Android Applications Improvements Based On Conference Navigator

Independent Study Report

INFSCI 2950

Spring 2015

Author: Zhu Wang

Supervisors: Peter Brusilovsky

Xidao Wen

I. Introduction

This report contains related work of android application of CN3. The first section is a brief introduction of CN3, goals and techniques that being used through developing process. Then there is a section to talk about what I did to improve application performance and to fix errors during multiple tests. Due to long-term use of application framework, there still are limitations for the future work.

a) CONFERENCE NAVIGATOR 3 (CN3)

Conference Navigator website is a personal conference-scheduling tool. Based on features of CN3, users can review conference information like available interesting conference program and proceedings, and then make personal settings, such as mark a paper and add a session to attend. What's more, the author list and their information can also be reached in CN. One of the wonderful functions is that people who have signed in can add a session they like easily into their own schedule and manage them conveniently. Conference Navigator's main aim is to enhance users experience at conferences.

Since the Android application based on Conference Navigator is mainly designed for the specific users preparing to attend conferences. Considering about functions, it contains schedule, presented papers, workshops, keynotes and also sign in, which covers most functions in CN website.

b) Goals

Based on previous work, the application still has some limitations. The major one is update functions, in the last version, after installation it cannot update for any information in the conference. Because they store data in local of configuration files, users cannot get newest data from server unless they reinstall the application. Another one, for each conference, users need to install related applications, which will waste storage and time for users. Thus, the goals is to initially load and update all data from server database.

c) Technical details

Development Environment and Techniques:

1. Android 4.2.2

2. Android 4.2 APIs

3. SQLiteDatabase

4. Eclipse with JDK

Emulation environment:

For initial simulation: Android Virtual Device (AVD) by Android SDK

For real emulation: ASUS Google Tablet NEXUS 7

Source link:

conference information:

http://halley.exp.sis.pitt.edu/cn3/loadAllConferences.php

sessions and paper information:

http://halley.exp.sis.pitt.edu/cn3mobile/allSessionsAndPapers.jsp?conferenceID=[conferenceid]&noAbstract=1

paper's abstract:

http://halley.exp.sis.pitt.edu/cn3mobile/contentAbstract.jsp?contentID=[contentid]

II. Finished work

a) Work with a new conference:

Based on old version of Hypertext2014, we continued to use the same database data structure and schemas, so in this step what we need to do is to replace contents of the new conference, such as conference information, and sessions or workshops detail information. First of all, I configured the environment of Eclipse and Android application, including to import source code, to set up android libraries, and to set developer options on tablet to test the applications. Then, I was familiar with the original source code. In the data part, we used ConferenceDataLoad to load local data in the app, so in the new version, the quickest way was to replace local data in this class. After we got sql db files from database, we parsed xml to store them in local json file in order to load to the android application. Then with the new data, we set up a new conference application.

b) Features and interface:

Based on existed features, clients can review conference, session, paper and other information, and they also can add some workshops or other sessions into their own schedules and can make papers as starred. There are interfaces to show keynotes, workshops, schedule, and proceedings. In parts of keynotes, workshops and schedules, there are sub-interface to show details, but they point to the same pages as proceedings. The update features are designed with a button in two interfaces.

One is in the main page in order to update all data. The other one is in my schedule page after clients logging in, which is used to sync data based on users' behaviors. Thus, in the task, we need to fix the update functions to improve performance of the application.

