CS 249 Happy Patients App

CS 249, Fall 2017, Final Project
Presented by -
Rashmeet Kaur Khanuja, Anusha Vijay, Steven Yen

Under guidance of -
Prof. Tanuja Phadke

Outline

- **Happy Patients** is an application that allows users to manage and view patient data.
- Users can add, delete, update, and retrieve patient information in this application.
- The application also uses Memcache to help improve retrieval speed so that not all requests requires a Cassandra query.
- A **Policy Server** maintains cache policies and users can retrieve and update the cache policies based on the need.
- **Message consumers** listens on a message queue (in ActiveMQ) for messages published by the Happy Patients app and runs appropriate routines when messages are received.

Technologies Used

- Jersey
- Cassandra
- ActiveMQ
- Memcached
- Jetty
- Maven
- Python
- Flask
- Javascript
- JQuery
- Ajax

What we already knew?

- Python Flask Server
- Jetty Container for Java Projects
- Maven Project and Dependency Management
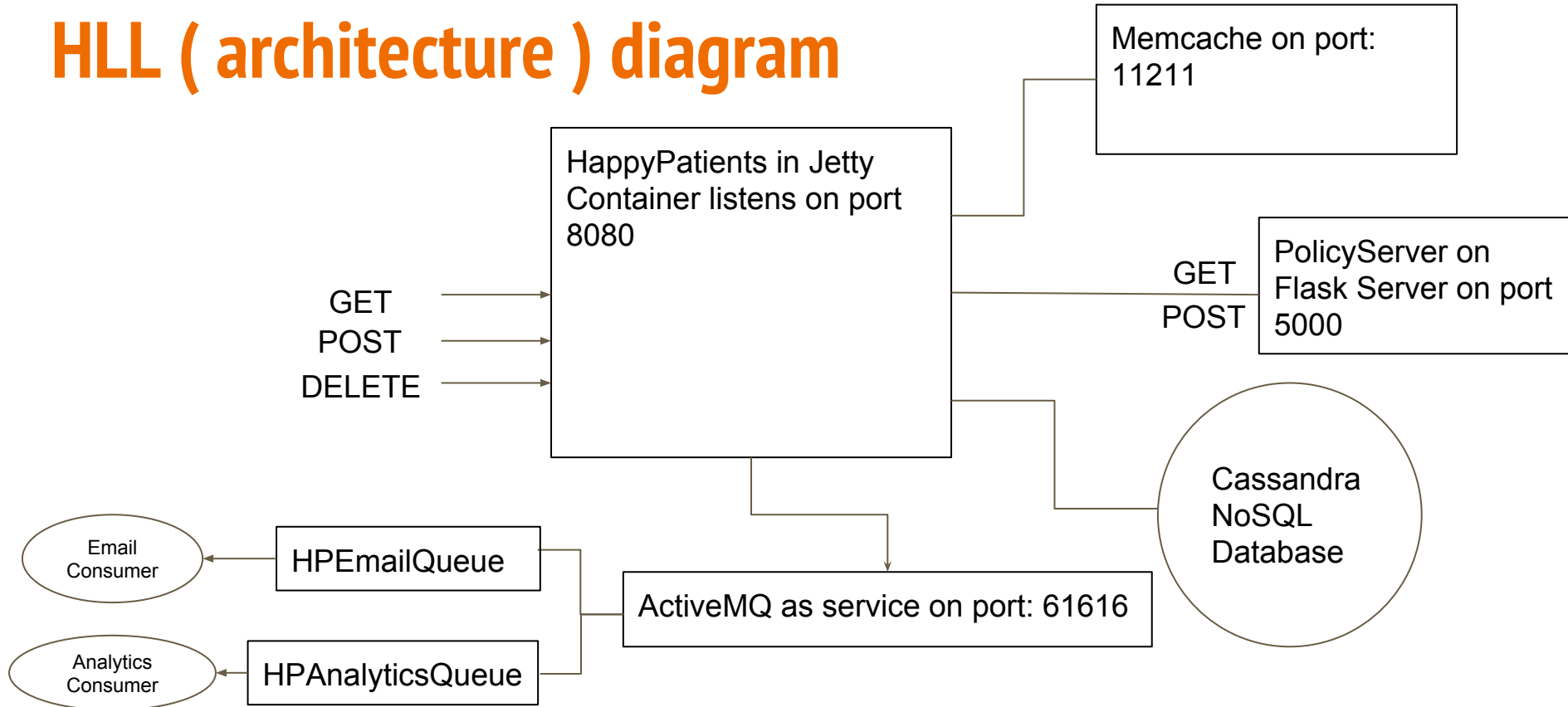
What we had to learn from scratch?

