Több app

Egy kódbázis

Supercharge

Agenda

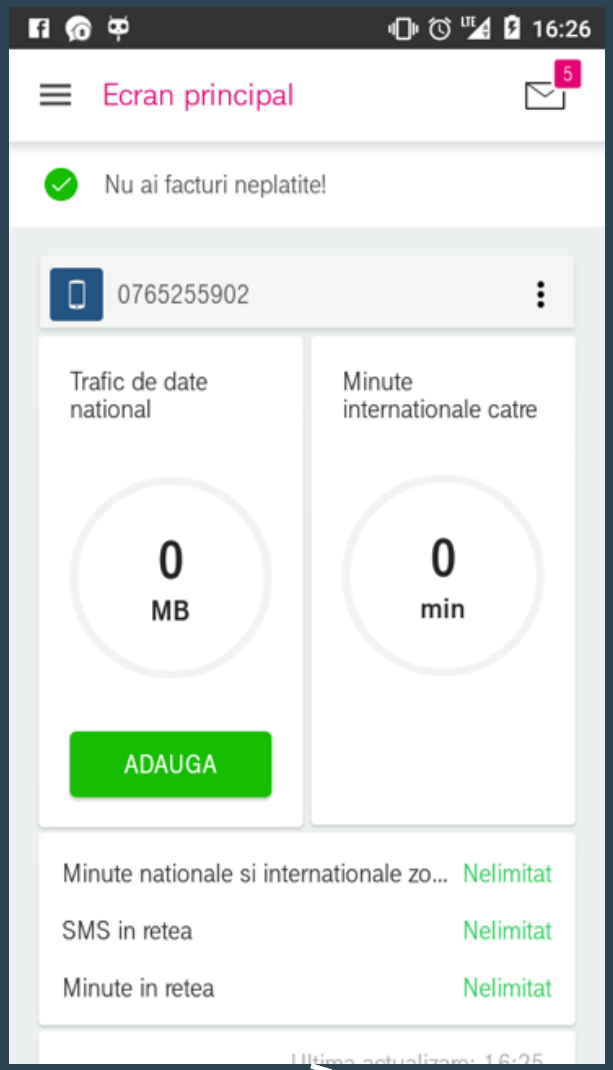
- Bevezető
- Technology stack
- A kód szervezése
- Debug és tesztelés
- CI

