GitHub Username: kewal07

ioCollect

Description

Data Collection is one of the most important phase of any feedback related study, be it a field survey or an event feedback. ioCollect is an Android App that makes the process of data collection seamless, effective and fast. In India, internet penetration is not good. There are many areas where data collection is done via pen paper mechanism, then the same is entered in an excel which is then uploaded somewhere where the final analysis takes place. ioCollec aims to simplify this process. The app It allows the user/ field agent to record data offline which automatically gets synced to an online repository whenever there is internet connection hence reducing any manual errors or intervention in the process and saving thousands of hours as well. It also analyses the data and shows the results in a graphical and visually appealing manner with filters to analyse the result by various demographics such as age, gender, location etc.

Intended User

The intended users are B2B organizations who employ field agents to collect data. The app can also be used by different Government organizations and NGOs who perform a lot of research studies and surveys.

Features

The main features of the app are:

- Saves information offline when no internet connection detected and syncs it back to the server whenever internet is back.
- Takes pictures and records audio in response to any media based question. The recorded media can be used as an evidence.
- Visual representation of the data on the app itself in terms of graphs and charts.
- Data Analysis (akin Tableau) with ability to do cross tab filtering and location based filtering as well.
- There will be a backend server which will host/store all the data. The forms and surveys will be a result of the api calls from the server.
- An App widget would notify the user about the number of unsynced responses that are locally stored on the device.

User Interface Mocks

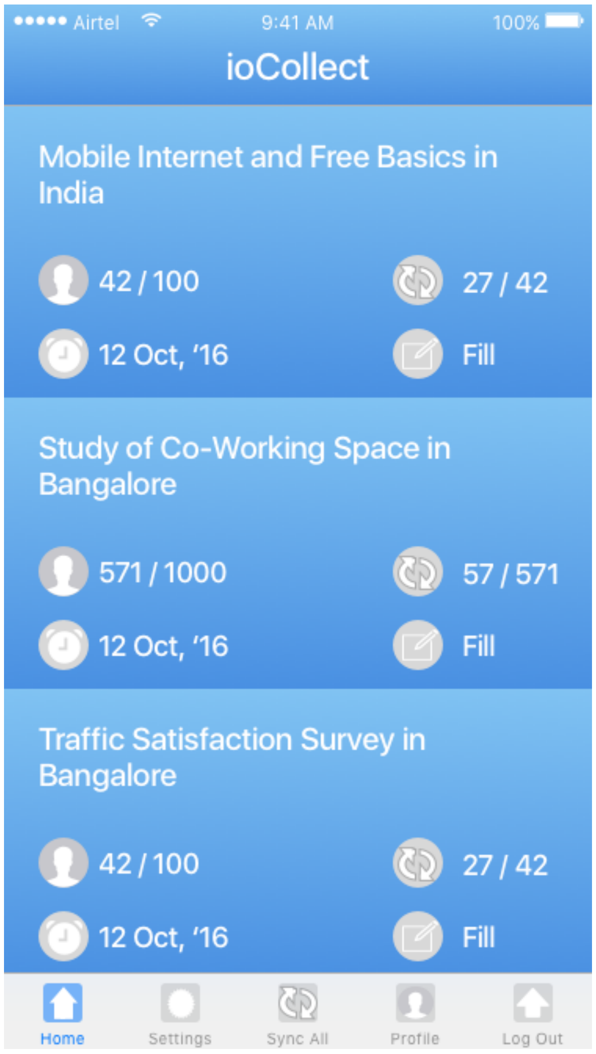
PS. These are just sample ui mocks and actual implementation would adhere to material design concepts.

Screen 1

- Description: The images shows the home screen of the app. Here a list of all the surveys are displayed. All the surveys are displayed using ListView.

  Each block contains the following information:
  1. Name of the survey (study)
  2. How many people have filled it (ex. 42/ 100 means 42 people of the required 100 have filled the survey)
  3. How many responses have been synced to server
  4. The last date by which the survey should be filed
  5. A button from where the actual survey can be opened and filled

There would also be a menu to log out, change settings, sync all the survey responses at once if internet connection is there and the home button which would take back to this screen.



