Zapp - Report

12 June 2014

Matricola studente: 146077 Leonardo Lanzinger

Matricola studente: 154849 Giulio Michelon

Matricola studente: 151665 Matteo Lever

Project video:

http://youtu.be/iGdvGbCEKGo

Testing

Per usare questa app potete usare un account di testing:
username: mailclientandroid@gmail.com
password: ciaozapp

Miniguida per l'utilizzo di Voice Control

Nell'interfaccia di lettura della email è possibile controllare l'app tramite Voice Control premendo il pulsante in basso al centro con l'icona del microfono. Zapp riconosce i seguenti comandi vocali:

- ***Rispondi*** + corpo del messaggio
- ***Rispondi a tutti*** + corpo del messaggio
- ***Ricordami / Ricordamelo oggi***
- ***Ricordami / Ricordamelo domani***
- ***Ricordami / Ricordamelo***

Abstract

Every day we receive a lot of emails. We rely on them to organize our tasks, to schedule our work and to update people in our projects. We want a better organization through emails. The usual inbox is full of noise and old emails. We want to organize the flow putting the important content into a todo list that helps you meet deadlines with scheduled notifications. Zapp will display by default only marked or unread emails to keep the main tab clear and organized while giving easy and fast control over the inbox through swipe gestures. The app will also feature a voice control that will provide even more speed and comfort while replying and organizing messages. Our solution is completely different from standard Android email clients: we turn your email client into a task-oriented reminder list that it's fast and easy to use.

Related work

All mail clients in the market are somehow similar to our app since they all try to solve the same problem: emails management.
Some app that uses emails as tasks are ***Mailbox***, ***Mailpilot*** *and* **Evomail**. *They all have very nice apps for iOS, Mac OSX and Android. Our competitors have a similar approach but they dont't follow the todo list method as much as our app. In fact* **Zapp*** focuses its main tab in the to do list.

Usage Model

Once the user has set his google account through the login he can start using the app.

On the left side the user can find the Drawer: the slide-in menu of the app. The sections are the Todo list, the Inbox, Sent mails and Trash Bin. The user can access the Settings touching on his name or through the physical settings button.

The user can organize his emails through gestures. He can swipe to the left in the **Inbox** to set the mail as a task that he has to do and he can choose the expiration of the task. Alternatively he can swipe to the right to delete the email.

Once the user completes his task he can set his tasks as done swiping to the right on the **Todo** section. In the **Todo** section all the tasks are listed in expiration order and there are labels displayed to help the user. ***Unread emails*** are also displayed in this todo list tab.

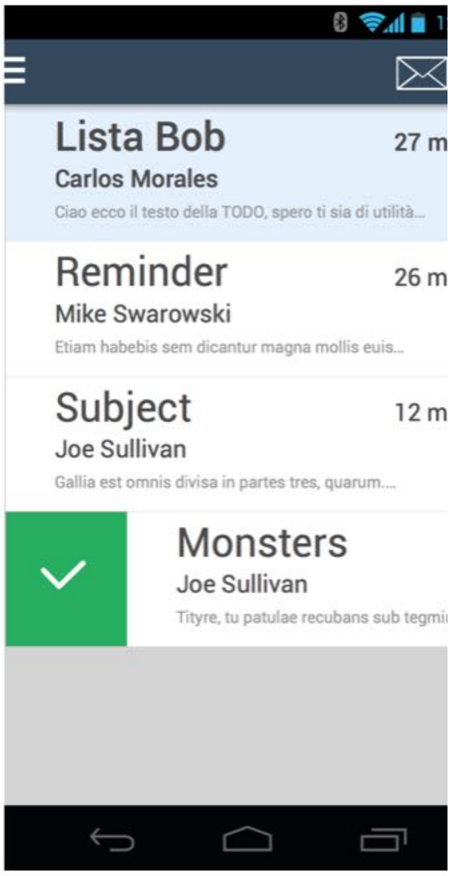
The **Sent** section is the place where he can find the mail that he has send and he can find deleted mails in the **Trash Bin** section.

In the **Settings** the user can change his email and password and he can decide his work time. The notification for his task will be displayed accordingly to his personal time table. For example if he set a task for tomorrow and his working time starts at 8.30 he will receive a notification at 8.30 that will help him remember the task. If a task is not completed before the work end time another notification is triggered accordingly to the set end time.

