Developing reactive multi-platform application in scala.js

By Milan Satala

Content

- Angular.js (Ionic framework) + scala.js
- Reactive GUI with scala.rx
- Client/Server source code sharing

Introduction

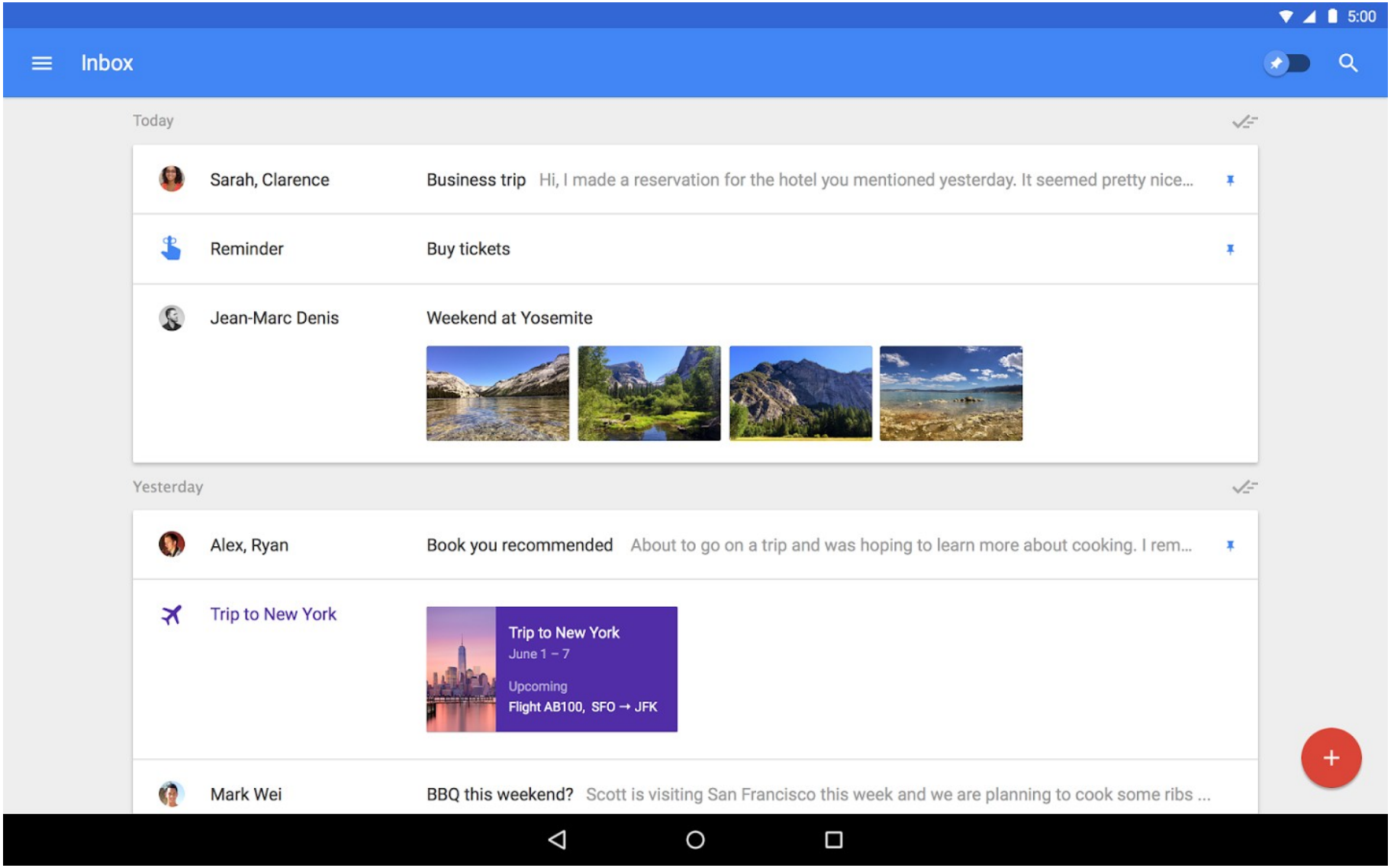
- Java EE programmer for 10 years
- One year break from work

