A Report on

Rural Net Connect

(An android app)

January 2015

Revision History

Version	Date	Description
v1.0	27/01/15	Initial version.

Table of Contents

Introduction

‘Rural Net Connect’ is an Android app through which people with sparse internet connection can download their desired content from net easily. This app is specially aimed for rural people who do not have good net connectivity. The app allows user to mark her preferences of type of content she wants to download and the app downloads the relevant content whenever net is available. This app is expected to be the first of its kind in India.

Need and Purpose

In the present Indian scenario, many rural areas have limited (part of the day) or no data access. So, ‘Rural Net Connect’ will download the relevant and popular content available on the android devices offline by pushing the data when the device comes online.

Features

- The app allows users to download the content according to their preferences.
- The users can mark their preferences of type of content they want to download from currently four different categories by selecting appropriate ‘tags’ under each category.
- The app then downloads the content according to the marked preferences whenever net is connected.
- The app allows to extensively search through the available ‘tags’ under each category.
- The users can be sent push notifications through server.
- The users can edit their preferences later in time whenever they like.
- The app allows to extensively search through the available ‘tags’ under each category.
- Any type of content (image, text, video etc.) can be downloaded through the app.
- The refresh button on action bar allows the user to refresh the available tags whenever the device is connected to net.

References

[1] https://developer.android.com/guide/index.html
[2] https://parse.com/docs/android_guide
[3] https://stackoverflow.com

Design Overview

The app has been built with the minimum SDK version 10 and works for all the later versions of Android SDK. The app is also compatible with older Android OS versions particularly till Gingerbread. The backend of the app is based on “Parse” which provide complete backend service to store and execute server side scripts and to store data.

To provide rich search experience Knuth-Morris-Pratt algorithm is used to extensively search the database.

We have 4 main categories and content is stored in these categories with certain tags relevant to the content uploaded. If the user searches for any tag in the search bar, we will provide the user with all the related content.

Components

Parse.com: Parse.com, viz. a complete mobile app platform, takes care of everything that the app needs, from the core of the app to analytics and push notifications. It acts as the third-party server for storing our videos and provides a platform to download and upload files to the cloud.

Java: The Back-end and Front-end work has been done using Java OOP. Java provides several classes which make the code more efficient and easy to understand.

Android Studio (IDE): Android Studio is an IDE for developing on the Android platform and it has a very good build system, has built variants and multiple-apk generations. The same has been used as an IDE to develop the app.

The various categories of users access *the app* through an android device and the app has been made very user-friendly.

Interfaces

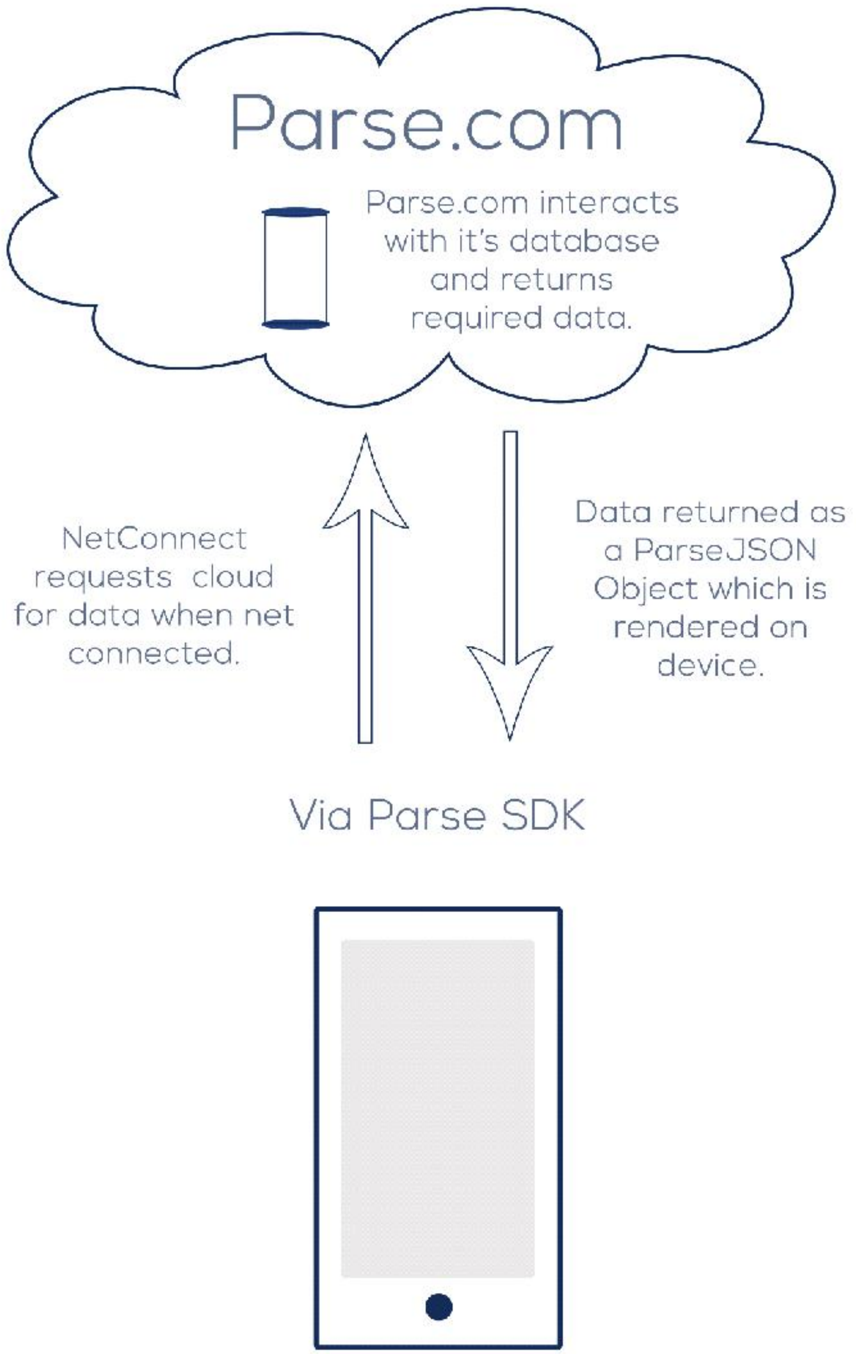
The interface between the front end Java code and the Parse server is through Parse SDK. The SDK provides functions to communicate with the server – to retrieve and store content from server, to extensively query the database, to run server side scripts and the like. Parse SDK also enables to have a localDataStore which again is interfaced through the inbuilt functions.