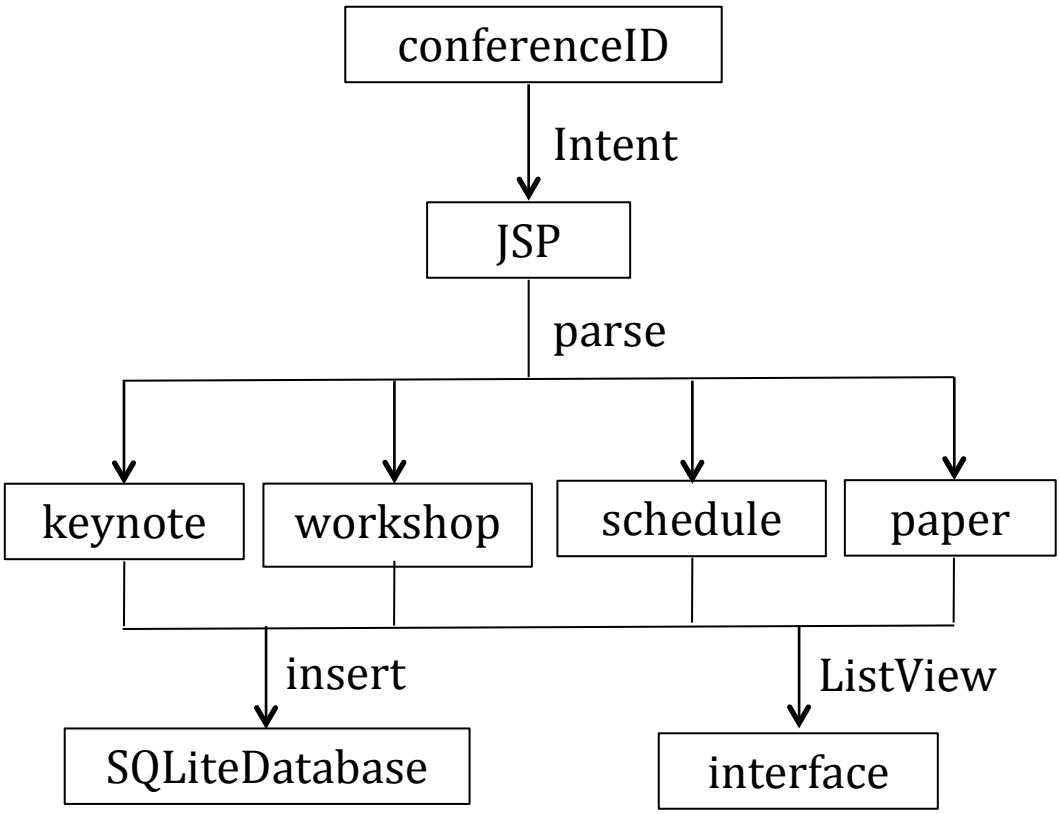


Fig1. Architecture of loading data

For initial data loading, there is a drop list to choose conferences which users want to load and then it will pass conferenceID as argument to server. We just need a framework without data in the application.

For later update function, we need to set keynote, workshop, session and paper parse to parse JSP pages and put values into interface and SQLiteDatabase.

c) Implementation:

Before the initial data loading, we need to keep framework of application, but to remove local data load class. Thus, the users firstly install application screenshots as below showed.

