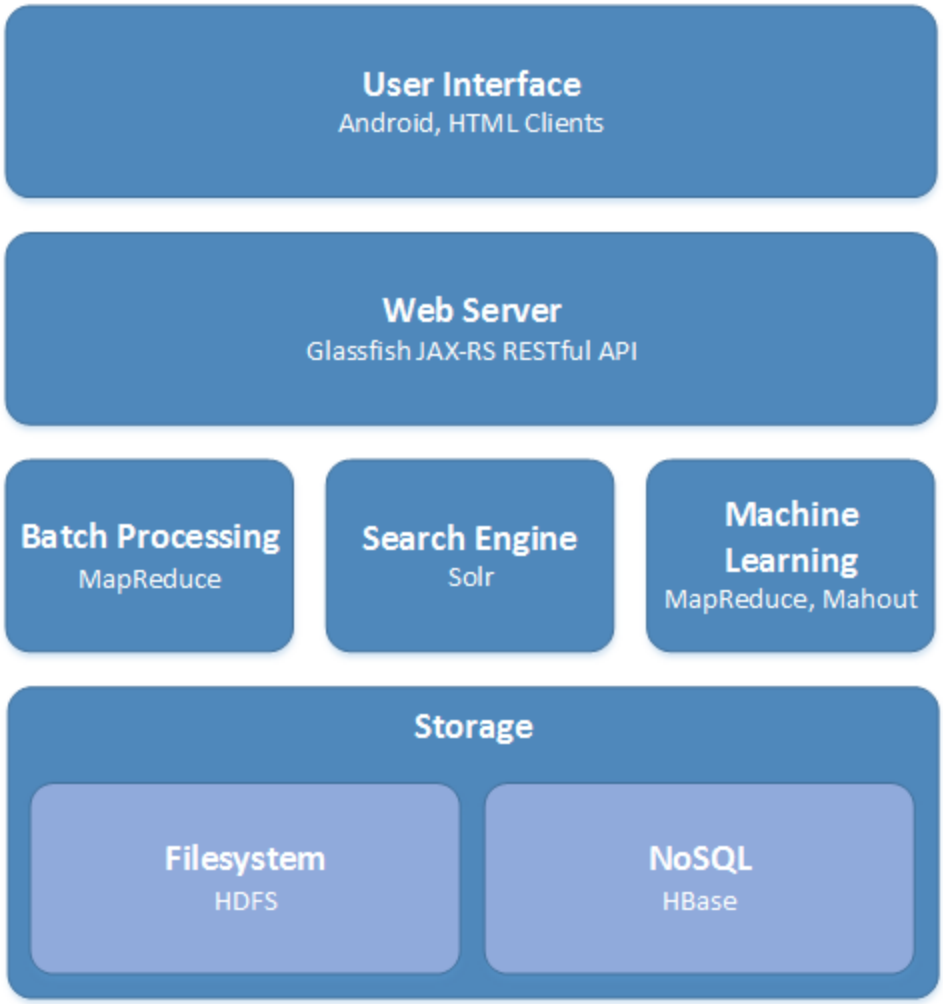
Increment 1

Group 6: LeeStudents
James Clark
Tirumala Konireddy
Jagadish Tirumalasetty
Wang Zhang
Dr. Yugyung Lee
CS 560: KDM
March 11, 2014

Framework Specification

System Architecture Diagram



Domain Model

Data Sources

- Consumer Product Safety Commission. The sources come from consumer product safety commission government website, it contain all kinds of categories recall records. http://www.cpsc.gov/en/Newsroom/CPSC-RSS-Feed/Recalls-RSS/
- General Recalls. The sources come from USA.gov website, it contain child satety seat recalls, food recalls, drug recalls, tire recalls and some other recalls information. http://www.usa.gov/Topics/Reference-Shelf/Libraries/RSS-Library/Consumer.shtml#Recalls