Architecture

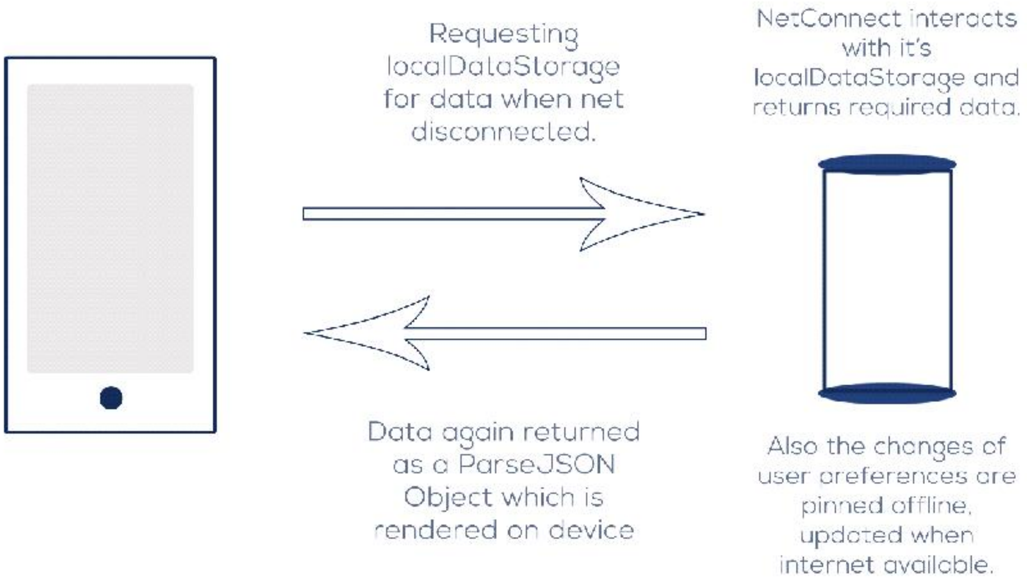
The MVC (Model-View-Controller) design, which is a framework for building applications is used to build the app. MVC allows to make the front-end and the back-end part of an app independent from each other.

The architecture of the program is further illustrated through these diagrams below:

When INTERNET Connected


Parse.com
Parse.com interacts with it's database and returns required data.
NetConnect requests cloud for data when net connected.
Data returned as a ParseJSON Object which is rendered on device.
Via Parse SDK

When INTERNET Disconnected


Requesting localDataStorage for data when net disconnected.
NetConnect interacts with it's localDataStorage and returns required data.
Data again returned as a ParseJSON Object which is rendered on device
Also the changes of user preferences are pinned offline, updated when internet available.