Fig2. Conference information

Fig3. Keynote

Fig4. Schedule

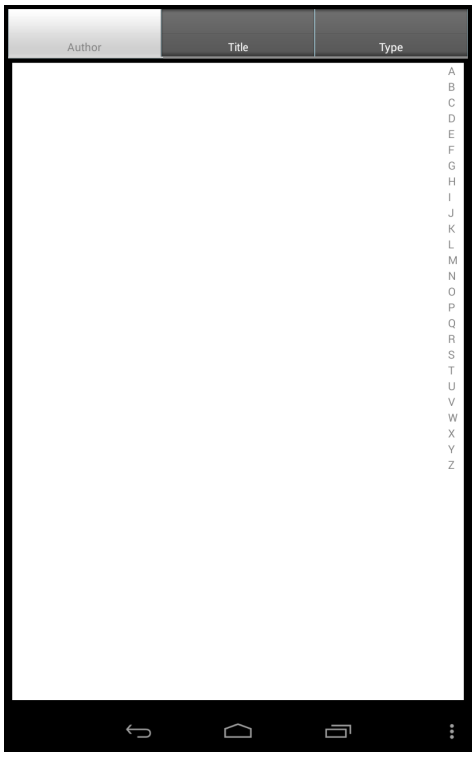
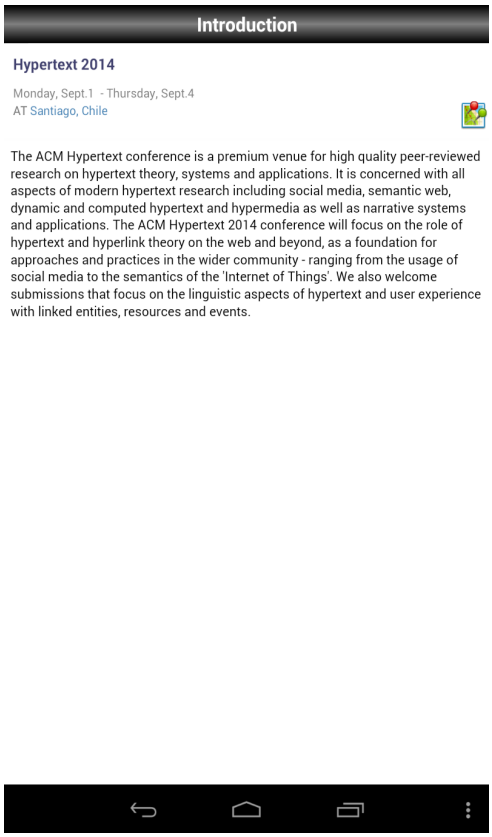


Fig5. Paper

For the first data loading main update function, we used the same update button to create activities and used conferenceURL to check update, and then set a status node to write in timestamps. Using timestamps is helpful to decide update needs and then to show results to clients. Then, we written class of LoadPaperFromDB and LoadSessionFromDB to get updated data from database. In this part, we used conference url to query database updated data and timestamps to check new data. We added new nodes, such as getTitle and getAuthor to implement update events. But there are issues in view events, it can retrieve paper information but in DBAdapter, there is no related objects, so the detail pages of workshop or keynote don't work as well.



