**Zappos Price Notification Application**

**Description:**

1. Developed GUI using JFrame.

2. Used different files for different functions like invoking Zappos API, Java objects, Adding product details in json to a text file(db), constants etc. for easy maintainability.

3. Used two runnable files

PriceNotificationFrame.java

* + - This is used to create the UI for the user to enter the product and email id
    - Used Gson library to convert Json Object to Java object.
    - User can first enter product that he is trying to add to favorites, once search icon is clicked the product details will be displayed along with the image. Email Id and Notify button fields are displayed. When Notify is clicked the user details and respective product details are added to a Json file.
    - Empty and email validations are handled.

ScheduleMailing.java

* + - This is used to send mail notifications to the users. It is scheduled to run once for every 2 Hours.
    - Once an email notification is sent, the user will not be notified again when it runs for the second time. Handled this using Notify flag.
    - Used SendGrid API for email notifications(Send Grid account credentials are included)

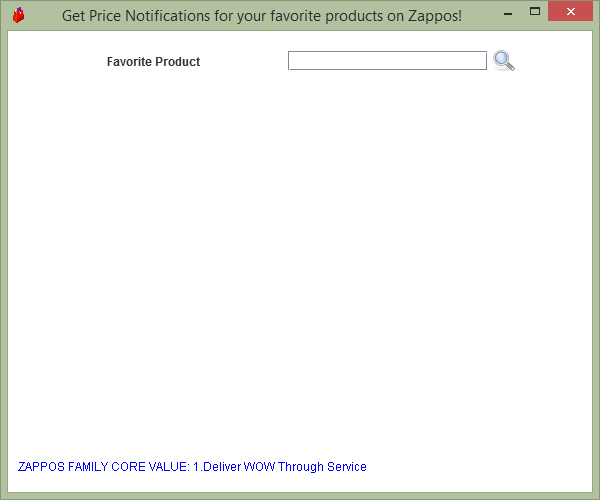
4. I had an issue in creating runnable jar files due to images. Please execute in Eclipse or cmd prompt.

**Steps to execute the application:**

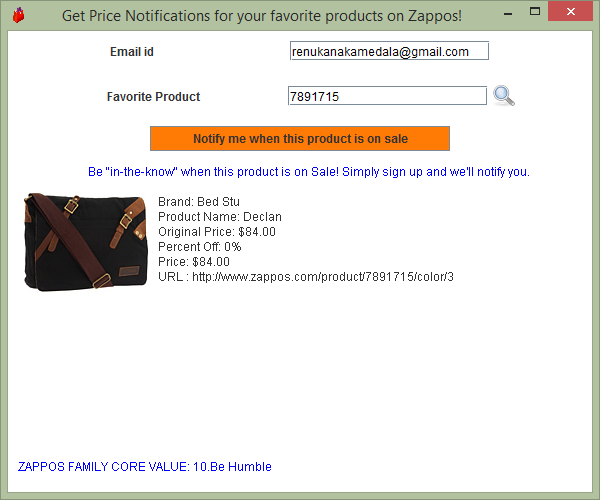
* Unzip the folder attached.
* Import it into eclipse as a Java Project
* Run "PriceNotificationFrame.java" as a Java Application
* Enter the product id --> Enter the email id -->Click "notify me" button.
* Run the "scheduleMailing.java" as a Java Application
* Check your email to start partying!

**Design :**

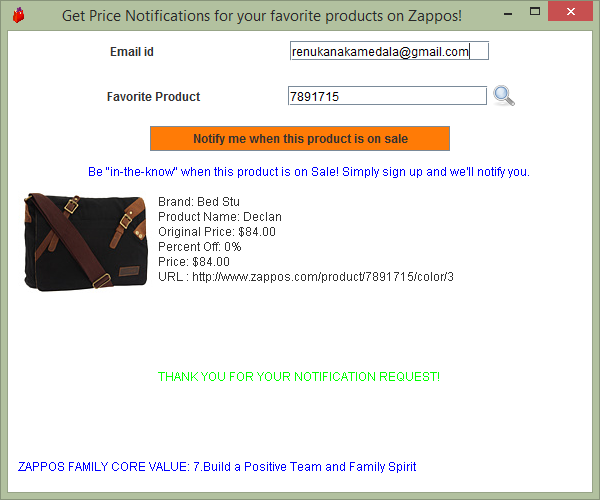
Below are the screenshots from the application.

****

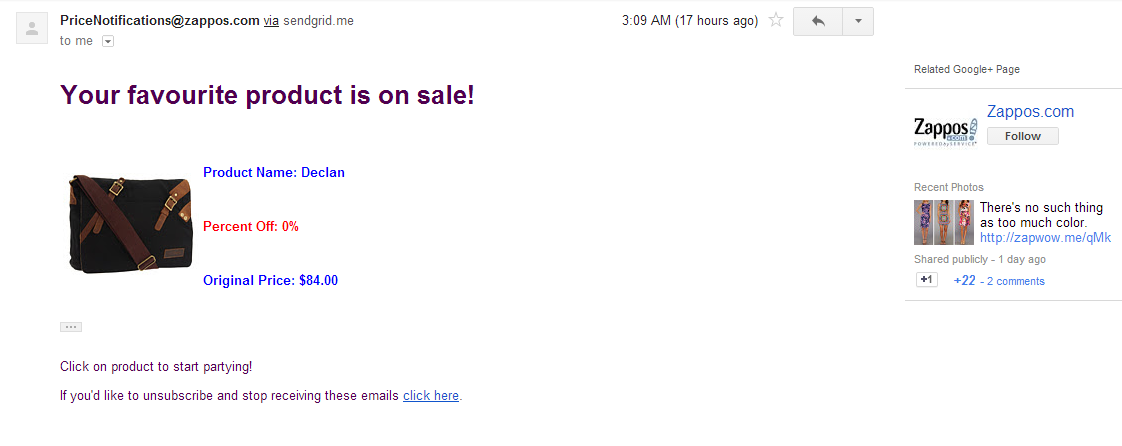
Once the Product ID/ Name is entered and clicked on search button the below screen is displayed in which user can enter email id and click on 'Notify me' button.

****

**Once user clicks on 'Notify Me' the below screen is displayed.**

****

**Sample Mail Notification:**

** Files Used:**

PriceNotificationFrame.java

PriceNotificationInvokeZAPI.java

PriceNotifProdDetDTO.java

ProductSearchResponse.java

AddToFavorites.java

FamilyCoreValues.java

FamilyCoreValuesResponse.java

scheduleMailing.java

NotifyUsers.java

ZapposConstants.java

**Database/Storage Files:**

Add\_Favorites.json

Notify\_Favorites.json

**Images:**

searchIcon.png

windowLogo.png

**Future Work:**

As of now the in the application, percent-off checking is done without connecting to Zappos API considering the traffic to the API. But it can be done by fetching the current Percent-Off field from Zappos API for that particular product and cross verifying with the Percent-off in the database.

**Note:**

When API reaches the max limit of hits, the application generates an exception. In such cases, please change API Key in ZapposConstants.java