- Food and Drug Recalls. The source come from US Food and Drug Administration website, it contain food and drug recalls. http://www.fda.gov/AboutFDA/ContactFDA/StayInformed/RSSFeeds/Recalls/rss.xml
- Consumer Product Safety Information Database. The source come from saferproducts government website. http://www.saferproducts.gov/WebApi/Cpsc.Cpsrms.Web.Api.svc/
- National Highway and Traffice Safety Information: safecar.gov http://www-odi.nhtsa.dot.gov/downloads/flatfiles.cfm

The finally converted structured data is stored in the server in the form of a json table as below. In this table, one dynamic column is allotted for every category. The category name or id and alerts were stored in the first rows of every column respectively. One dynamic row is created for every user in which the first column contains the username or id, and parity bits '1' or 'o'. ('1' under subscribed category column and '0' under unsubscribed category or '0' is the default parity for every category for user unless user is subscribed to that category. If a query is received by server to retrieve a particular user alerts, it first searches in the first column for username or id and then send the alerts stored in the second row for the subscribed categories(categories whose parity bits are '1' in the username row) or search category.

Fig. 1: Json array table

Category	Climate	Honda	Windows 8
Alert	1. Today Temperature 15^0 C. 2. Showers from 06.00AM to 10.00AM. 3. Severe Thunderstorm alert in evening.	1. Speed limit 35mph in today's snow. 2. Park your car in some hide.	1. Upgrade your software Windows 8.1. 2. New features were added to ms word.
James	1	1	1
Tiru	1	0	1
Wang	1	1	1
Jagadish	0	1	0

Methodologies

Front End Methodology

1. Get User credentials from user.
2. Send a query about the requested alerts through webservice to data-base server.
3. Retrieve the whole json array of alerts using hadoop file system that were stored in the server under that user credentials.
4. Display those alerts on the user home screen and now user can see all his alerts.
5. If user request any new category alerts through search, the app sends a query of 'search text' to server through Solr to analyze search text and then hadoop file system to is used to search the category in the server to retrieve that subscription array.
6. If user wants to subscribe that service, then that service is appended to the user's json array in server using mahout and from then the that service alerts were synced with the recent recall alerts that were shown in home screen.

Back End Methodology

1. Our system keeps on updating the rss feeds and websites data document(google document) every day for any appended data to them.
2. If any new data or information is detected, our system stores that data into a temporary dynamic variable.
3. Run adaptive decision tree using mahout where we use Case base reasoning and naïve bayes classifier to check whether there is any relevant information to alert or not.
4. If the decision tree returns a alert of some category, store that alert in server of json array.

Algorithms

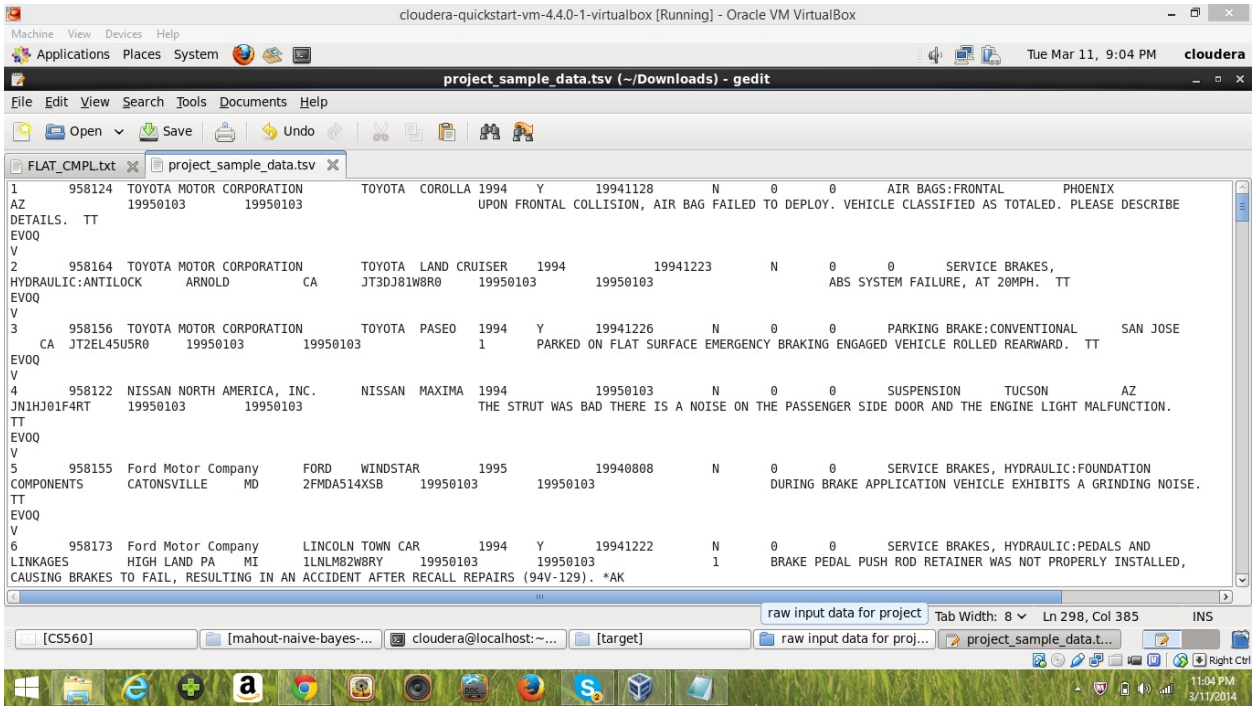
It makes sense to begin work using a naive Bayes classifier while continuing research in new algorithms and systems that improve relevance.