Wireframes

To do list view

The default and main interface of the app is the todo list. The user can interact with his email as tasks.
In this interface you can see only the emails marked as “todo” or unread mails.
By swiping to the right the user can set as complete a task.



Lista Bob 27 m
Carlos Morales
Ciao ecco il testo della TODO, spero ti sia di utilità...

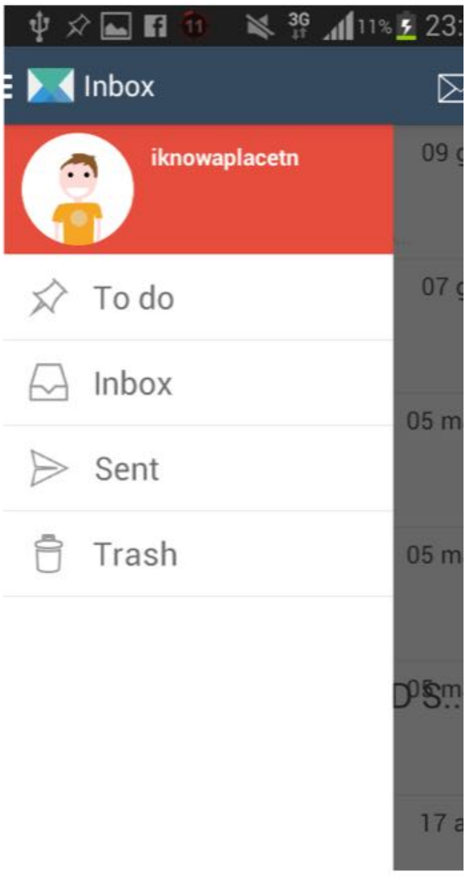
Reminder 26 m
Mike Swarowski
Etiam habebis sem dicantur magna mollis euis...

Subject 12 m
Joe Sullivan
Gallia est omnis divisa in partes tres, quarum....

Monsters
Joe Sullivan
Tityre, tu patulae recubans sub tegmi

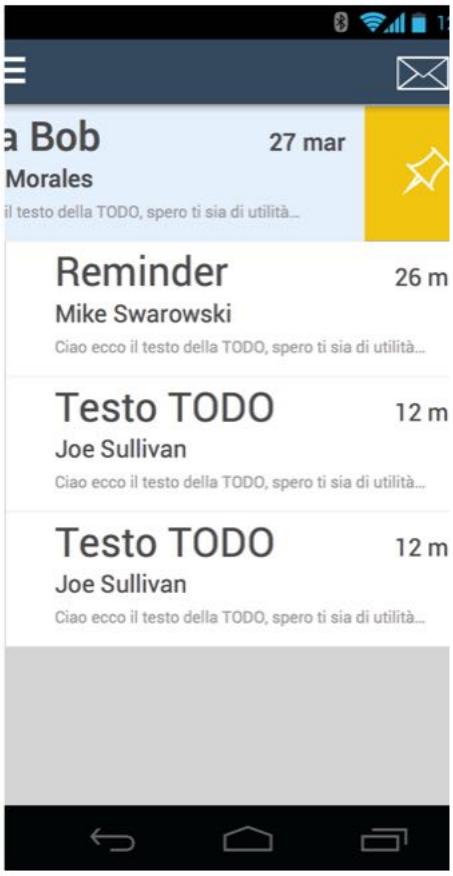
Navigation drawer

You can still find your inbox and all your emails in a convenient drawer menu.



Pin emails

Pin the emails in your inbox swiping to the left. They will appear in you task list.


a Bob 27 mar
Morales
il testo della TODO, spero ti sia di utilità...

