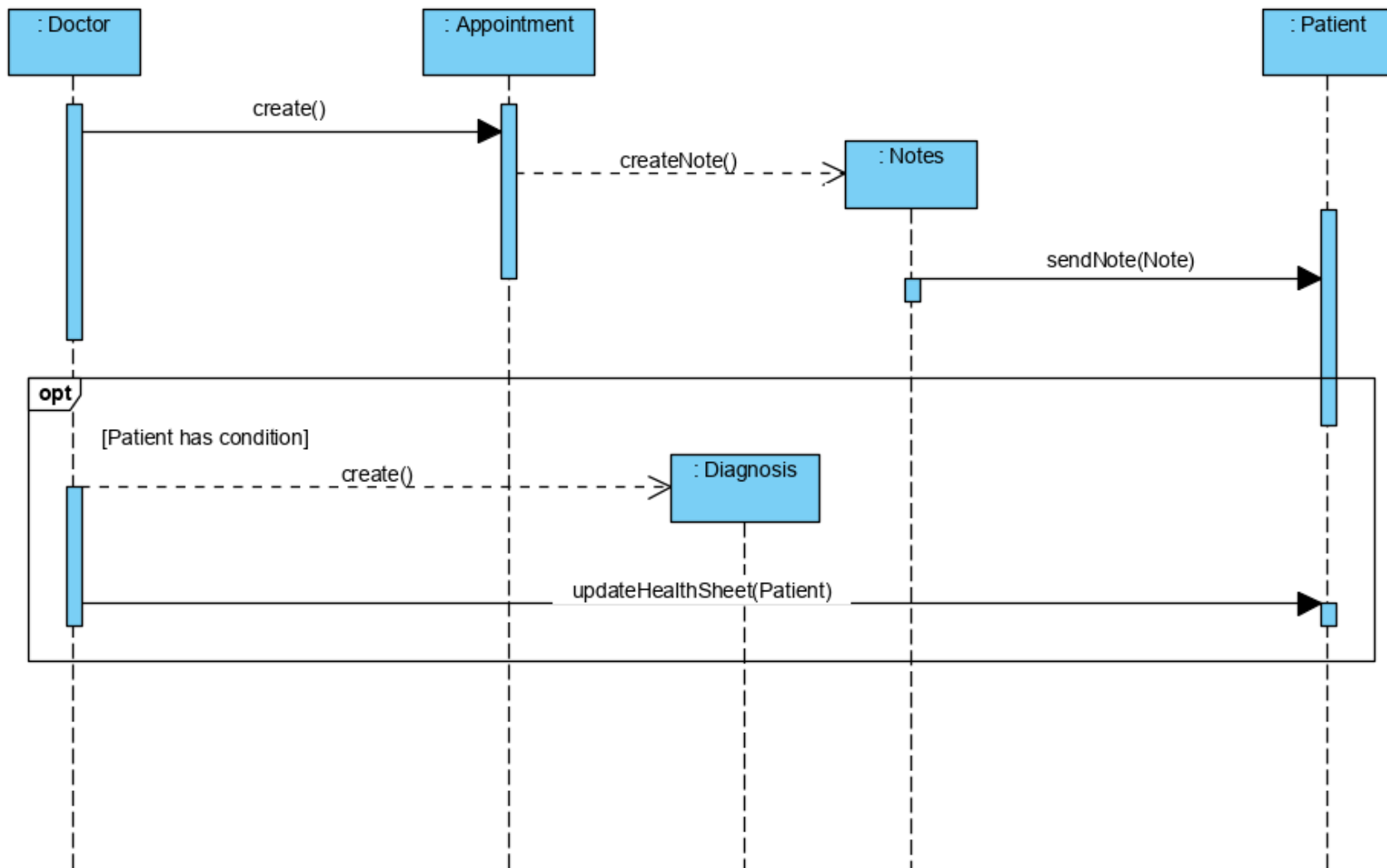


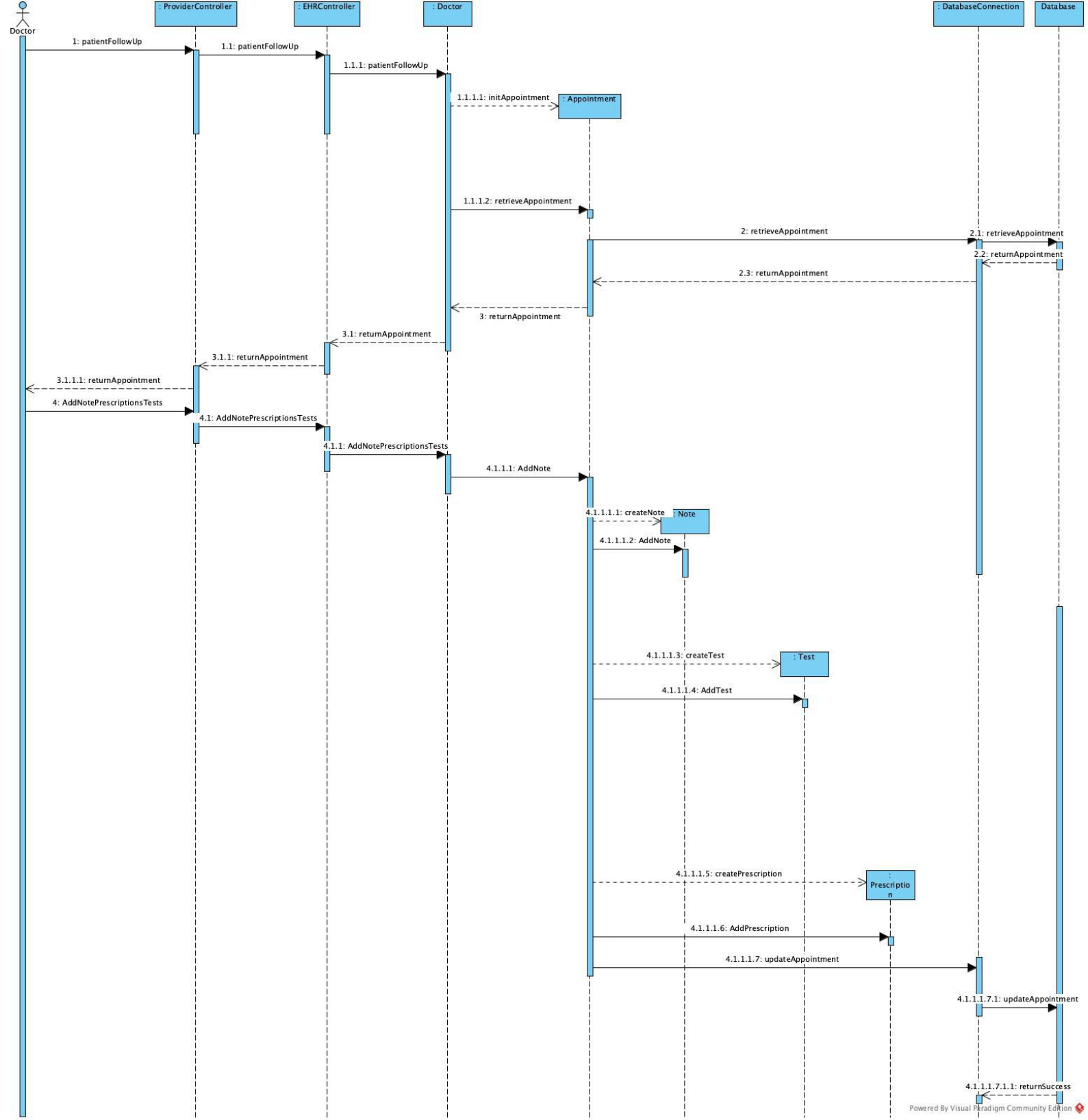


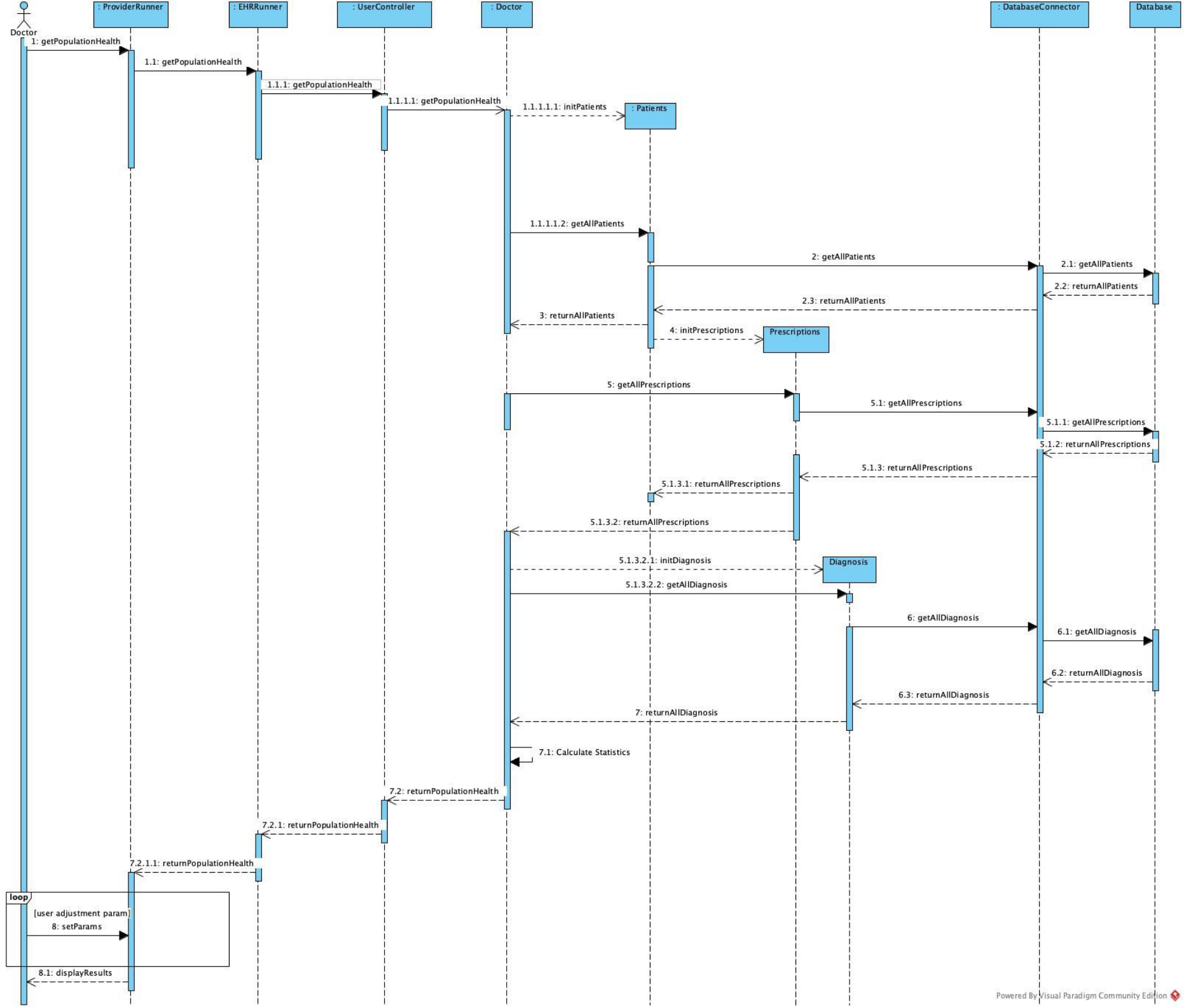


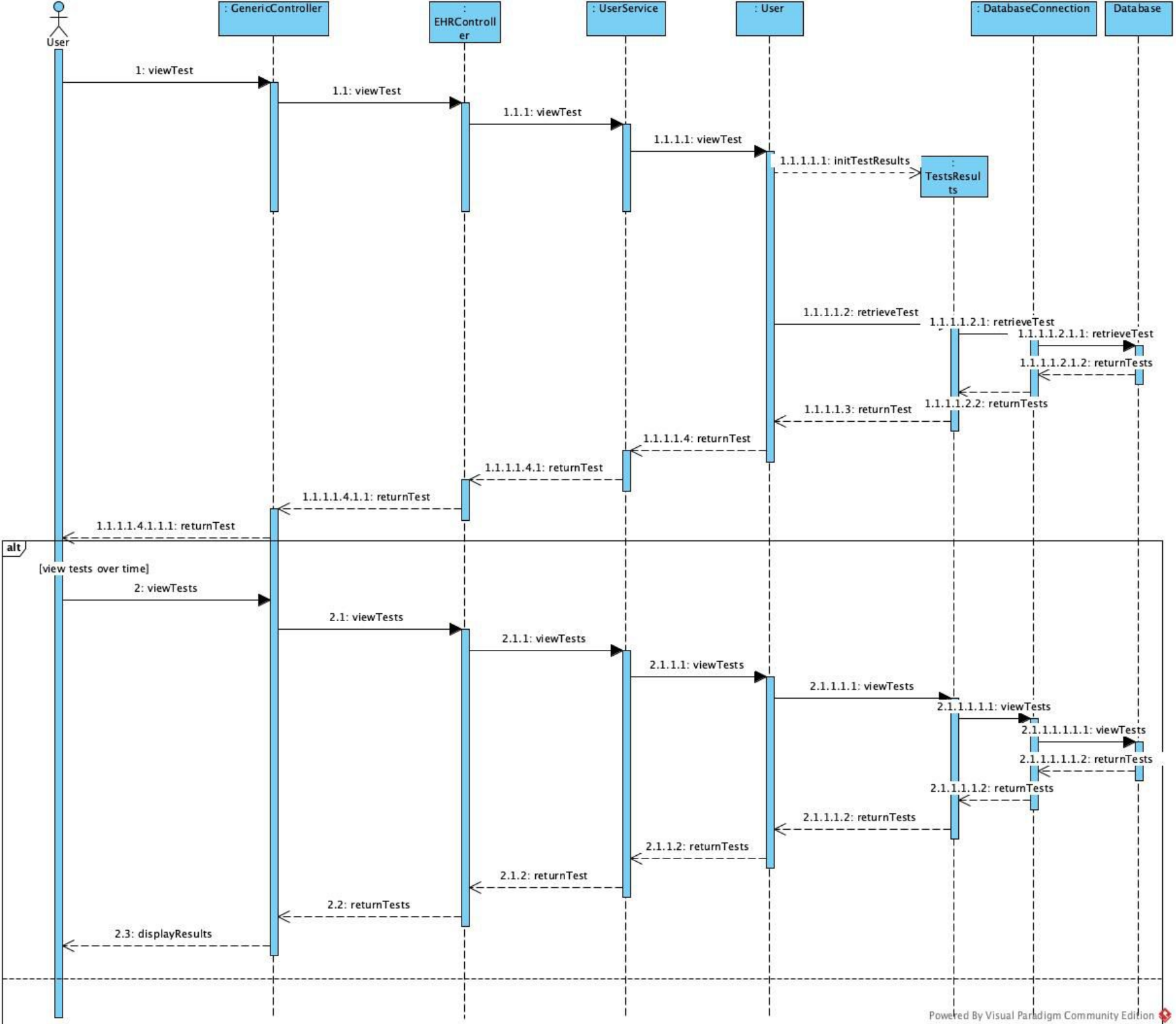












## Test Coverage Plan

- Doctor
  - Test that a doctor get data for a specific patient
    - Test that a doctor can get prescriptions
    - Test that doctor can order tests
    - Test that a doctor can get referral information
    - Test that a doctor can update a patients health sheet
    - Test that a doctor can diagnosis a patient
  - Test that a doctor can interact with appointments
    - Test that a doctor can view their appointments
