HarperDB for Healthcare Team: Pangongso

Bridging the gap between challenges and opportunities for a better quality of life

Github:
https://github.com/ashkumar/HackOnHarperDB

Challenges and Solutions

❖ Healthcare is essentially a collaborative effort between several stakeholders with different roles. We chose enterprise grade Liferay Portal and HarperDB to facilitate this.

❖ Given the dynamic nature of collecting and updating patient data, we used Angular within Liferay and HarperDB to respond to these changes with unprecedented agility.

❖ We devised a means of populating HarperDB with data from various Healthcare monitoring systems. The resulting database events were used to collect various other details about the patient, invoke rules and conditionally push real time alerts to web browser based clients.
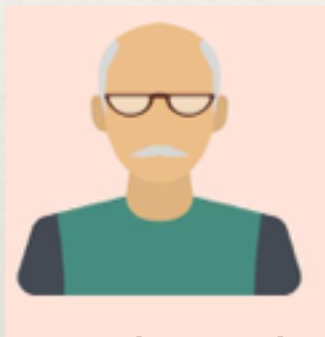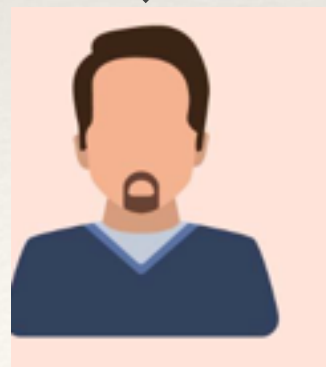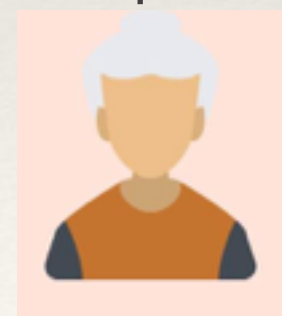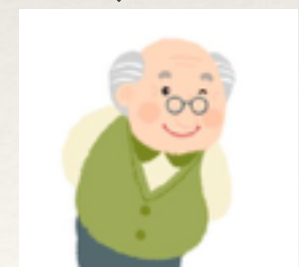
Doctor