- Java Jersey Web Services Framework
- Connecting to Cassandra through Java
- Connecting to ActiveMQ through Java, and using JMS API
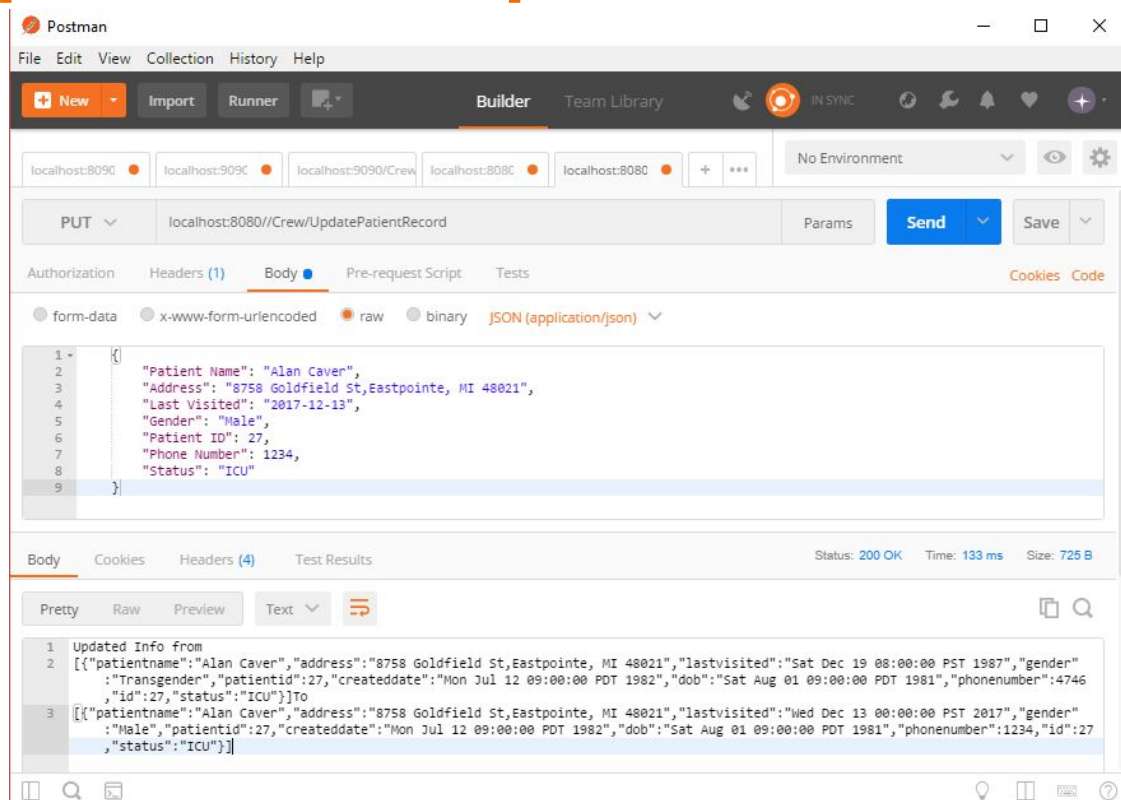- Connecting to Memcache

The POC we did

- Generated sample patient data as CSV file and import into Cassandra
- Used POSTMAN to send GET/POST/PUT requests to endpoints
- Created mock front-ends that generates HTTP requests and parses the responding JSON object

Happy Patients

Home

Welcome to Happy Patients

Your health is our priority

At happy patients, you can:

- View paitent info using patient id
- Add patient info
- Update patient info

HAPPY PATIENTS

View Patient Details

Add New Patient

Update Patient Details

COPYRIGHT CS-249 GROUP 2
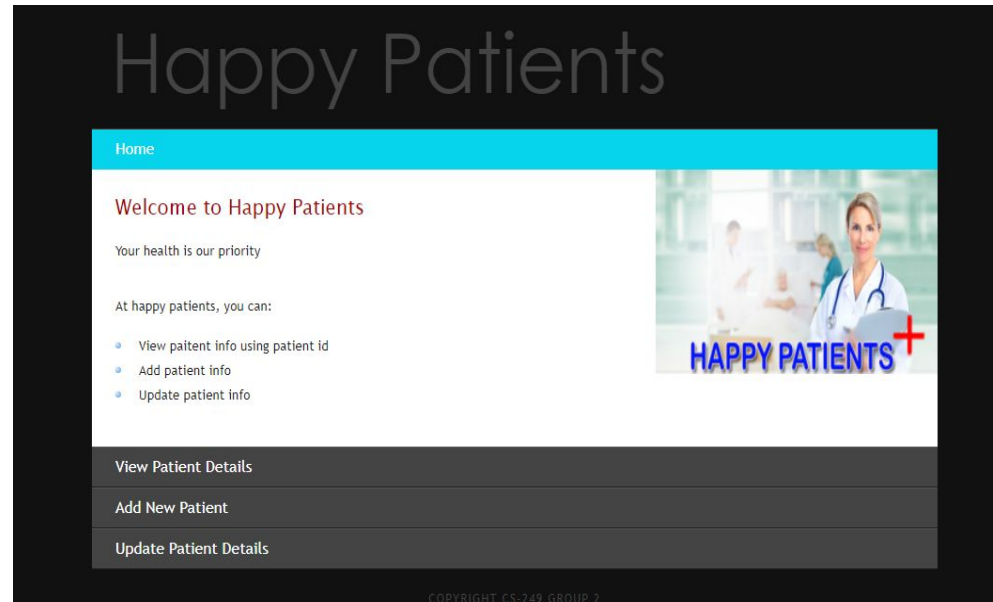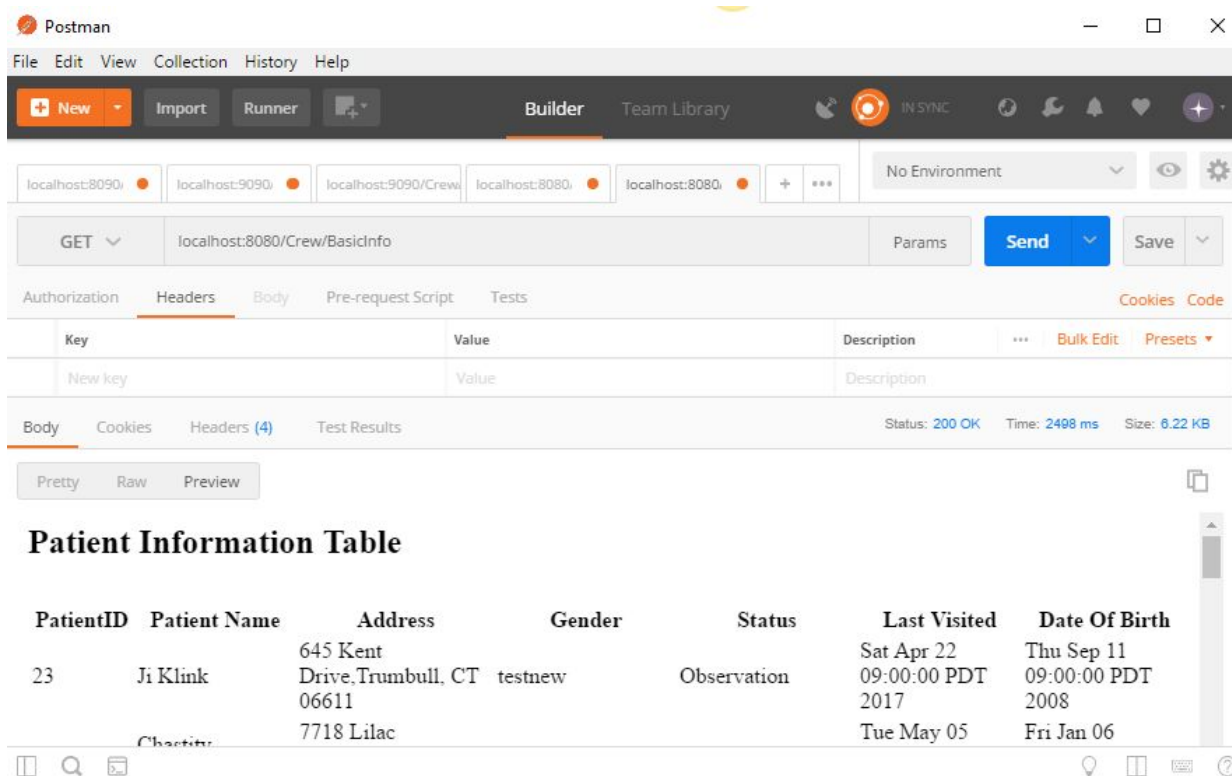
HLL (architecture) diagram