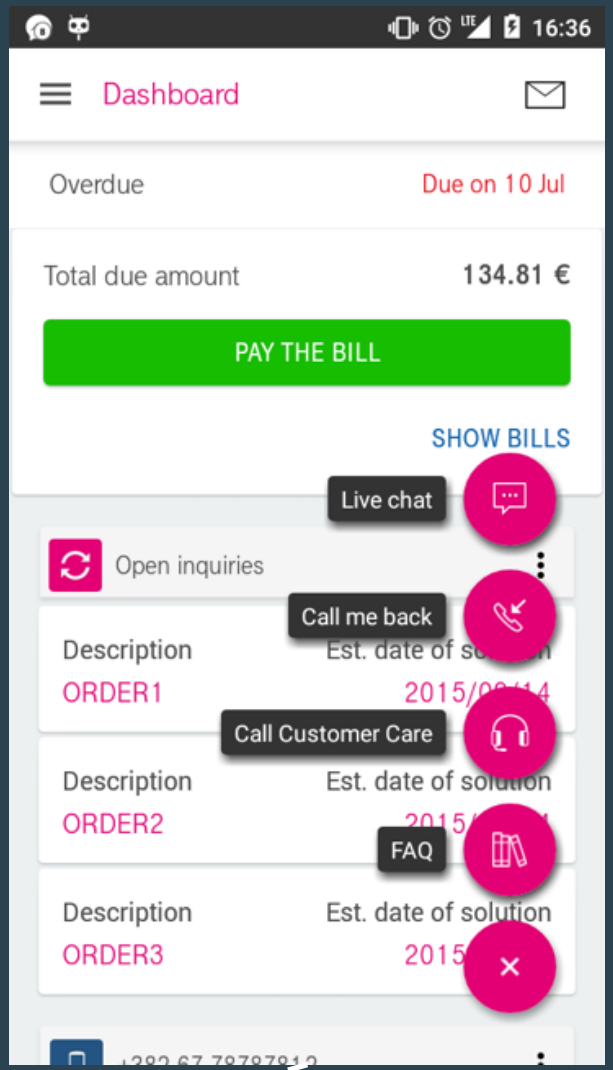
Bevezető

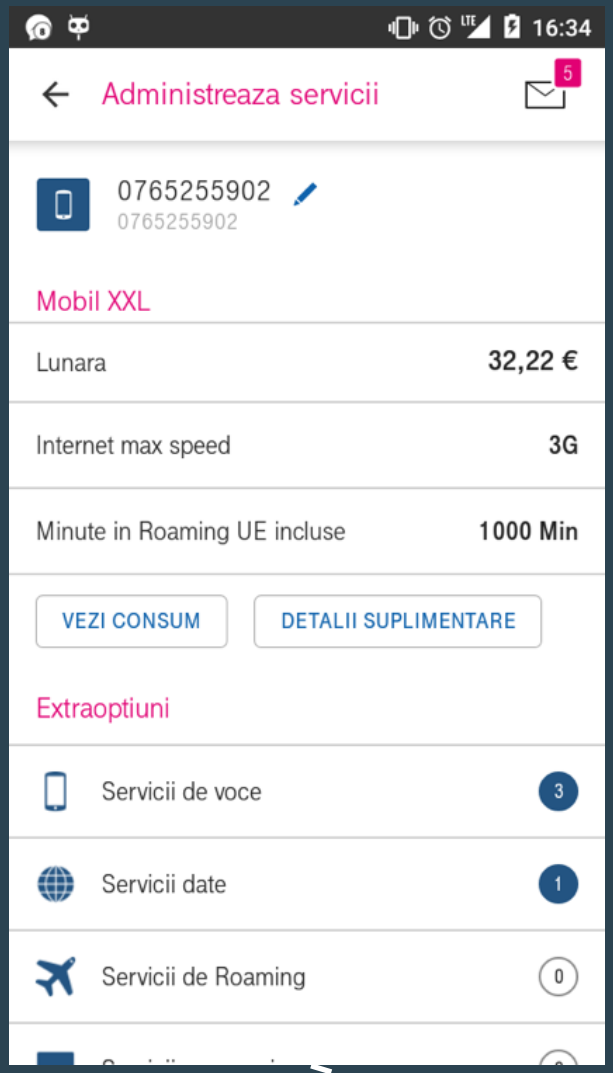
- Adott egy vezető telekommunikációs vállalat Self-care alkalmazása

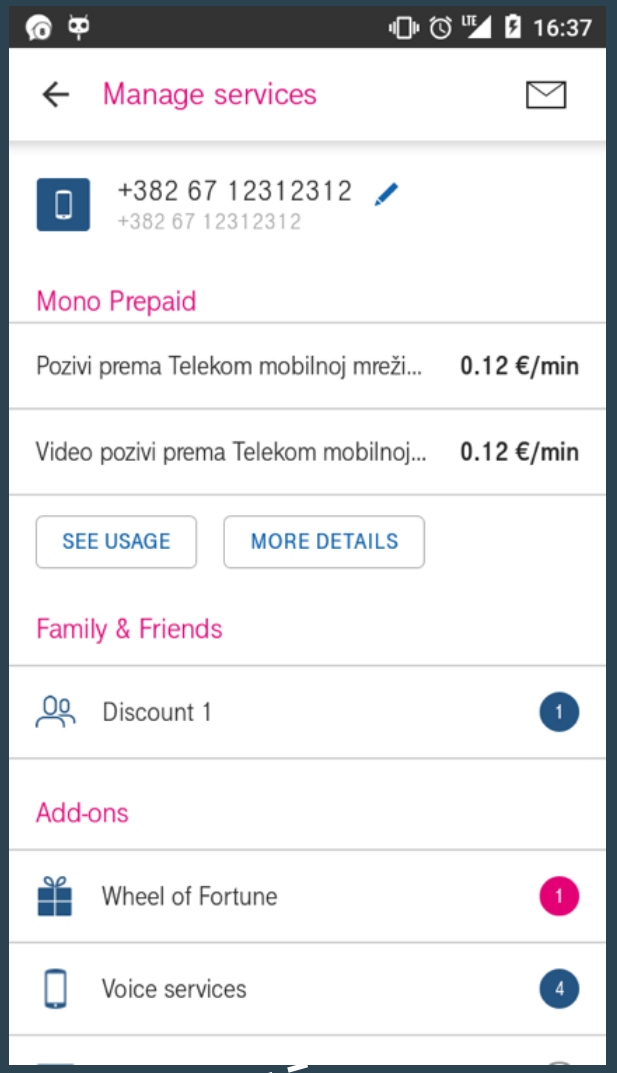
- Ezzel az alkalmazással a felhasználó:
  - Befizetheti a számláit
  - Módosíthatja szerződéseit
  - Megnézheti az adatforgalmát