Analytic Tools

1. Mahouth: For implementing Adaptive decision tree, Case base reasoning and naïve bayes classifier.
2. Hadoop: For using distributed system to run Mapping, reducing functions and store data into server(json table).
3. Solr: To map user entered search text to relevant category while searching and retrieving data from server. We may use it for analyzing data from rss feeds and html files.

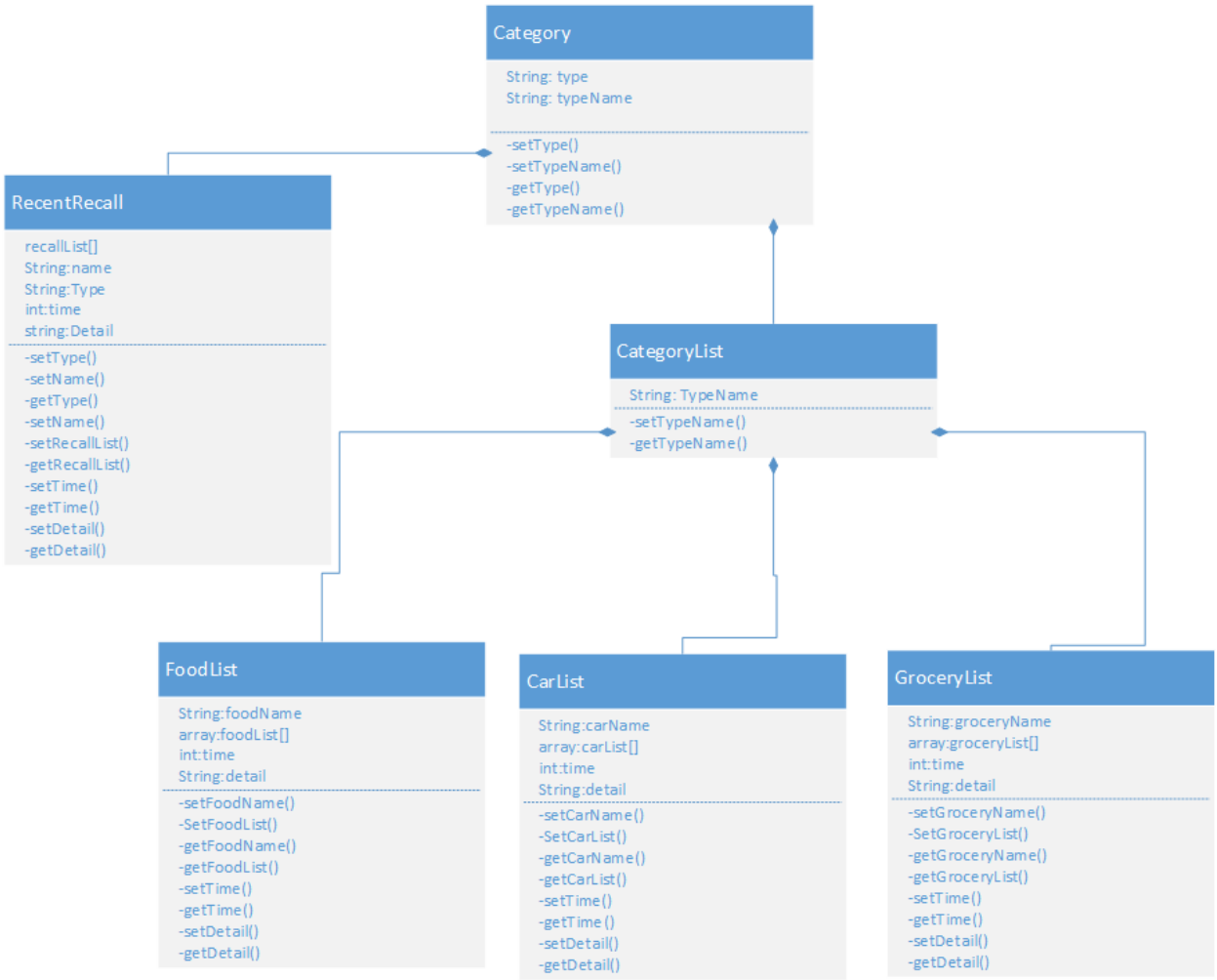
Analytical Tasks

1. Collection of data : We get data from http://www-odi.nhtsa.dot.gov/downloads/flatfiles.cfm about all the product recalls that had done in past till now. Now, we are manually monitor and download the updated recall data flat .txt file.