PUT Endpoint: /Crew/UpdatePatientRecord

GET Endpoint: /Crew/BasicInfo

POST Endpoint: /Crew/PatientInfo

POST Endpoint: /Crew/PatientInfoFilter

POST Endpoint: /Crew/NewPatientInfoRecord

DELETE Endpoint: /Crew/PatientInfo/{patientID}

PolicyServer end points:
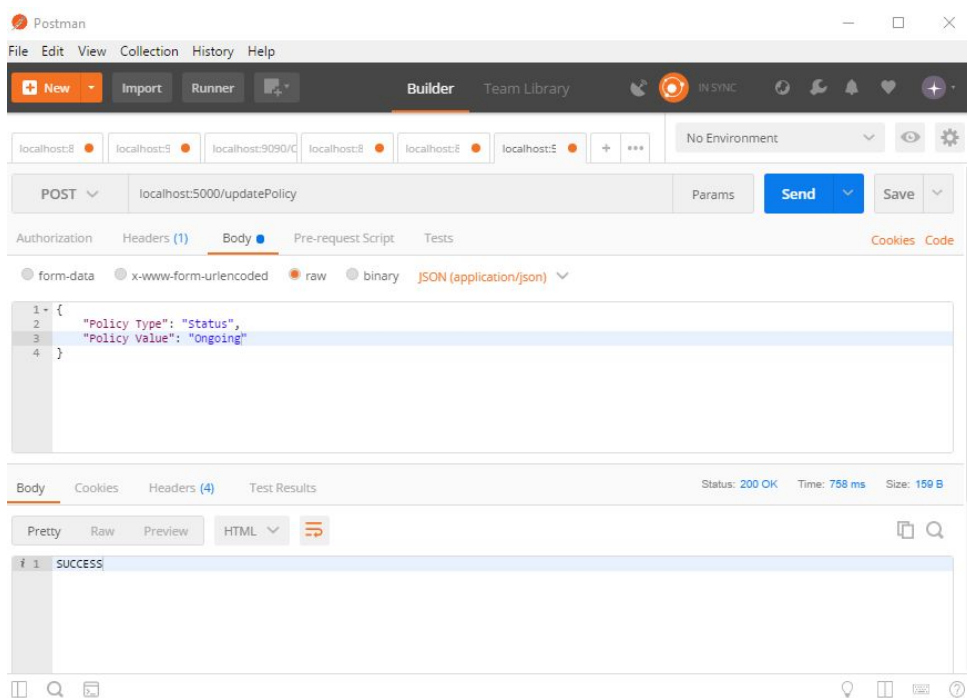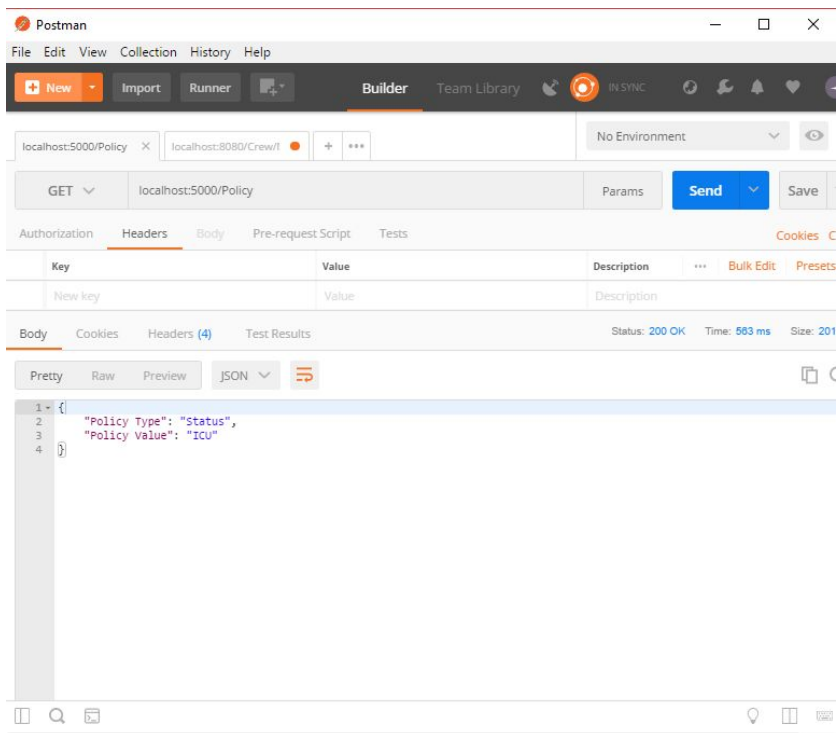
How we set up the environment?

- We set up the environment on both MAC and Windows
  - Install DataStax Community wrapper for Cassandra
  - Install Apache ActiveMQ as a service running in background
  - Download and run Memcache.exe
  - Do pip install for Flask for Python

Environment: Cassandra

Environment: ActiveMQ

Environment: Memcache

Memcahce telnet console:

What we referred to learn concepts

- Jersey example:
  - https://www.mkyong.com/webservices/jax-rs/jersey-hello-world-example/
- Java Cassandra interaction:
  - http://www.baeldung.com/cassandra-with-java
- Java ActiveMQ Interaction:
  - https://www.youtube.com/watch?v=yQcC6Z0PPes&t=879s
  - https://www.youtube.com/watch?v=3T-lDT-vALE
- Memcache Interaction:
  - https://www.tutorialspoint.com/memcached

Code walkthrough

- 3 Main Projects:
    - "HappyPatients" main App
    - "CS249PolicyServer" cache policy server
    - "hpMessageConsumers" email and analytics consumers

Thank You