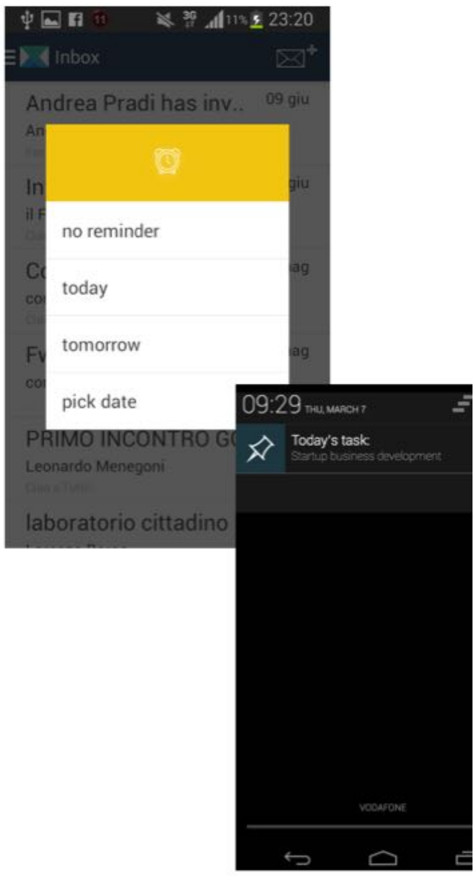
Reminder 26 m
Mike Swarowski
Ciao ecco il testo della TODO, spero ti sia di utilità...

Testo TODO 12 m
Joe Sullivan
Ciao ecco il testo della TODO, spero ti sia di utilità...

Testo TODO 12 m
Joe Sullivan
Ciao ecco il testo della TODO, spero ti sia di utilità...

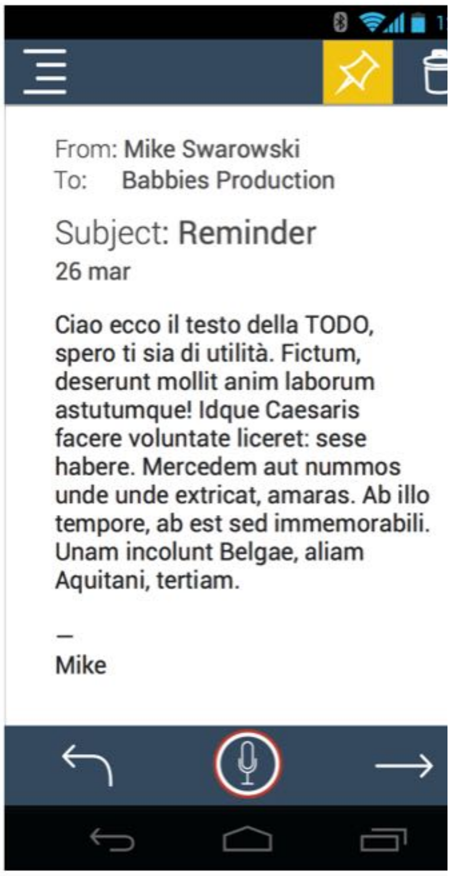
Schedule your tasks

Set an expiration date for each pinned email. The app will automatically remind you deadlines through notification.



Read and pin

Read the new mail and decide if you need to pin them as a task. Tap on the top pin icon to set them as pinned.
Otherwise you can delete it by clicking on the trash button.



From: Mike Swarowski
To: Babbies Production

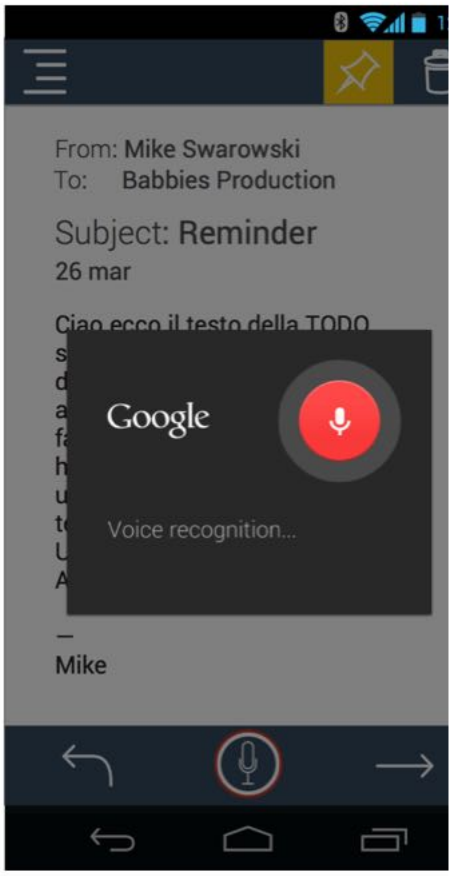
Subject: Reminder
26 mar

Ciao ecco il testo della TODO, spero ti sia di utilità. Fictum, deserunt mollit anim laborum astutumque! Idque Caesaris facere voluntate liceret: sese habere. Mercedem aut nummos unde unde extricat, amaras. Ab illo tempore, ab est sed immemorabili. Unam incolunt Belgae, aliam Aquitani, tertiam.