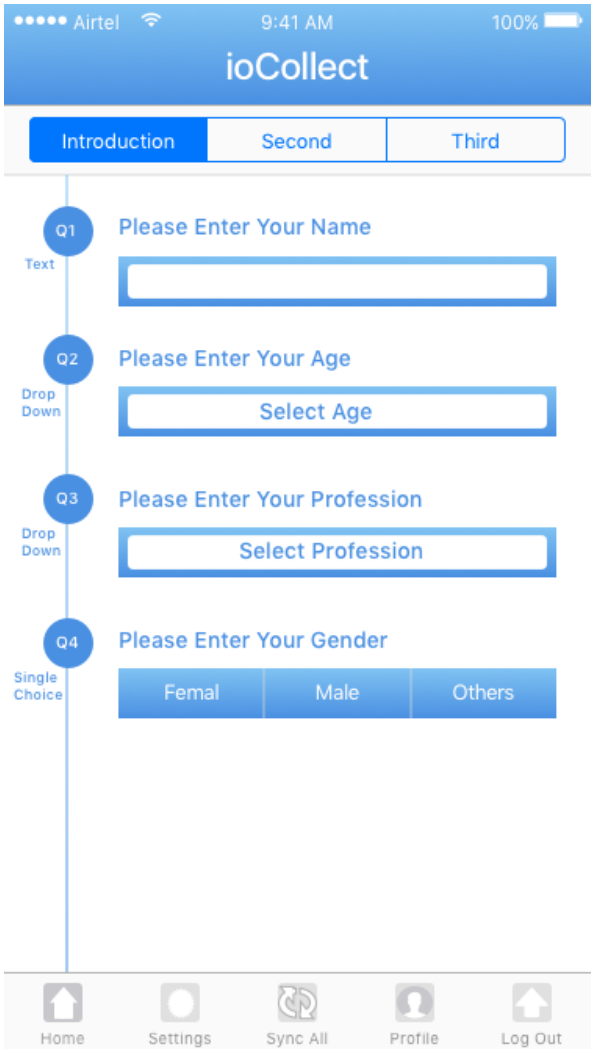
Mobile Internet and Free Basics in India
42 / 100 27 / 42
12 Oct, '16 Fill

Study of Co-Working Space in Bangalore
571 / 1000 57 / 571
12 Oct, '16 Fill

Traffic Satisfaction Survey in Bangalore
42 / 100 27 / 42
12 Oct, '16 Fill

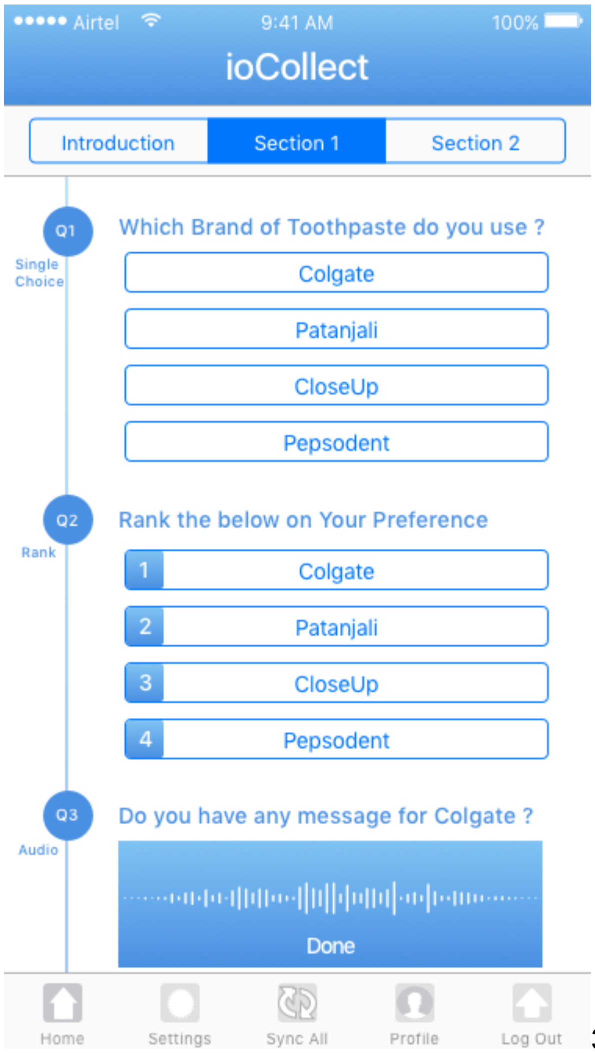
Screen 2

- Description: The images below shows the screen of the app where a survey has been clicked or opened from the home screen. Here a list of all the questions and sections within the survey are displayed. The questions can be of various types such as
    1.
    2. Radio Button Questions (Single Select)
    3. Checkbox Questions (Multiple Select)
    4. Text Entry Questions
    5. Media Entry Questions (Audio , Video)
    6. Rating Question