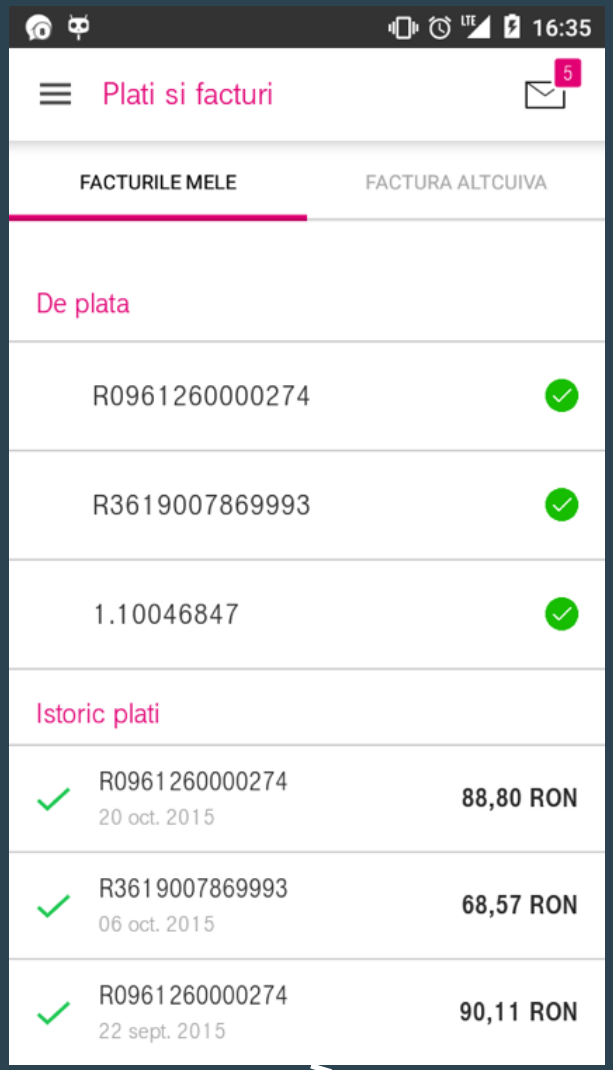
  …stb.