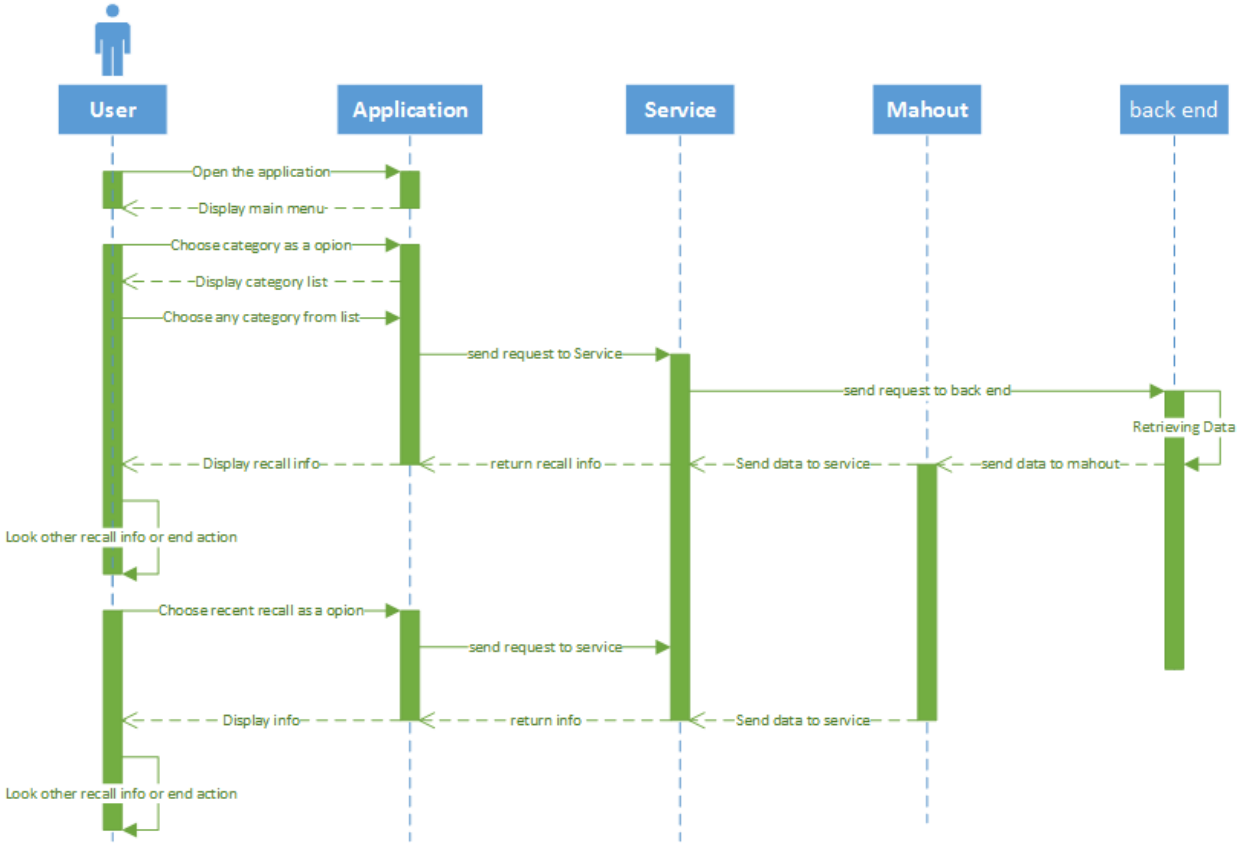


Application Specification

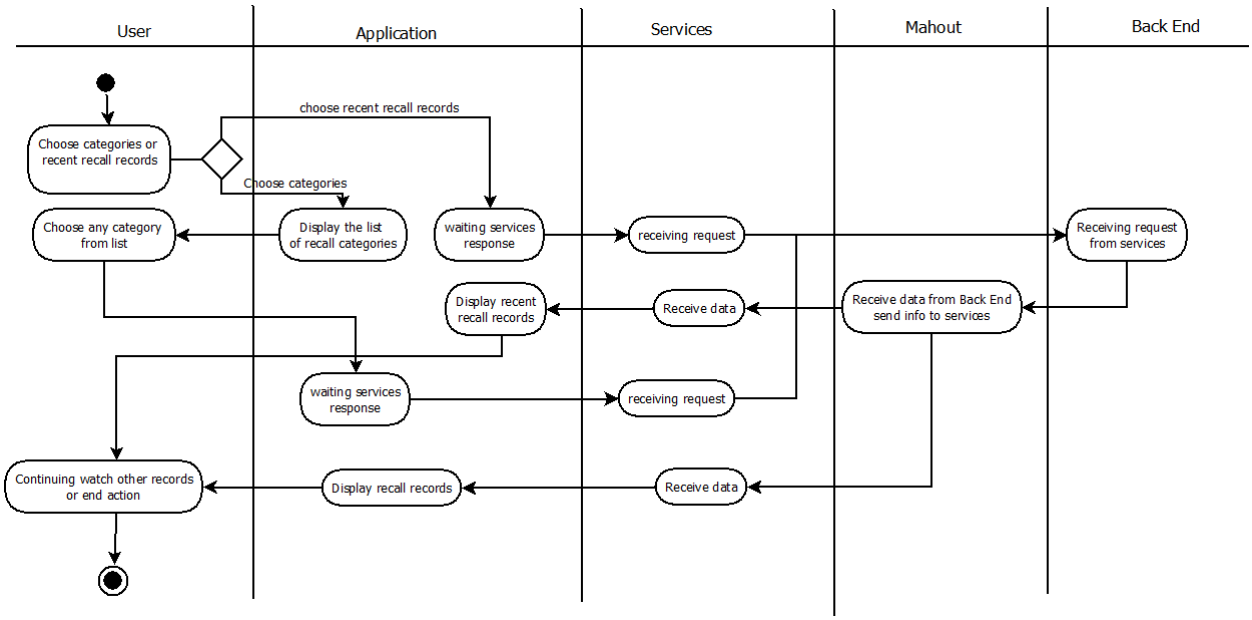
Class diagram



Sequence diagram



Activity Diagram



Service Specification

Base on the services (CPSC, NHTSA, FDA, and USDA), we will store these data as XML or Json format, then we will extract recall information from services, such as the name of recall items, the date of recall items, the reason of item recall and description of recall items. as output we will display exact recall information.

