

README

Technology used:

- REST services – Jersey.
- Front End : HTML, CSS, jQuery, JavaScript

Required installations:

- MySQL, Apache Tomcat.

Deploying the services:

- Run the .sql scripts under “sqlScripts”
- Copy the war price-poller.war and deploy it in Apache Tomcat server.
- Once the services are deployed, supply the appropriate credentials required in db.properties(to establish db connections) and credentials.properties(used for email client).
- The application can now be accessed at <http://localhost:8080/price-poller>
- Configure the price-poller service running at <http://localhost:8080/price-poller/services/pricePoller> as a scheduled job running at **X** interval.


REST Services:

- *Subscription Service* – The service that creates a new subscription for the given user and the product.
- *Product Service* – Fetches the list of all products as a JSON.
- *Price Poller Service* – This can be configured as a job that can be set to run at specific intervals.

Tasks done by the PricePoller job includes,

- Polling the zappos API for the prices of the products in the database.
- Updating the prices in the database.
- The trigger written in the database automatically populates the NotificationLog table with the subscriptions of the products that has a increased off and also has a minimum of 20% off its original price.

Screen shot:



Price watcher

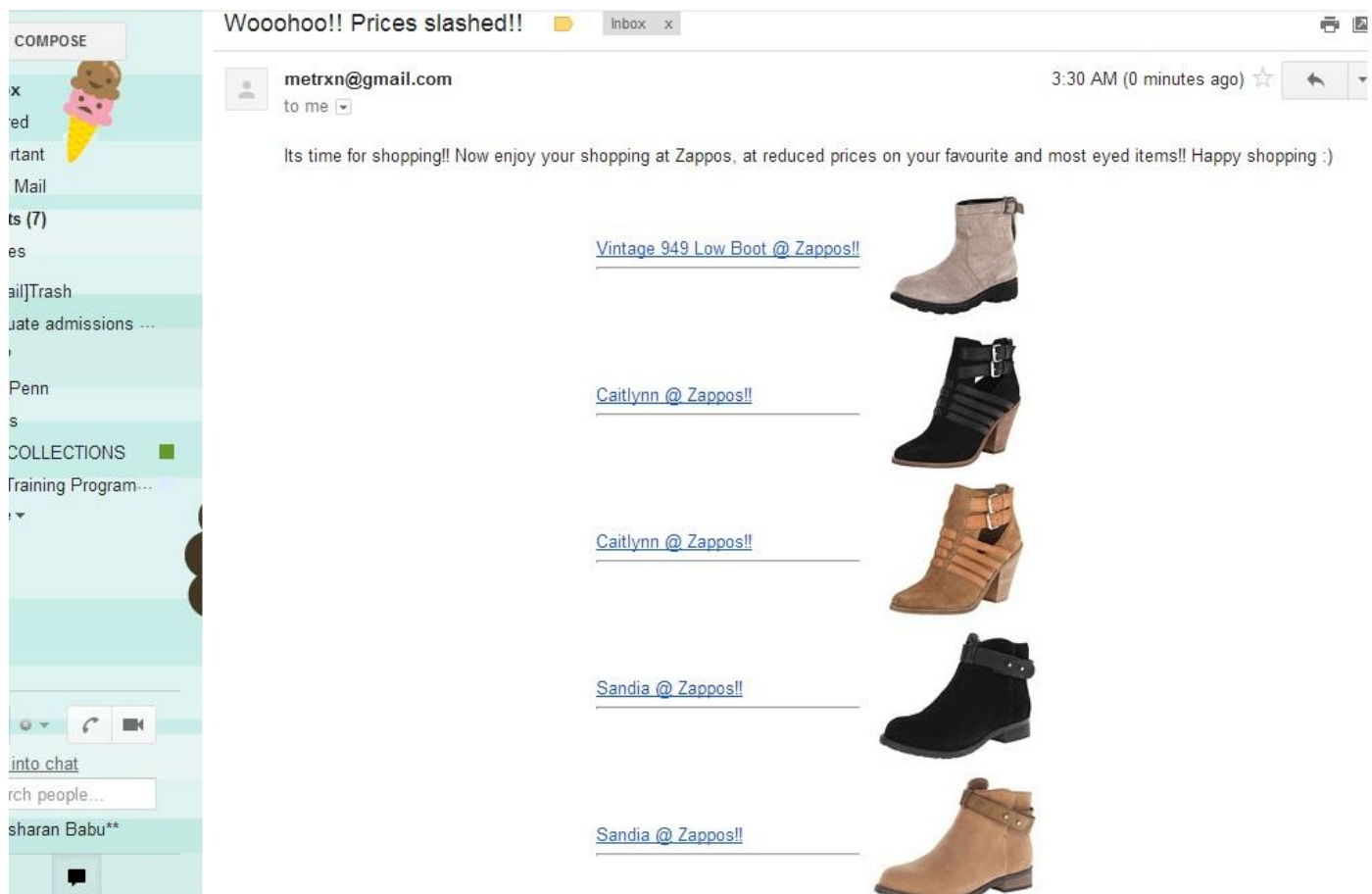
Email-Id

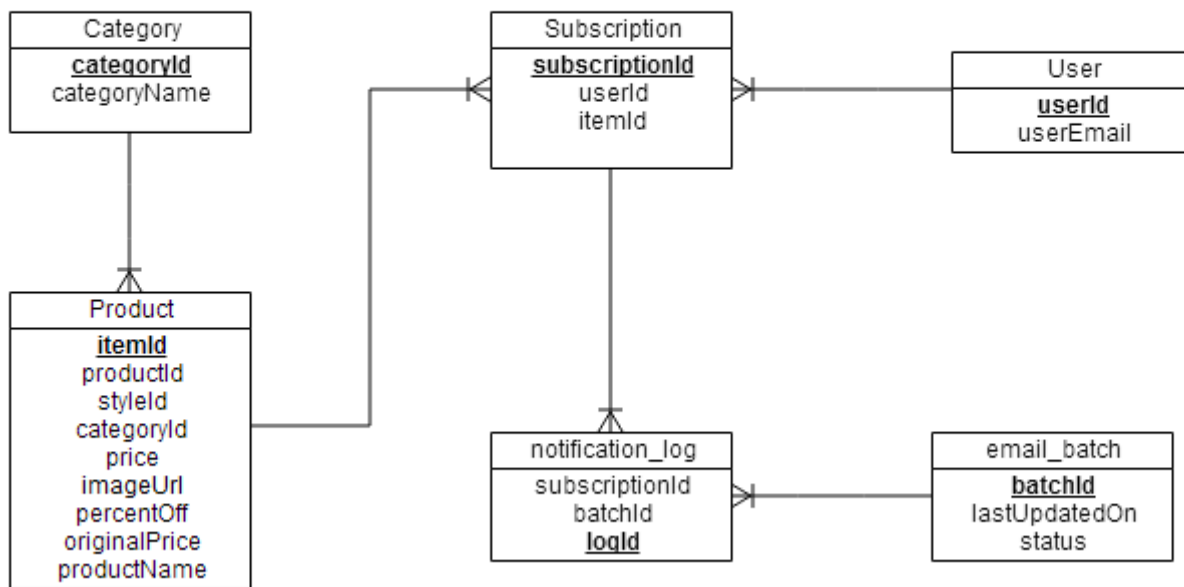
Product-Id

Successfully subscribed

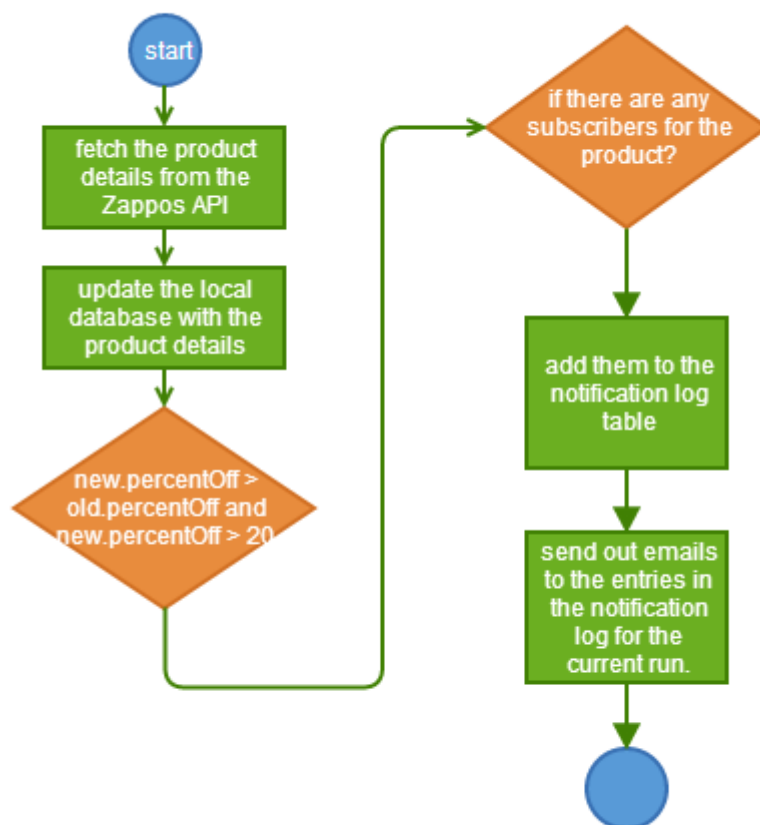
Subscribe

Sample Email screen shot:





PRICE POLLER



Note: This service is scheduled to run for every X hours, ie., works as a polling job

SUBSCRIPTION WORKFLOW