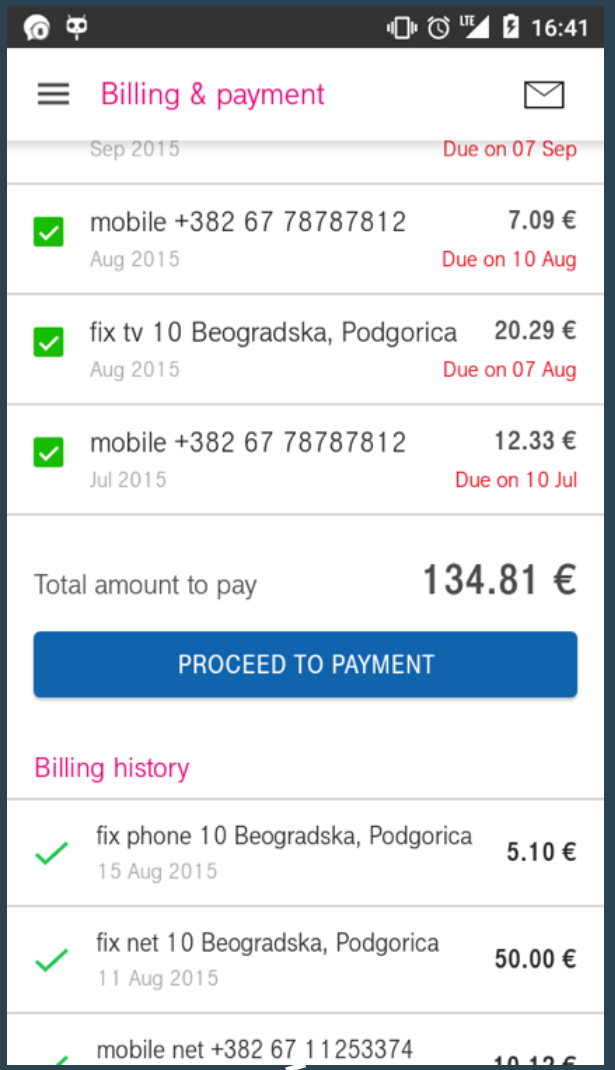

16:26
Ecran principal
5
Nu ai facturi neplatite!
0765255902
Trafic de date national
0
MB
Minute internationale catre
0
min
ADAUGA
Minute nationale si internationale zo... Nelimitat
SMS in retea Nelimitat
Minute in retea Nelimitat


16:36
Dashboard
Overdue
Due on 10 Jul
Total due amount
134.81 €
PAY THE BILL
SHOW BILLS
Live chat
Open inquiries
Call me back
Description
ORDER1
Call Customer Care
Description
Est. date of solution
ORDER2
FAQ
Description
Est. date of solution
ORDER3


16:34
Administreaza servicii
5
0765255902
0765255902
Mobil XXL
Lunara
32,22 €
Internet max speed
3G
Minute in Roaming UE incluse
1000 Min
VEZI CONSUM
DETALII SUPLIMENTARE
Extraoptiuni
Servicii de voce
3
Servicii date
1
Servicii de Roaming
0


16:37
Manage services
+382 67 12312312
+382 67 12312312
Mono Prepaid
Pozivi prema Telekom mobilnoj mreži...
0.12 €/min
Video pozivi prema Telekom mobilnoj...
0.12 €/min
SEE USAGE
MORE DETAILS
Family & Friends
Discount 1
1
Add-ons
Wheel of Fortune
1
Voice services
4


16:35
Plati si facturi
5
FACTURILE MELE
FACTURA ALTCUIVA
De plata
R0961260000274
R3619007869993
1.10046847
Istoric plati
R0961260000274
20 oct. 2015
88,80 RON
R3619007869993
06 oct. 2015
68,57 RON
R0961260000274
22 sept. 2015
90,11 RON


16:41
Billing & payment
Sep 2015
Due on 07 Sep
mobile +382 67 78787812
Aug 2015
7.09 €
Due on 10 Aug
fix tv 10 Beogradska, Podgorica
Aug 2015
20.29 €
Due on 07 Aug
mobile +382 67 78787812
Jul 2015
12.33 €
Due on 10 Jul
Total amount to pay
134.81 €
PROCEED TO PAYMENT
Billing history
fix phone 10 Beogradska, Podgorica
15 Aug 2015
5.10 €
fix net 10 Beogradska, Podgorica
11 Aug 2015
50.00 €
mobile net +382 67 11253374

Problémák

- Különböző szerverekkel való kommunikáció

Problémák

- Különböző szerverekkel való kommunikáció
- Feature
  - Ugyanaz, ugyanúgy

Problémák

- Különböző szerverekkel való kommunikáció
- Feature
  - Ugyanaz, ugyanúgy
  - Az egyikbe kell, másikba nem

Problémák

- Különböző szerverekkel való kommunikáció
- Feature
  - Ugyanaz, ugyanúgy
  - Az egyikbe kell, másikba nem
  - Mindegyikbe kell, de eltérően

Problémák

- Különböző szerverekkel való kommunikáció
- Feature
  - Ugyanaz, ugyanúgy
  - Az egyikbe kell, másikba nem
  - Mindegyikbe kell, de eltérően
- Navigáció

Problémák

- Különböző szerverekkel való kommunikáció
- Feature
  - Ugyanaz, ugyanúgy
  - Az egyikbe kell, másikba nem
  - Mindegyikbe kell, de eltérően
- Navigáció
- Lokalizáció
  - Pénznemek
  - Dátumok
  - Fordítások