Design of Mobile Client

Our project Application is Android based. For our app GUI, it contains two main feature. One is recall categories, in this part, we will separate each category base on our data for user convenience. Another is recent recall records. In this part, our App will display most recent recall records. Further more, we will add a search bar to enable user search whatever they are looking for. Also we may implement a user login function and registration function.

Implementation

Implementation of data model and algorithms (Machine Learning)

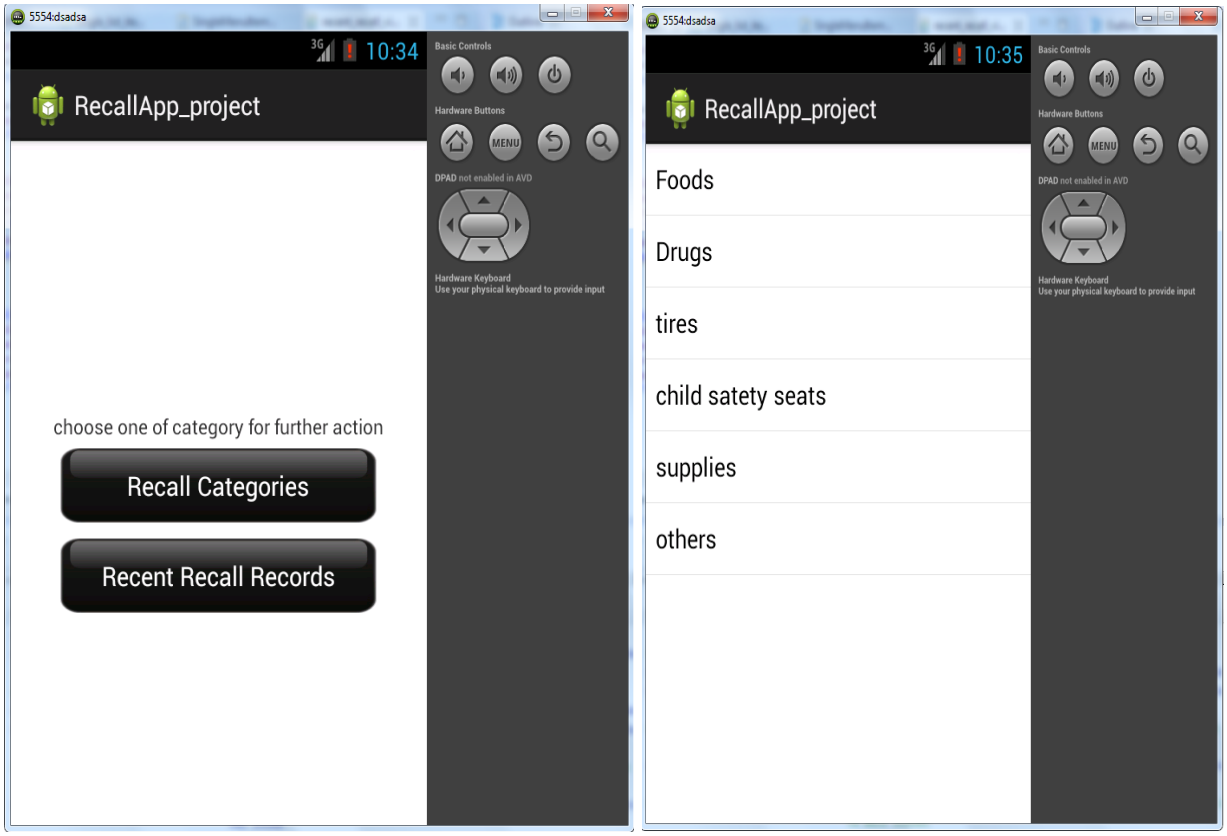
We are deficient in machine learning algorithms as we continue to accomplish systems administrative tasks such as choosing a cloud server solution, virtual machine deployment, and configuration. We plan to resolve these administrative tasks and produce machine learning results for increment two.