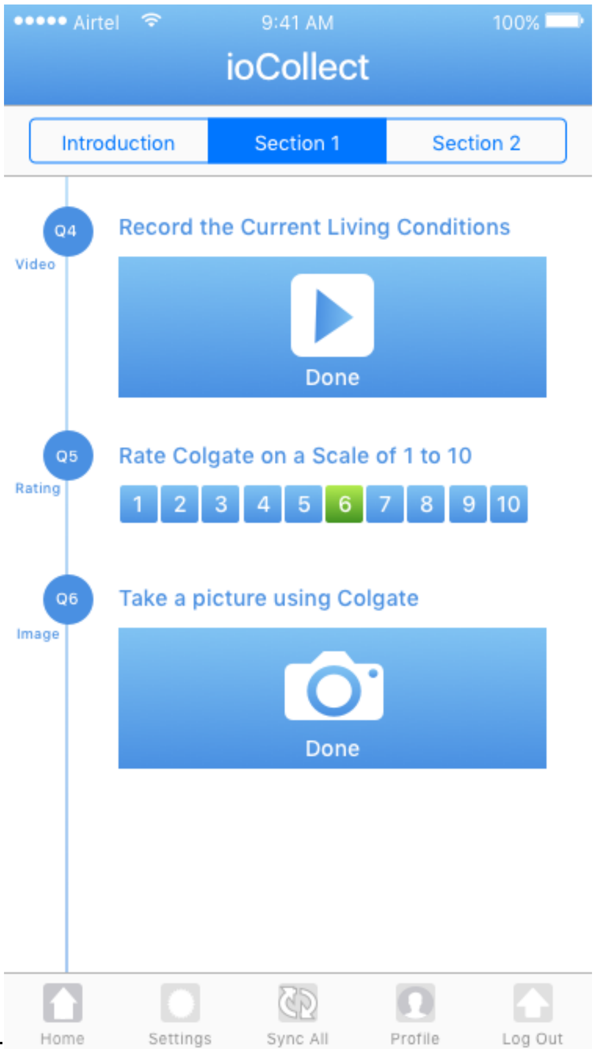


Screen 3 & 4

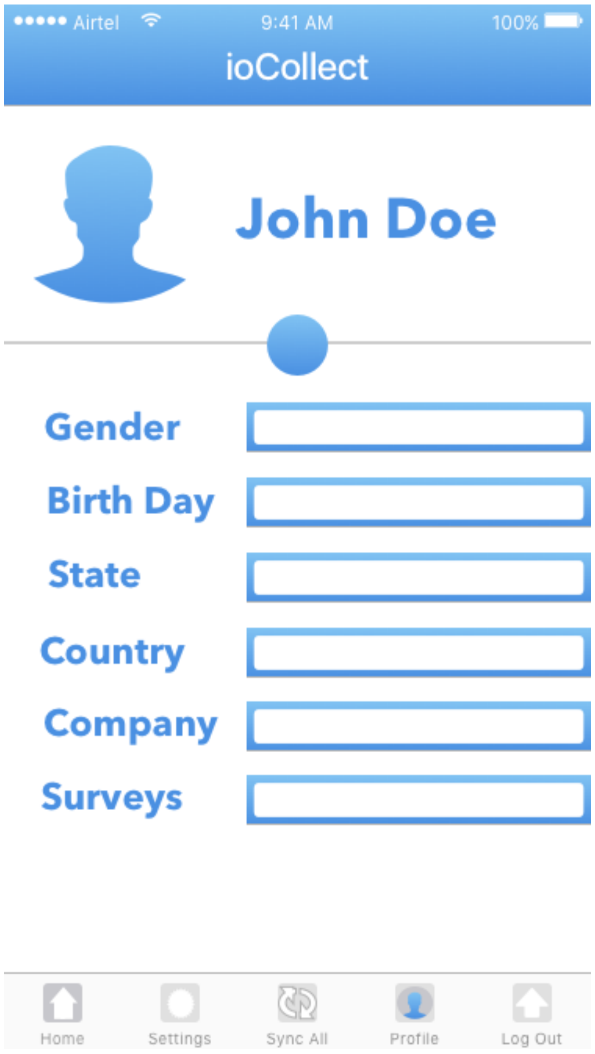
- Description: The images below shows the different types of questions that can be in a survey.