Problémák

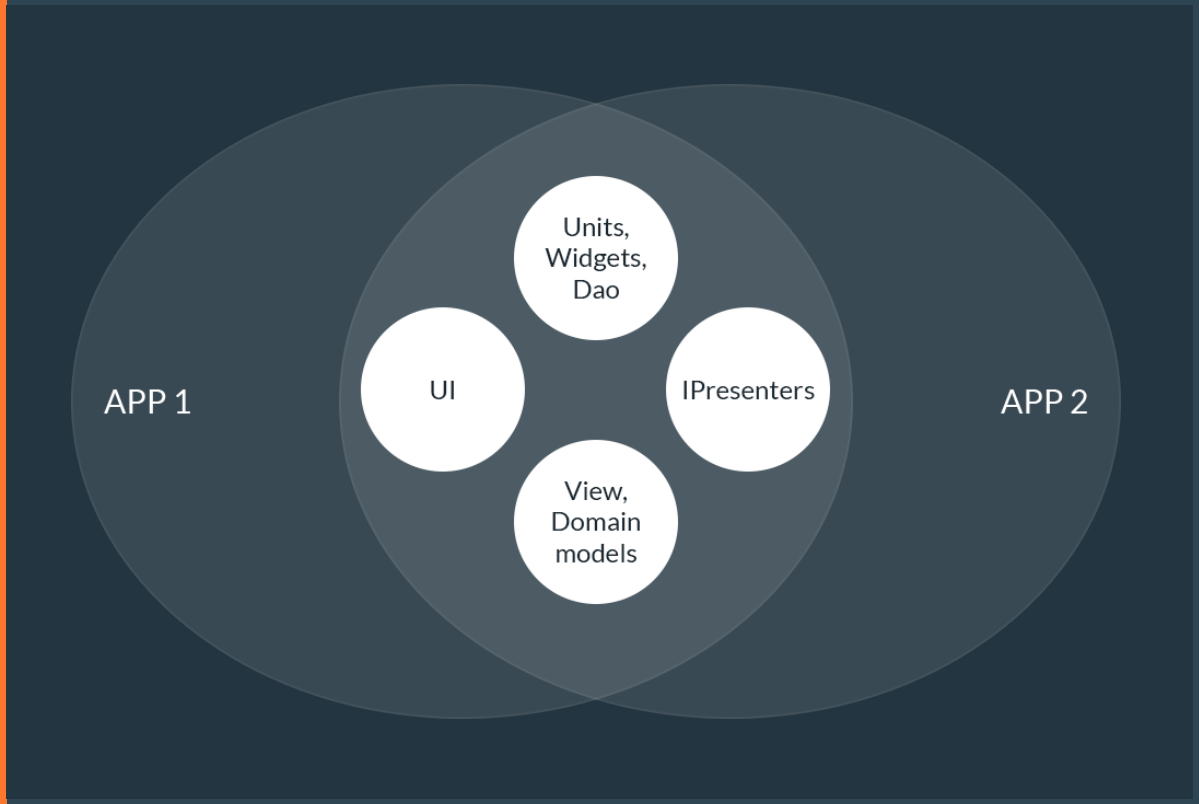
- Különböző szerverekkel való kommunikáció
- Feature
  - Ugyanaz, ugyanúgy
  - Az egyikbe kell, másikba nem
  - Mindegyikbe kell, de eltérően
- Navigáció
- Lokalizáció
  - Pénznemek
  - Dátumok
  - Fordítások