–
Mike

Talk to me

Use your voice to reply or pin the email. The app will understand your message and write down the answer. The app will also understand if you want to “pin” the email and will set it’s deadline according to your voice command.


From: Mike Swarowski
To: Babbies Production
Subject: Reminder
26 mar

Ciao ecco il testo della TODO
s
d
a
f
h
u
t
U
A

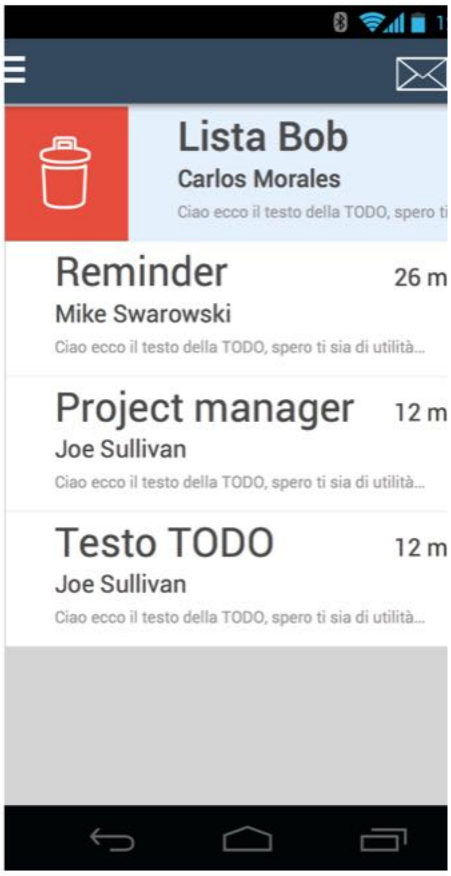
—
Mike

Google

Voice recognition...

Organize through gestures

Swipe right and delete emails.
Light speed organization.



Lista Bob
Carlos Morales
Ciao ecco il testo della TODO, spero ti

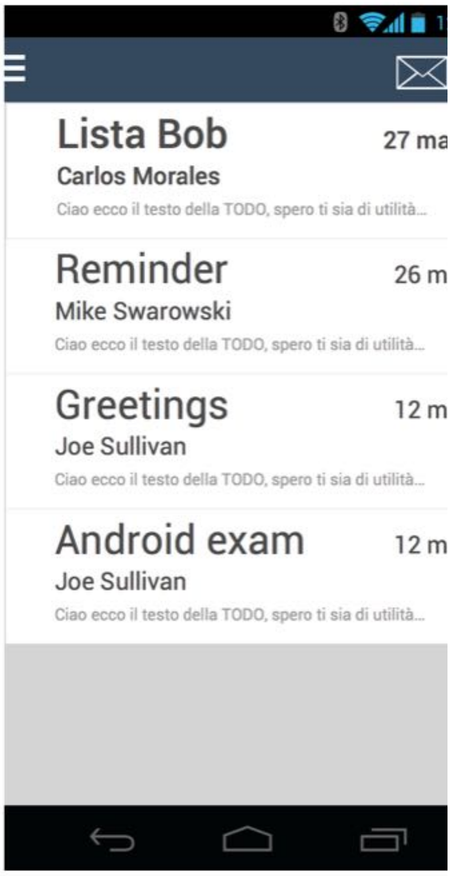
Reminder 26 m
Mike Swarowski
Ciao ecco il testo della TODO, spero ti sia di utilità...

Project manager 12 m
Joe Sullivan
Ciao ecco il testo della TODO, spero ti sia di utilità...

Testo TODO 12 m
Joe Sullivan
Ciao ecco il testo della TODO, spero ti sia di utilità...

Expiration mark

Mails which deadline is set for today are marked with a yellow border.



Lista Bob 27 ma
Carlos Morales
Ciao ecco il testo della TODO, spero ti sia di utilità...