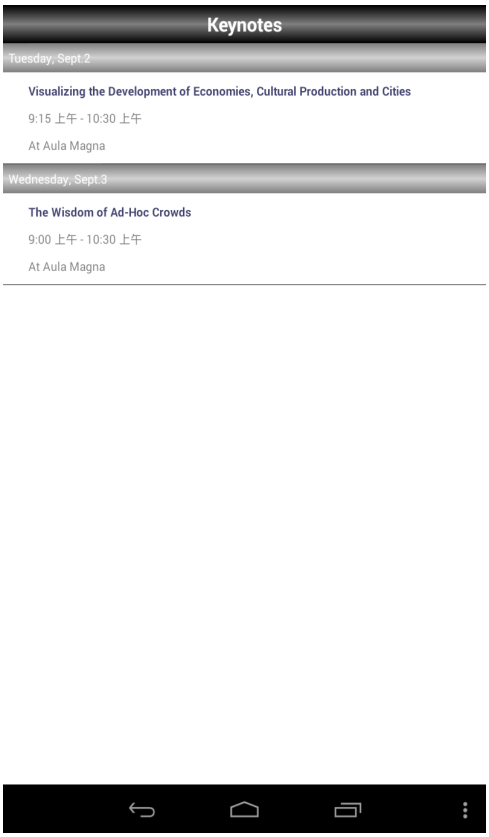


Fig6. Conference information

Fig7. Keynote

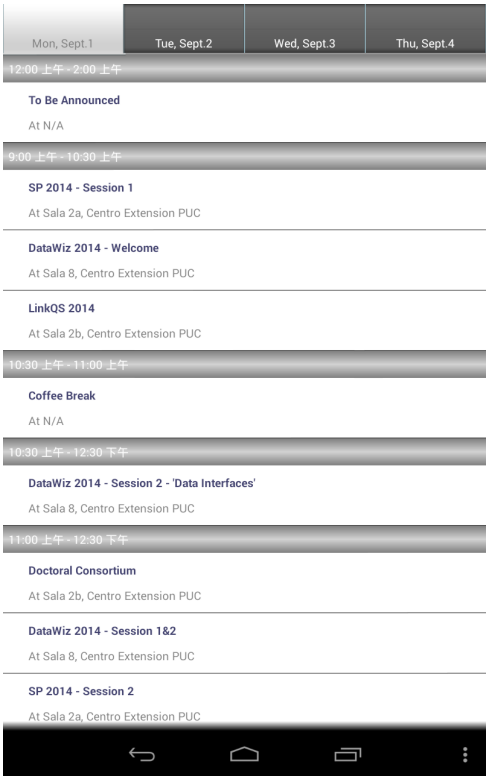


Fig8. Schedule

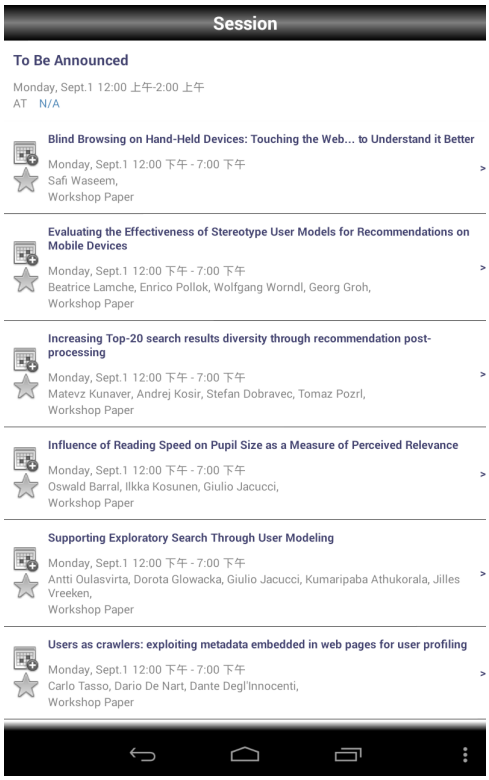


Fig9. Schedule detail

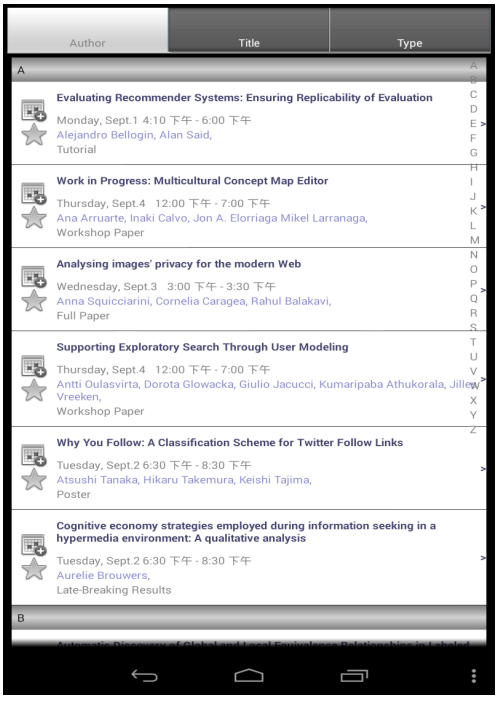


Fig8. Paper

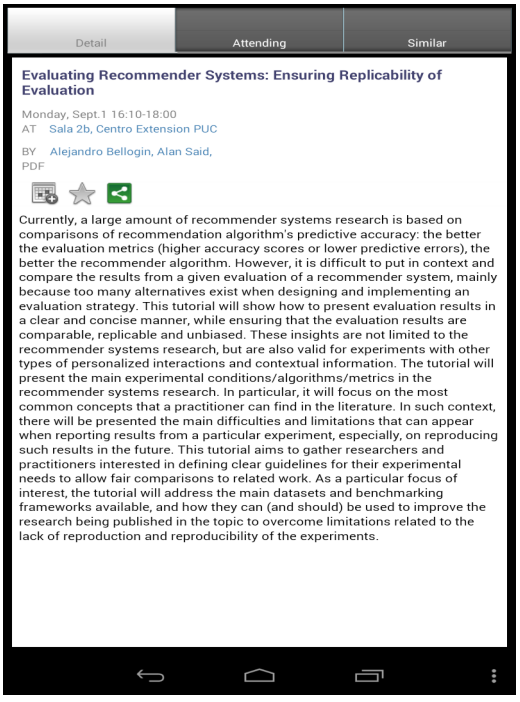


Fig9. Schedule detail

Thus, we used related parse classes to parse JSP pages of conference navigator to get keynote, workshop, sessions, paper abstracts and paper contents. Then we insert values of SQLiteDatabas and pass arguments into ListView and TextView to show them in the related interfaces. The update and loading results showed in the below figures.

At the same time, in UpdateOption class we re-writed method of update paper content information. For the logged in users update function, it has same parts as sync paper and schedule. But one more thing, we need to upload users' behaviors to server and then sync them in the application end.

If we want to provide an application to apply for every conference, we need to delete initiate loading data in the app, instead to provide a blank application. Therefore, I removed data load class and related method in main function, and added a method in DBAdapter about conference information. In the method, I set a hash table to store conference URL to query in the database. Until now, we can load conference basic information from update function in a blank application.

Parts of code screenshots are showed as below figures.

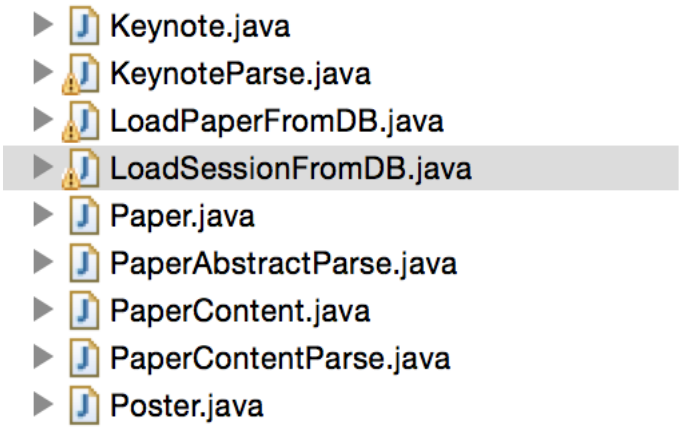


Fig10. Classes for parsing

```
private class KeynoteParseHandler extends DefaultHandler {
    private int state = 0;
    private Keynote ke;
    private boolean keynoteStart = false;

    public void startDocument() throws SAXException {
    }

    public void endDocument() throws SAXException {
    }

    public void startElement(String namespaceURI, String localName,
            String qName, Attributes atts) throws SAXException {
        if (localName.equals("Items")) {
            keynoteStart=true;
            return;
        }
        if (localName.equals("Item")) {
            ke = new Keynote();
            ke.ID= atts.getValue("eventSessionID").toString();
            return;
        }
        if (localName.equals("sessionName")&&localName.contains("Keynote")) {
            state = 1;
            return;
        }
        if (localName.equals("sessionDate")) {
```

Fig11. Parts of keynote parse

III. Conclusion and future work

From this project experience, I think I learned not only things about how to develop and test android applications, also how to build stable architecture and to improve coding skills. The application needs long-term management, so I also learned how to make readable and easy extended data structure and codes.

In the future work, we want to load different conferences in the same applications, so we need to create a new page for users to choose which conference they want to load, and then initially load related conference information. For this work, we need to create new node to store and pass conference ID argument to all parse classes with JSP URL.