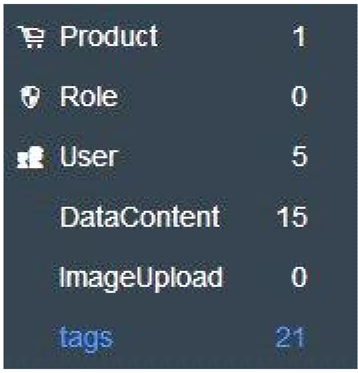
Algorithms and Data Structure

Search has been implemented through KMP (Knuth-Morris-Pratt) algorithm.
The Data Structures are based on JSON/JavaScript format. The relational tables are based on NoSQL, in that Java based queries are used instead SQL ones.

External Data

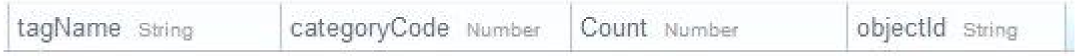
Databases

The whole database is stored on parse.com servers which acts as third-party platform for cloud data storage. The database gets updated as soon as any new user opens the app for the first time, or any user updates her preferences.

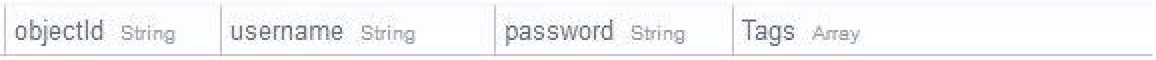


Product	1
Role	0
User	5
DataContent	15
ImageUpload	0
tags	21

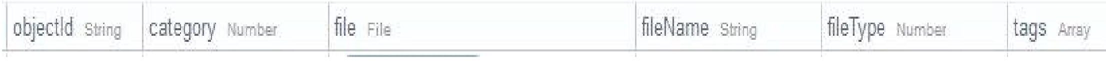
The tables in the database stored on Parse.com which are as follows:



tagName String	categoryCode Number	Count Number	objectId String



objectId String	username String	password String	Tags Array



objectId String	category Number	file File	fileName String	fileType Number	tags Array

Performance

The external server used (Parse) provides very high performance result to user queries. So, the time in retrieving or sending data essentially depends on user’s net speed. The performance is expected to remain almost constant with increasing number of users.

Test Users

We asked some of our college friends to sign up for our app and test its functionality.

In the initial state of our app, some of the users weren't able to navigate through the app with ease. This was because the front-end part of the app was not up to the mark.

To step it up a notch, we added some more functionality and the final version of our app is now easily understandable by all the users and has quite a good UI.

The database tables are successfully updated as a new user installs the app.

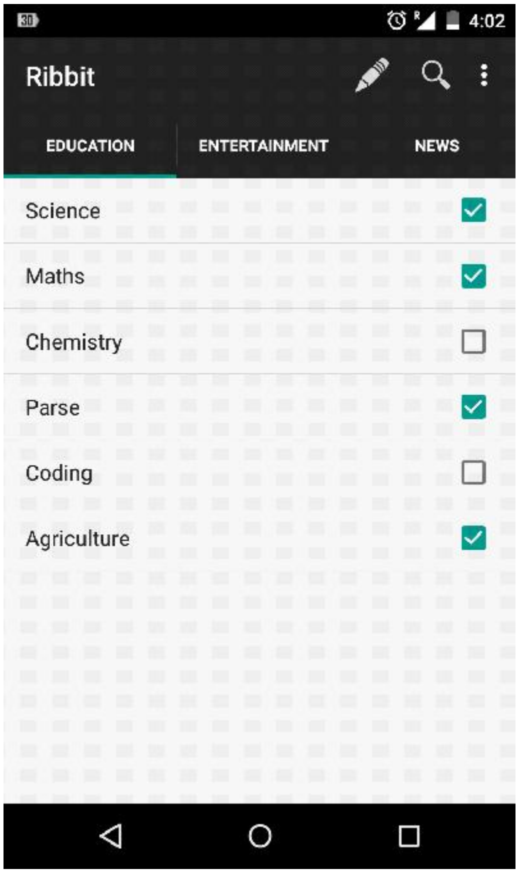
Future Aspects

We intend to develop this app much further so that it is much more useful by:

- Enabling the users to share relevant content with each other through Bluetooth.
- Extending the features keeping in mind the Agriculture needs of rural people.
- Providing more features like parsing the weather content in the app itself to present it in better format.
- Providing customised suggestions to user based on her past activity on app.

