README_BLOODPOOL

System Architecture of BloodPool

BloodPool is designed on a MVCS design architecture, where MVCS stand for Model, View, Controller and Services respectively. A Google Cloud Datastore serves as the NoSQL database entity model.

In Java EE, BloodPool uses these packages:

1. com.bloodpool.controller
2. com.bloodpool.dao
3. com.bloodpool.service
4. com.bloodpool.entity

1. Controller Package: This package deals with all classes that handle the requests and responses with the user when he/she interacts with the UI directly. It directs work to different services.

2. Dao Package: This package directly with the datastore and fetches the data and saves the data.

3. Service Package: This package does all the servicing of the website.

4. Entity Package: This package contains two Entity classes that serve as the two table; UserDetails and PatientDetails.

Selecting Suitable Donors

When the request is raised our algorithm checks for all registered users that fulfil these conditions: -

1. Is the user willing to donate blood?
2. Is the user aged above 18?
3. Is the user's donation status good?
4. Is the user capable of donating to that patient?
5. Is the user located within 5 km of the patient's hospital or not? (This radius is increased to 10km and so on if no user is found)

This generates an ArrayList of selected users. The selected users are then notified via an email and push notifications. These push notifications also reach the Raspberry Pi devices installed at Blood Banks.

Technologies used for BloodPool

BloodPool uses these technologies:

1. Java EE
2. Google Cloud DataStore
3. JavaScript
4. jQuery
5. Java Server Pages