Implementation of user interface (Mobile Apps)

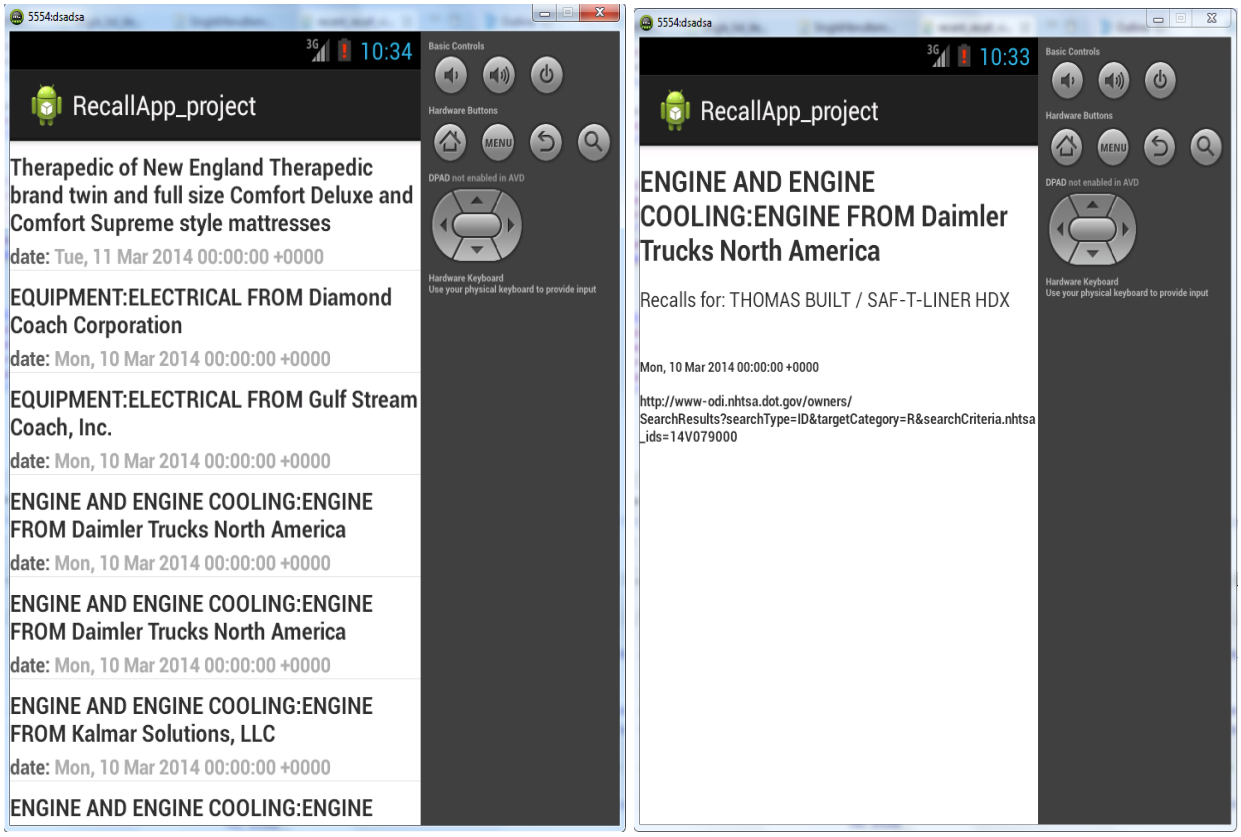
The recent recall records part is most likely done. The recent records will display as listview. For each record, only display title and recall date. User click on each record, it will jump to detail page, in this page, user will be able to view item name, description, recall date and link. For the recall categories part, it only has list of category for now.