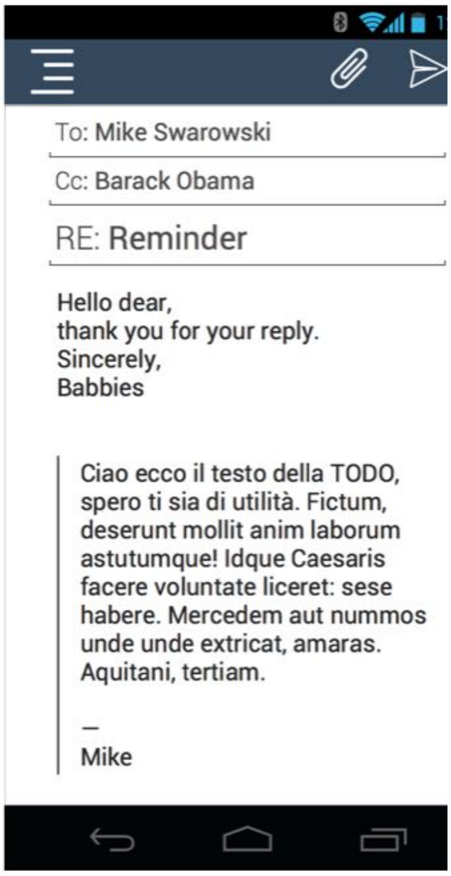
Reminder 26 m
Mike Swarowski
Ciao ecco il testo della TODO, spero ti sia di utilità...

Greetings 12 m
Joe Sullivan
Ciao ecco il testo della TODO, spero ti sia di utilità...

Android exam 12 m
Joe Sullivan
Ciao ecco il testo della TODO, spero ti sia di utilità...

Reply

Reply to your emails.


To: Mike Swarowski

Cc: Barack Obama

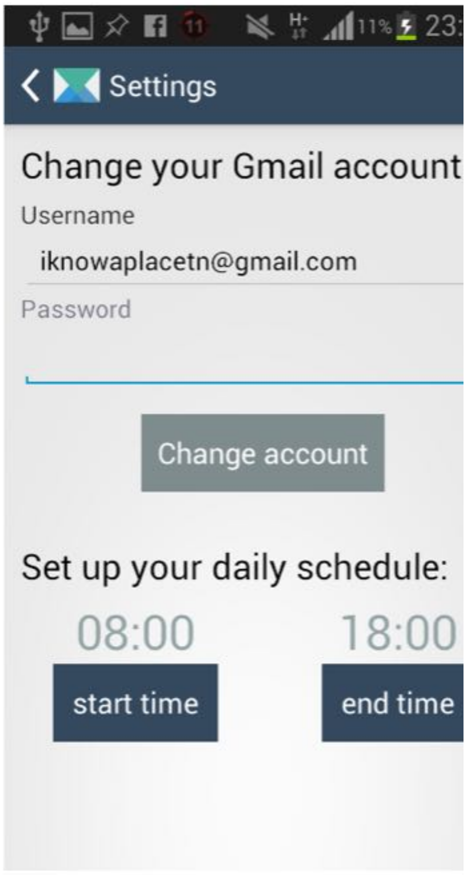
RE: Reminder

Hello dear,
thank you for your reply.
Sincerely,
Babbies

> Ciao ecco il testo della TODO, spero ti sia di utilità. Fictum, deserunt mollit anim laborum astutumque! Idque Caesaris facere voluntate liceret: sese habere. Mercedem aut nummos unde unde extricat, amaras. Aquitani, tertiam.
>
> —
> Mike

Settings

Change your address and your daily schedule.



Architecture Design

The app is based on the IMAP protocol and works only with Gmail accounts. We chose to go with google accounts because JavaMail, the library which supports mail transfer protocols in Java, has a different implementation for each mail server. So to support different mail providers the app would have implemented several version of its core classes. Gmail has been chosen over different email providers since it is the most used email server in the Android environment.

Zapp implements a mail client, as well as a todo list. Both components work toghether, as they share the same Collection of Email objects inside the app. Zapp uses several Android built in services, such as ***Notifications***, ***VoiceRecognition*** and ***AccountManager***.

Data is stored both in Android SharedPreferences (account data) and in LocalStorage (email list).

Requirements / assumptions

As well as a valid Google account (which can be created through Zapp) the application needs data connection to contact the remote email server.

Tools and software used

The team used Android Studio as a IDE to program Zapp, as well as Sketch 3 to design mockups and graphics and Final Cut to edit the final video.

Implementation

JavaMail controls all the IMAP logic that works in the background, and is implemented mainly in four Java classes that downloads Inbox emails, send mail and keep Zapp and Gmail server up to date with each other.