…és nap mint nap más probléma

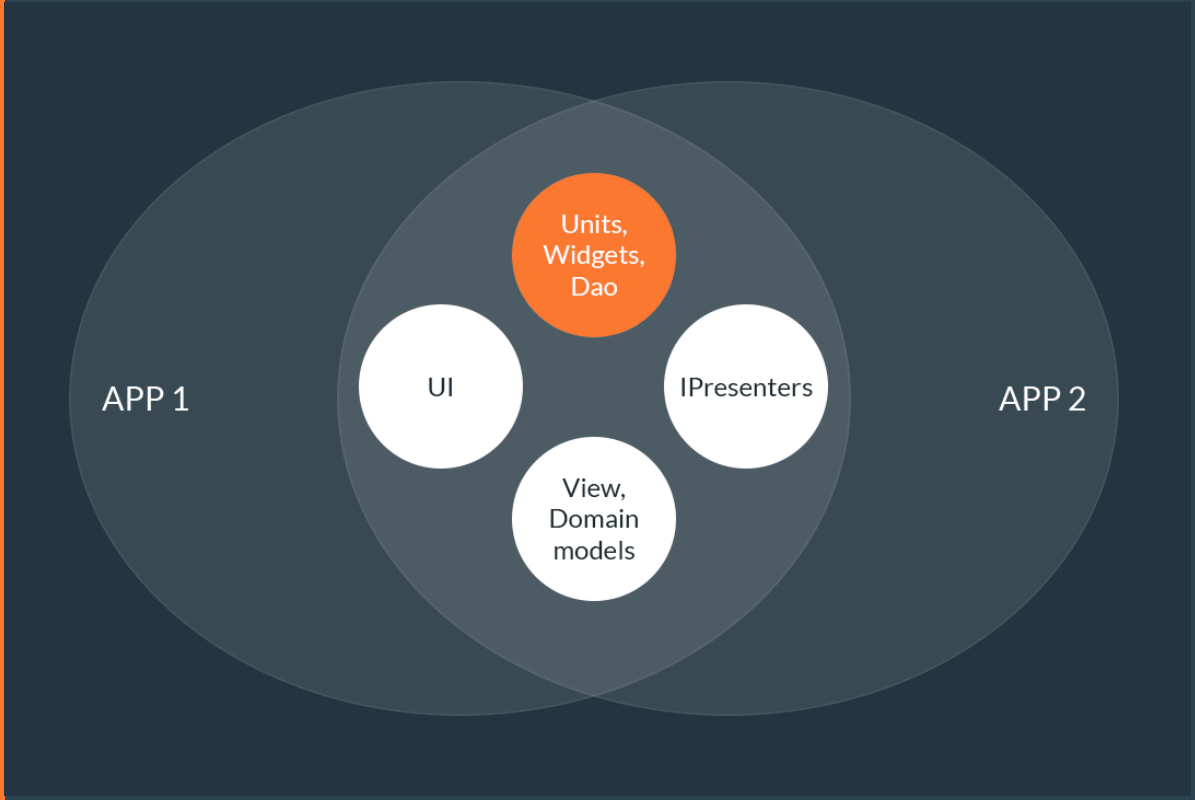
Technology Stack

- Retrolambda
- RxJava – RxAndroid – RxLifeCycle – RxBinding
- ActiveAndroid (ORM)
- Dagger (v1/v2)
- Retrofit
- JodaTime