Nurse

Caregiver

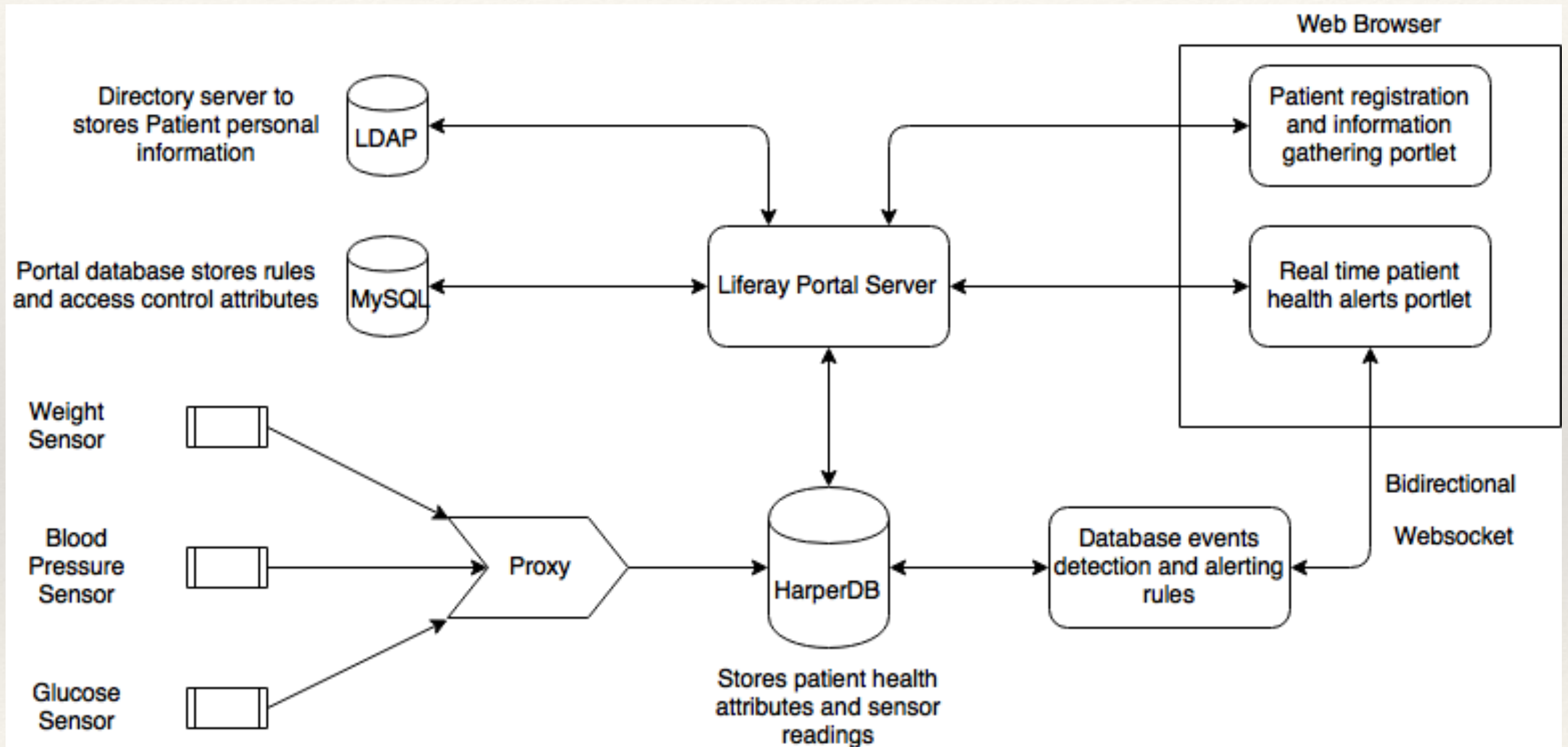Nutritionist

Patient

Medical Devices

Son

Spouse

Brother

YOGAFITNESS

System Components

Information architecture

- Static data that almost never changes

  - Personal identifier data in LDAP used by Liferay identity management

  - Roles, access control, collaboration, assets and content stored in Liferay database

- Semi Static data that changes at most 10 times a year

  - Patient information like allergies, pains, medication, diet, exercise care plans stored in HarperDB (schema: patient table: patientattributes)

- Highly dynamic data updated at least daily

  - Sensor data stored in HarperDB (schema: sensor, tables[weight, bp, sugar]

Angular Portlet (HarperForm)

❖ Patient data collection form designed for extremely agile initialization and modification. It also provides very simple and secure server side REST calls (HarperFormReadService.java & HarperFormService.java)

❖ Adding or deleting fields require only model changes in question.service.ts and redeploying the portlet

❖ Given the schema less features in HarperDB, we are able to save the state of un submitted forms. This is not possible in out of the box Liferay forms and even Google Docs. This feature is critical for complex forms and is a very big win for all solutions on Liferay.

Data acquisition (HarperSensor)

❖ This is a HTTP proxy built on express.js

❖ It's role is only to transform the payload from what the sensor sends to a format that can be used by HarperDB to insert records in the database.

❖ A minimal {"measure":"weight", "userid": 35713, "sensorid": 4, "reading": 150} from sensor is transformed into a valid insert query for HarperDB

❖ From the message, the proxy identifies the correct table to add the sensor data to and also adds a unique identifier (hash) defined by the current epoch time as one of the columns in the table.

❖ This epoch time value not only provides an effective unique identifier in this setup but would also facilitate good time series analysis in future.

Real time alert notification (server)

- This program provides real time alerts to client based on predefined rules and sensor data as they become available.

- A socket.io connection is established when a browser based client first connects to the server.

- A chokidar instance is initiated only once for all clients.

- Events in HarperDB are detected upon database inserts and the following occurs

  - user id is read from the new file that got created.

  - for that user, the latest reading for weight, sugar and bp are read from corresponding tables.

  - Rules are invoked for these readings

  - The browser clients are alerted right away if necessary.

Conclusion

- ❖ HarperDB, Liferay Portal, Angular and <u>socket.io</u> are ideally suited to build enterprise grade solutions in support of highly collaborative, agile and responsive communities in the healthcare domain

- ❖ The solution provided promises to dramatically improve the quality of lives of millions of patients.

- ❖ HarperDB design has enabled innovation at the grass roots level. This can be a catalyst to transform the healthcare experience for all stakeholders.