Accounts are added from Android *AccountManager* and managed inside the app. Zapp retrieves from Google the account name and surname as well as the user profile picture.

Notifications are managed through Android *NotificationManager* and BroadcastReceiver.

Voice control is implemented using Android *RecognizerIntent* and then passing the result to ***SpeechAnalizer.java*** that analizes the resulting voice command and performs actions.

The ***main app layout*** consists in a *FragmentContainer* that displays one at a time in the main tab four different fragments (***TodoFragment.java,***

InboxFragment.java, SentFragment.java and ***TrashBinFragment.java***). MainActivity also implements *DrawerLayout* for the lateral slide-in menu. Each content fragments displays a ***PullableListView*** (an external library) that contains the email list and when pulled down triggers an animation and the *ReceiveEmail* method.

Custom swipe library

Each *Email* item in the *PullableListView* implements swipe gestures to perform different actions. Swipe gestures have been implemented from scratch by the team as Android doesn't provide a native implementation of swipe gestures. Those custom Java classes are ***Animator.java*** and ***SwipeDetector.java***. This is a relevant part of the programming process as it is an emerging desing trend though still not supported by native libraries.

External libraries

Zapp uses (as well as ***Pulltorefresh Listview*** and ***JavaMail***) also ***Castorflex.Smoothprogressbar*** library to provide an colorful progressbar while receiving emails and ***Doomonafireball.Betterpickers*** to implement new google-like Date and Time Picker that are used in new Google Calendar app but not included in the standard Android JDK.

Evaluation

Zapp has been tested on real smartphones using real Gmail accounts. We asked several users to try and use our app and we got positive feedback mostly because of the quickness of the interaction with the system. ***Voice Recognition*** and ***Swipe Gestures*** are the most enjoyed features as they reduce significantly the interaction effort of the user.

In order to keep data load to the minmum Zapp doesn't automatically download attachments from emails and only donwloads them after the users tells the app to do so.

Limitations

Although the app has been tested several times using real Gmail accounts, sometimes it crashes when trying to perform network operations while losing network service.

Member contributions

Giulio Michelon 60 commits / 3664 ++ / 4,820 --

- Android app development (frontend)
- Mockup design
- Graphic design
- Swipe gesture implementation

Leonardo Lanzinger 125 commits / 14,224 ++ / 6,363 --

- Android app development (backend and frontend)
- Set up of Github repository
- JavaMail logic and implementation
- External libraries import
- Swipe gesture implementation
- Mockup design
- Logo design

Matteo Lever 46 commits / 6674 ++ / 3,619 --

- Android app development (backend and frontend)
- JavaMail logic and implementation
- Account management implementation
- Mockup design

Lesson learned

We should have planned better the implementation of the JavaMail part of the app, as it happened a few times that we had to refractor the logic in order to work with additional features. This would have been possible only if we had known before the features of JavaMail.

References

Github repository

https://github.com/leolanzinger/mailclient

Project video

http://youtu.be/iGdvGbCEKGo

External libraries

- ***https://github.com/johannilsson/android-pulltorefresh***
- ***https://github.com/castorflex/SmoothProgressBar***