Modulok


APP 1
APP 2
Units, Widgets, Dao
UI
IPresenters
View, Domain models

Modulok



Utils, Widgets, Dao

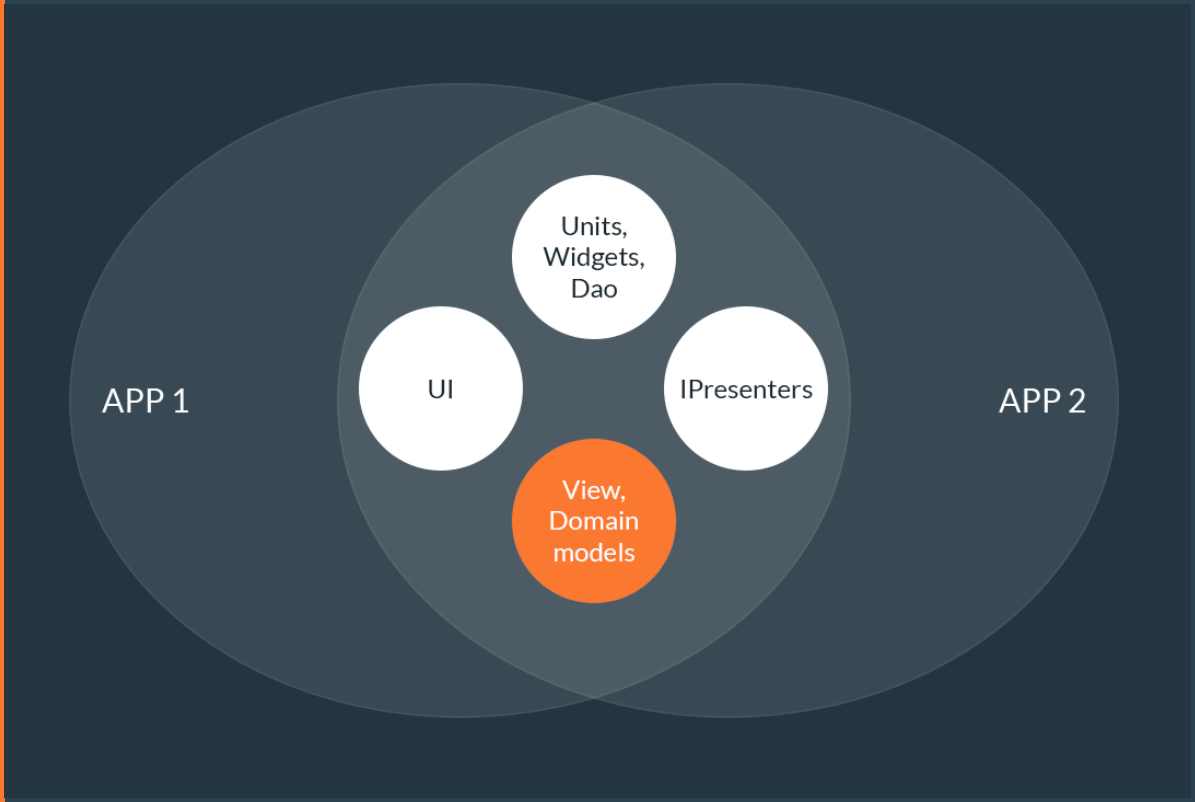
- Java és Android megoldások

“Minden amire egyszer rákerestél a stackoverflow-n és működött!”

- IConfig
- INavigator
- ICurrency
- IDatePrinter

…stb.