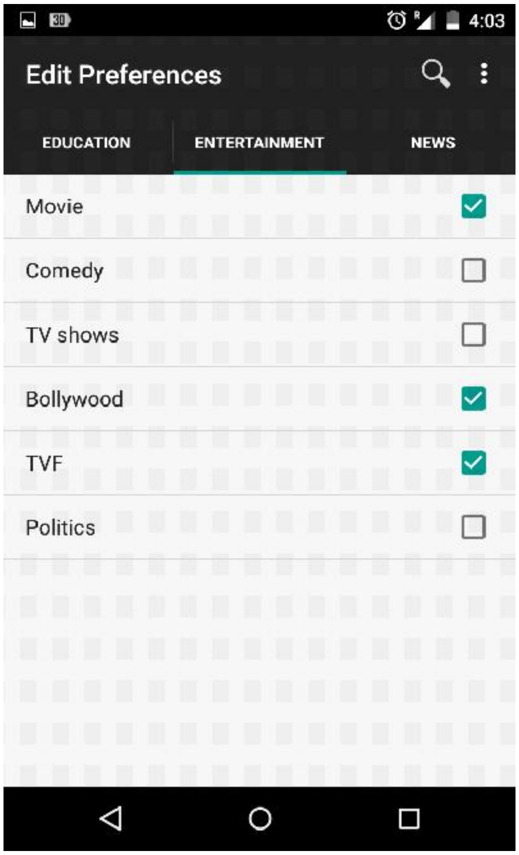
Snapshots

This is the screen which displays a category named 'Education' which has several tags associated with it.



Science ☑
Maths ☑
Chemistry □
Parse ☑
Coding □
Agriculture ☑

This is the sceen which describes the feature given to the user to edit his/her preference according to his/her requirements.



Movie ☑

Comedy ☐

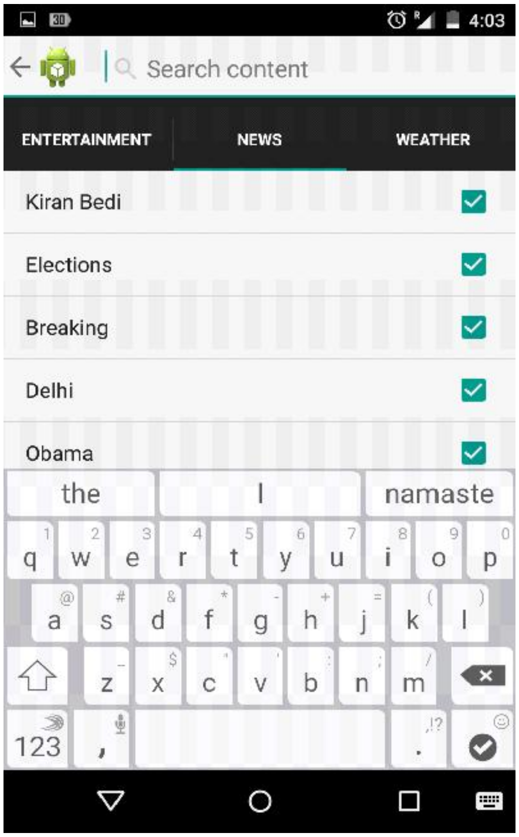
TV shows ☐

Bollywood ☑

TVF ☑

Politics ☐

The following two snapshots show the rich search experience given to the user. Substring search has been implemented.



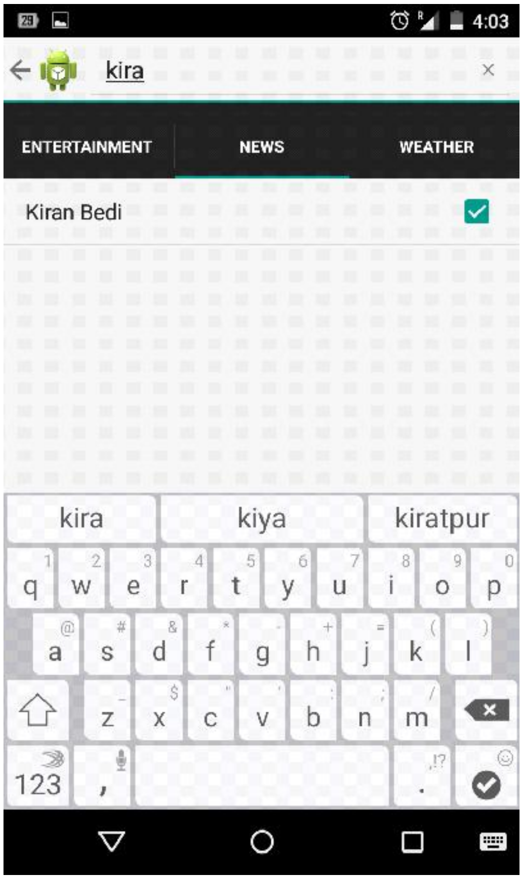
Kiran Bedi ☑

Elections ☑

Breaking ☑

Delhi ☑

Obama ☑



kira

Kiran Bedi ☑