    - Test that a doctor can create notes associated with an appointment
- Staff
  - Appointments
    - Test that a staff member can create an appointment
    - Test that a staff member can update appointments
  - Billing
    - Test that a staff member can bill a patient
    - Test that a staff member can correctly code insurance information
- Patient
  - Test that a patient can view an appointment
    - Test that a patient can access notes from appointment
  - Test that a patient can access diagnosis
  - Test that patient can access prescriptions
  - Test that patient can access test results
  - Test that a patient can their information
  - Test that a patient cannot access other patients information
- Prescriptions
  - Test that prescriptions can be sent out
  - Test that a prescription can be created
  - Test that a prescription can be stored
- Test
  - Test that a test can be created and linked to a patient
- Test Result
  - Test that a test can be linked to a patient
  - Test that a test result can not be updated after its creation
- Appointment
  - Test that appointments can be created between doctor and patient
  - Test that appointments can not be created if there is already another appointment at the same time
  - Test that notes can be associated with appointment
- Patient Diagnosis
  - Test that a diagnosis can be created for a patient
  - Test that a diagnosis can be modified and updated



- General
  - Test that viewing of screens is access specific
  - Test that information can be shared
  - Test that population health parameters can be generated correctly

GITHUB: <https://github.com/shenoisam/Software1/issues>

Timecard Report: <https://shenoisam.github.io/Software1/>

Linked Issue tracking system: <https://github.com/shenoisam/Software1/issues>

MAVEN demo: available within code on github

Suggested Point Redistribution:

At the moment we feel no need to redistribute the points as the group members have been contributing evenly.