Georgia-Armenia-Kurdistan

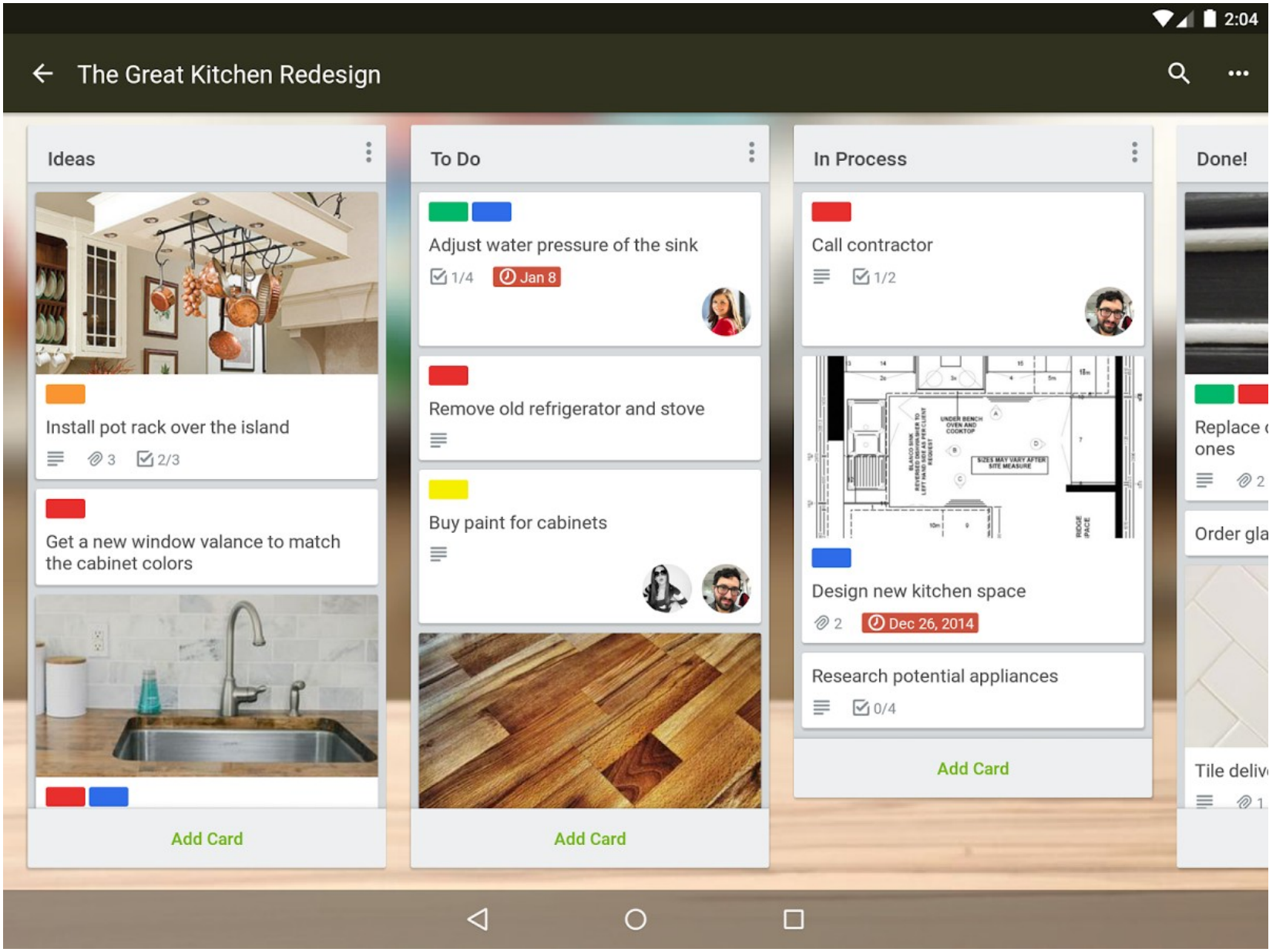
Google (and similar) like apps

- Apps for all platforms
- Unified design
- Respects specifics of platforms
- Takes advantage of platform possibilities