●●●●● Airtel
9:41 AM
100%
ioCollect
Introduction
Section 1
Section 2
Q1
Single Choice
Which Brand of Toothpaste do you use ?
Colgate
Patanjali
CloseUp
Pepsodent
Q2
Rank
Rank the below on Your Preference
1 Colgate
2 Patanjali
3 CloseUp
4 Pepsodent
Q3
Audio
Do you have any message for Colgate ?
Done
Home
Settings
Sync All
Profile
Log Out

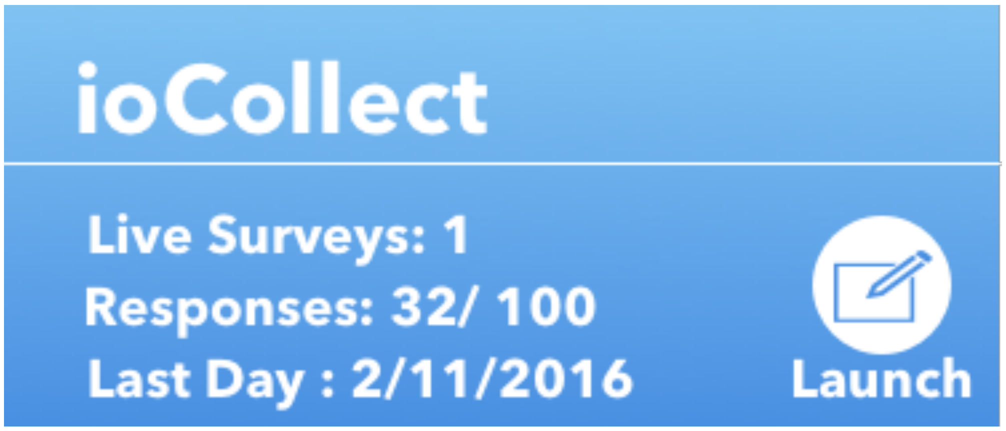
3&4


●●●●● Airtel
9:41 AM
100%
ioCollect
Introduction
Section 1
Section 2
Q4
Video
Record the Current Living Conditions
Done
Q5
Rating
Rate Colgate on a Scale of 1 to 10
1 2 3 4 5 6 7 8 9 10
Q6
Image
Take a picture using Colgate
Done
Home
Settings
Sync All
Profile
Log Out

Screen 5 : Profile



Widget



Other important screen would be the analysis/dashboard screen. **However, will prepare the UI and UX for that after the basic app is up and running as that would be a good add on rather than core feature.**

Key Considerations

How will your app handle data persistence?

Use of content providers to handle locally save data and upload data to a server when sync is on using REST APIs.
Will be using API's provided by my own site which is a survey platform known as AskByPoll. Will be using REST APIs to handle data upload and all the data would be saved in a MySql DB hosted on an AWS server.

Describe any corner cases in the UX.

If a user clicks the back button or home button while filling the survey then comes back then it can be a corner case. Also, movement between different sections can be tricky to handle as per user expectation.

Describe any libraries you'll be using and share your reasoning for including them.

Picasso: to handle images; Reason: familiarity during the earlier projects ☺
Will be using API's provided by my own site which is a survey platform known as AskByPoll. Will be using REST APIs to handle data upload and all the data would be saved in a MySql DB hosted on an AWS server. The survey creation takes place online on this platform only.

Describe how you will implement Google Play Services.

1.Location & Context: Will map responses to location and limit number of responses from a certain location to prohibit fraud.
2.AdMob: For free users

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

The subtasks.:
- Implement login
- Implement the home page showing the list of surveys to be filled along with some minor details
- Implement the survey detail page
- Implement the Vote functionality and store the response locally
- Implement the sync functionality to send the data to the server

Task 2: Implement UI for Each Activity and Fragment

Subtasks:
- Build UI for Login Activity
- Build UI for Main Activity (List of surveys)
- Build UI for Detail Activity

Task 3: Implement the Login
- Build the UI
- Implement login with the server/facebook
- Implement the fetching of the list of surveys assigned
- Saving the data locally with the use of content providers

Task 4: Implement the Home Page

- Build the UI
- Show the grid with the list of surveys fetched after login

Task 5: Implement the Details Page

- Build the UI
- Implement the Survey details page and save the details locally using content providers
- Implement the vote functionality and save the data locally

Task 6: Implement the Sync Functionality

- Implement the sync of the local data with the server

Task 7: Implement the Widget

- An app widget would show the number of responses that are stored locally and have not been synced to the server yet.

PS: Will use Loaders to show data from local device. Will use Async Task to fetch data from server. Will evaluate between different options as I develop the app. For example, I will also try to use SyncAapters to schedule data sync whenever there is internet connection and at a predefined daily time.

PS2. Thanks for suggesting different libraries as well. Will definitely have a look and try to use something like RxJava, Retrofit or Dagger.