Documentation

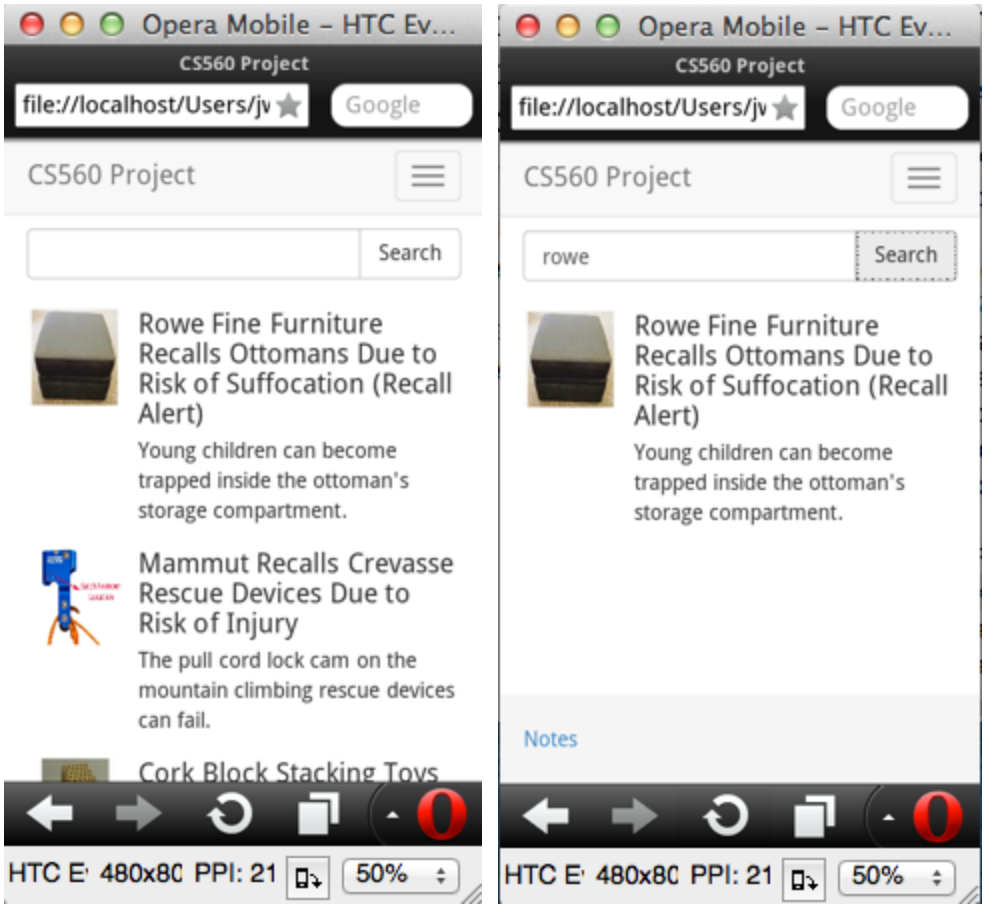
Screenshots and explanation on the design, modeling and implementation.



The first screenshot is our app main menu. The second screenshot is category list, it shows as listview. The third screenshot is recent recall records view. in this view, it displays most recent recall records by using XML parser extract data from web. The third screenshot is single detail page. In this page, it displays the recall item name, description, recall date and link.



Next page, separate but additional GUI implementations demonstrate query capability against Solr server. To accomplish this, we deploy a proxy Glassfish web server running JAX-RS restful web services to make calls to Solr. An HTML client uses cross browser Javascript to make AJAX calls to the Glassfish server.



Project Management

Project Timelines, Members, Task Responsibility

Implementation status report

Work completed

- Wang: start to working on recall categories part of app interface and search function. it may take about another 12 hours.
- Wang: Finish the recent recall records part of app interface,12 hours.
- James: Integrated Glassfish 4 web server, Solr server, Solr HTML client, 25 hours.
- Tiru: worked on Solr server, found more data sources, researched how to collect data into our system and studied machine learning algorithms to analyze data, and managed Scrumdo.

Work to be completed

For our next increment, we intend to bring our platform to the cloud and integrate a workflow from data collection to analysis, to publishing our results for availability by search server.

Deployment

At this time, we do not have internet-facing public deployments. Our work is limited to local virtual machines as we continue to learn the Cloudera CDH system.