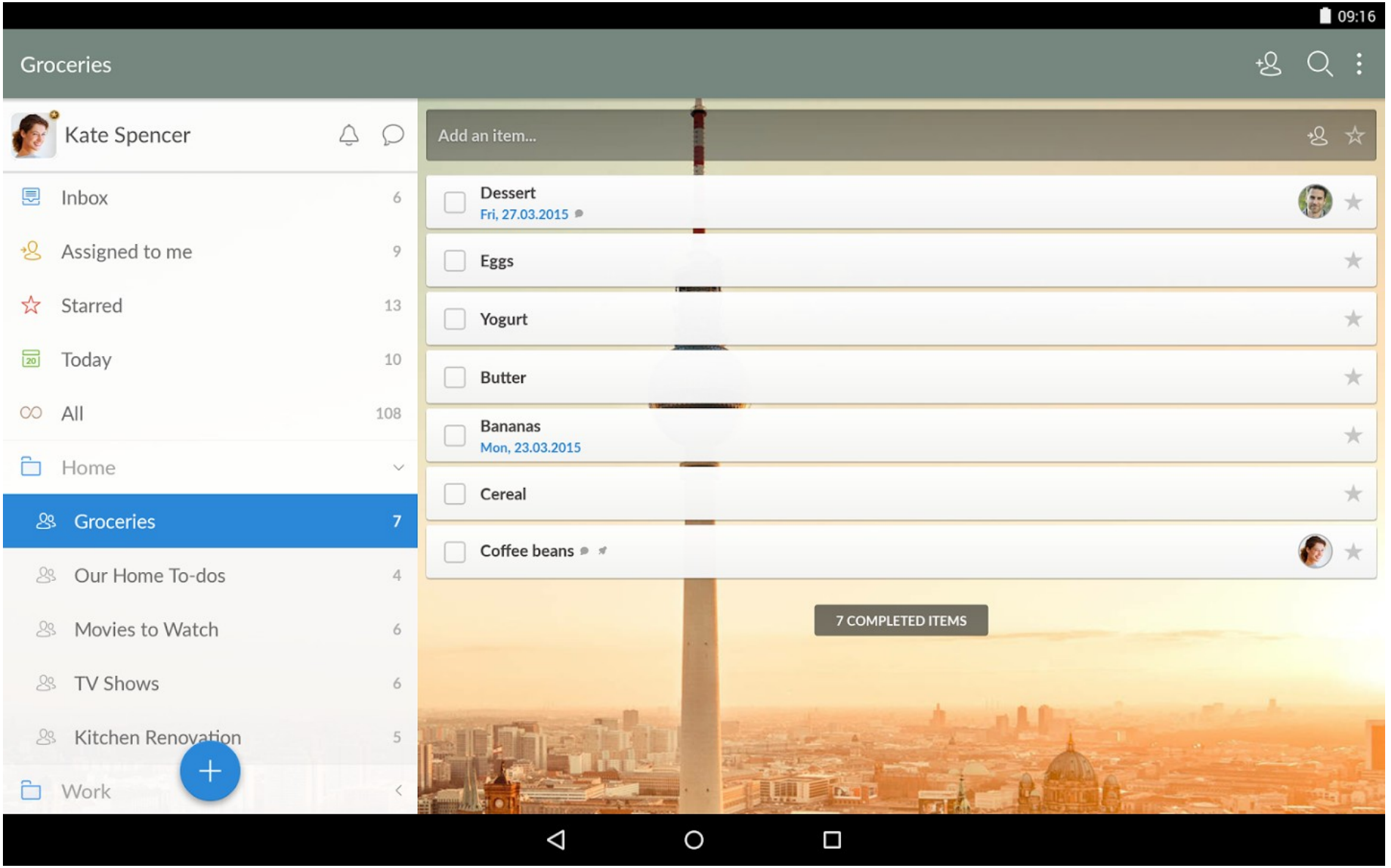
Inbox


5:00
Inbox
Today
Sarah, Clarence
Business trip
Hi, I made a reservation for the hotel you mentioned yesterday. It seemed pretty nice...
Reminder
Buy tickets
Jean-Marc Denis
Weekend at Yosemite
Yesterday
Alex, Ryan
Book you recommended
About to go on a trip and was hoping to learn more about cooking. I rem...
Trip to New York
Trip to New York
June 1 – 7
Upcoming
Flight AB100, SFO → JFK
Mark Wei
BBQ this weekend?
Scott is visiting San Francisco this week and we are planning to cook some ribs ...

Trello


2:04
The Great Kitchen Redesign
Ideas
Install pot rack over the island
3
2/3
Get a new window valance to match the cabinet colors
Add Card
To Do
Adjust water pressure of the sink
1/4
Jan 8
Remove old refrigerator and stove
Buy paint for cabinets
Add Card
In Process
Call contractor
1/2
Design new kitchen space
2
Dec 26, 2014
Research potential appliances
0/4
Add Card
Done!
Replace
ones
2
Order gla
Tile deliv
1

Wunderlist


09:16
Groceries
Kate Spencer
Inbox 6
Assigned to me 9
Starred 13
Today 10
All 108
Home
Groceries 7
Our Home To-dos 4
Movies to Watch 6
TV Shows 6
Kitchen Renovation 5
Work
Add an item...
Dessert
Fri, 27.03.2015
Eggs
Yogurt
Butter
Bananas
Mon, 23.03.2015
Cereal
Coffee beans
7 COMPLETED ITEMS

Instant reactions

- <100ms – user perceives as instantaneous
- >1s – flow of thought is broken
- Loading – features gradually become available
- User actions – instantaneous, independent of network speed

Moment of realization

Ionic framework

- Angular.js
- CSS and directives for mobile components
- Coordova
  - Native apps
  - Fullscreen WebView
  - JS binding for native features

Scala.rx

```
import rx._
val a = Var(1); val b = Var(2)
val c = Rx{ a() + b() }
println(c()) // 3
a() = 4
println(c()) // 6
```

Source code sharing

- No push&pull
- Synchronization – only changes are transferred
- Same algorithms on both ends
- Autowire
- JsActors

Shameless self-promo

Thank you!
Questions?