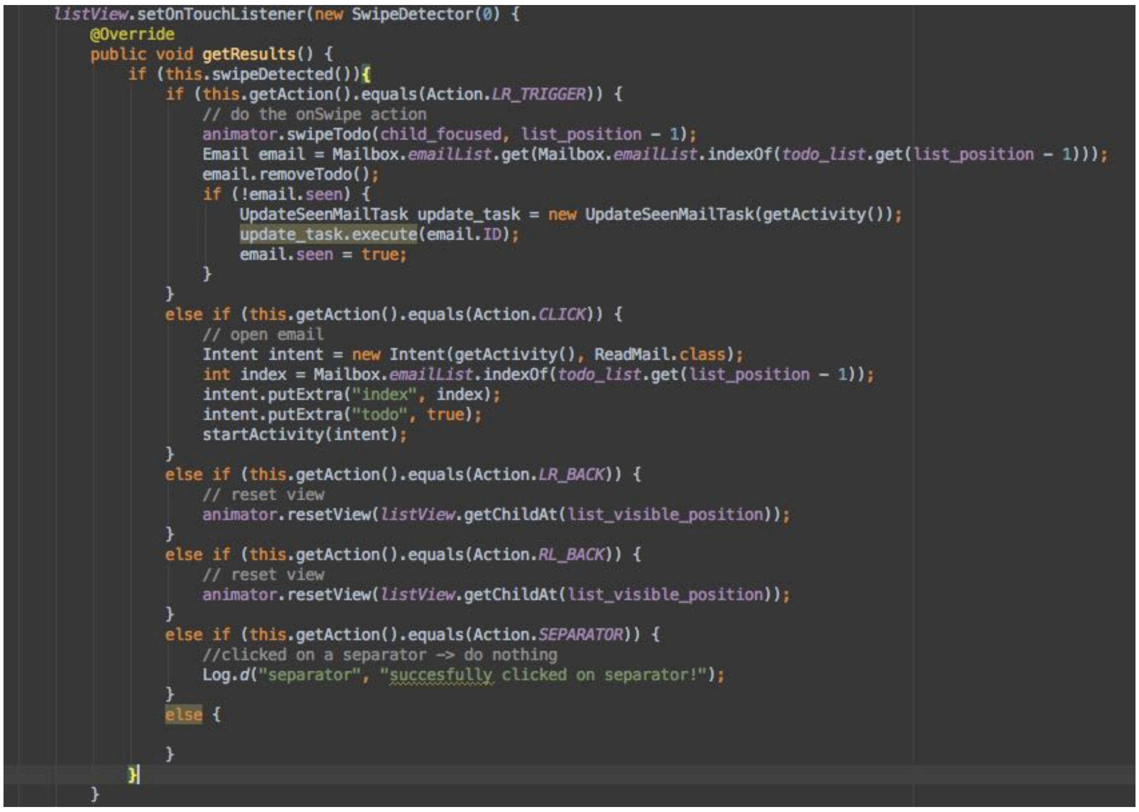
- ***https://github.com/derekbrameyer/android-betterpickers***

Similar apps

- www.mailpilot.co
- http://evomail.io/
- http://www.mailboxapp.com/

Code Appendix

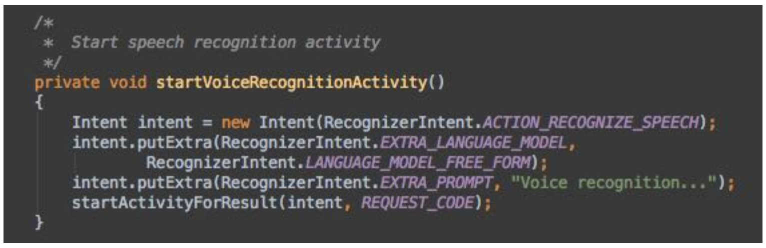
Swipe gestures



```
listView.setOnTouchListener(new SwipeDetector(0) {
    @Override
    public void getResults() {
        if (this.swipeDetected()){
            if (this.getAction().equals(Action.LR_TRIGGER)) {
                // do the onSwipe action
                animator.swipeTodo(child_focused, list_position - 1);
                Email email = Mailbox.emailList.get(Mailbox.emailList.indexOf(todo_list.get(list_position - 1)));
                email.removeTodo();
                if (!email.seen) {
                    UpdateSeenMailTask update_task = new UpdateSeenMailTask(getActivity());
                    update_task.execute(email.ID);
                    email.seen = true;
                }
            }
            else if (this.getAction().equals(Action.CLICK)) {
                // open email
                Intent intent = new Intent(getActivity(), ReadMail.class);
                int index = Mailbox.emailList.indexOf(todo_list.get(list_position - 1));
                intent.putExtra("index", index);
                intent.putExtra("todo", true);
                startActivity(intent);
            }
            else if (this.getAction().equals(Action.LR_BACK)) {
                // reset view
                animator.resetView(listView.getChildAt(list_visible_position));
            }
            else if (this.getAction().equals(Action.RL_BACK)) {
                // reset view
                animator.resetView(listView.getChildAt(list_visible_position));
            }
            else if (this.getAction().equals(Action.SEPARATOR)) {
                //clicked on a separator -> do nothing
                Log.d("separator", "succesfully clicked on separator!");
            }
            else {

            }
        }
    }
```

Voice recognition



```
/*
 *  Start speech recognition activity
 */
private void startVoiceRecognitionActivity()
{
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
            RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT, "Voice recognition...");
    startActivityForResult(intent, REQUEST_CODE);
}
```