Modulok

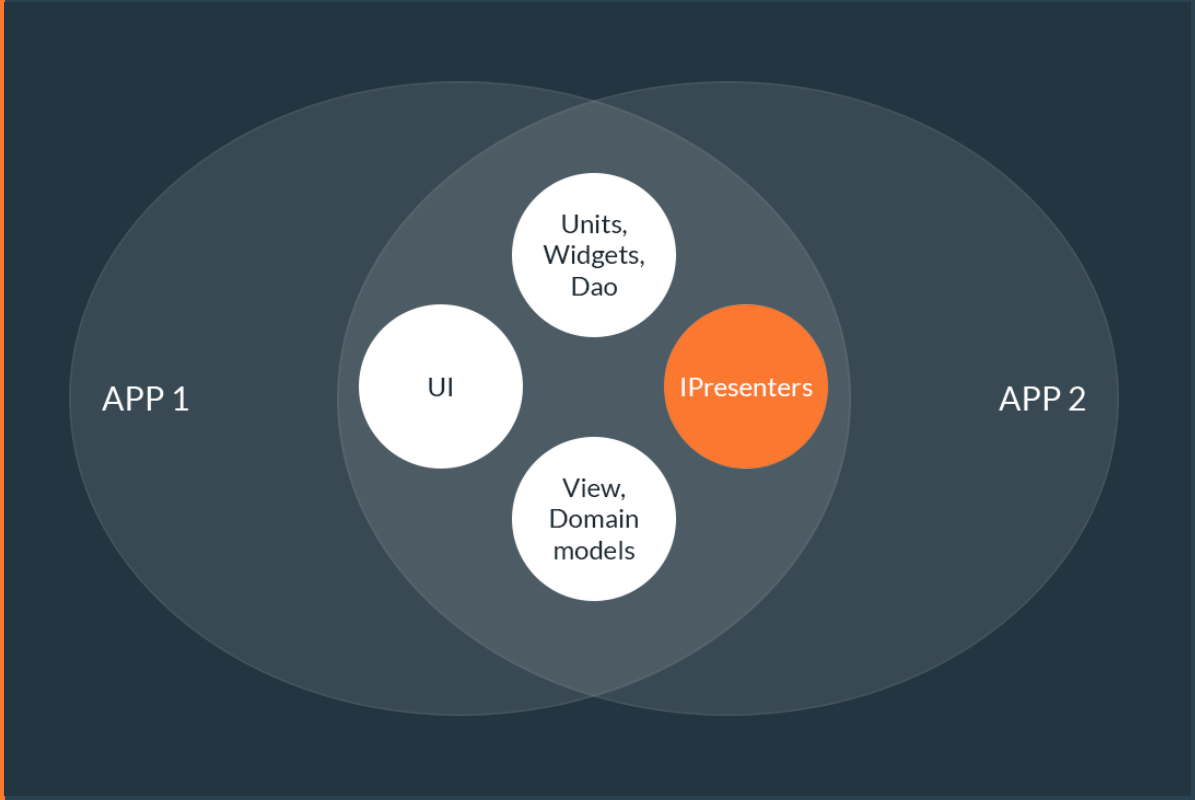


Models

3 fajta model:

- API
- Domain
- View

Modulok

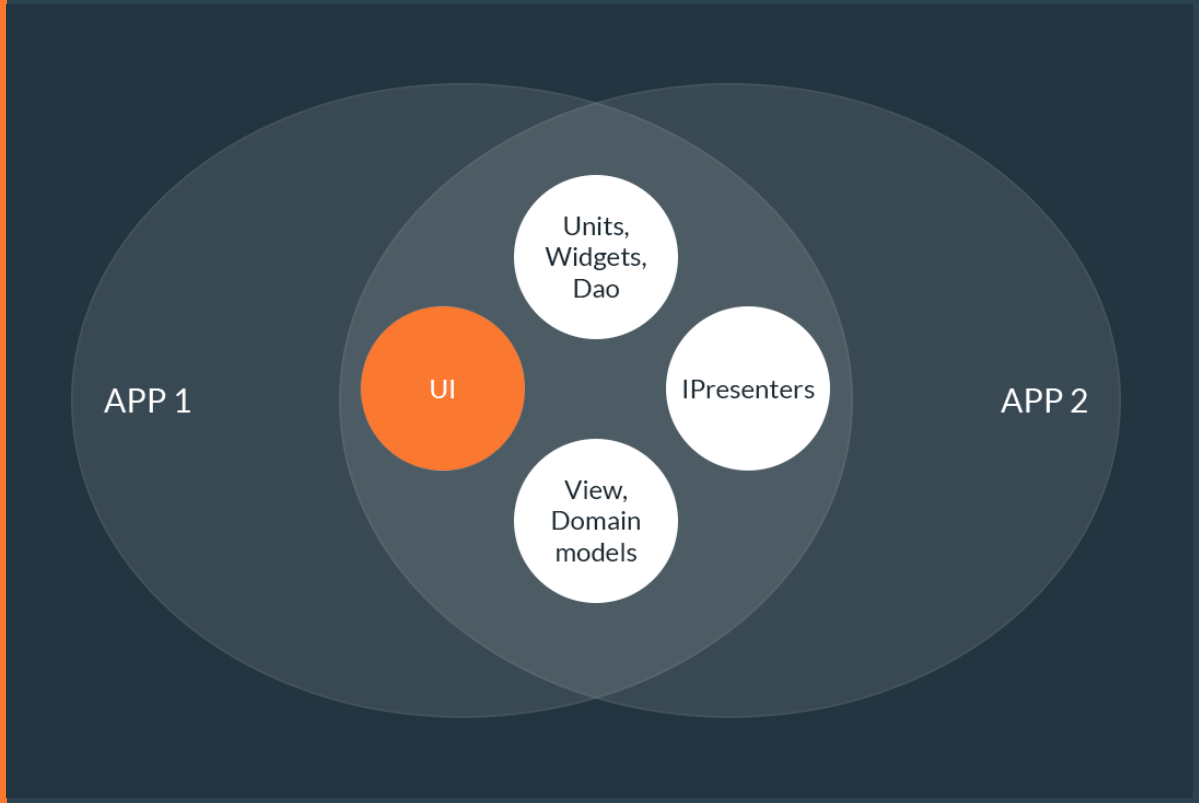


Presenters

Felelősek azért hogy előállítsák az adatokat a UI rétegnek.

Facade (Homlokzat) design pattern

Modulok

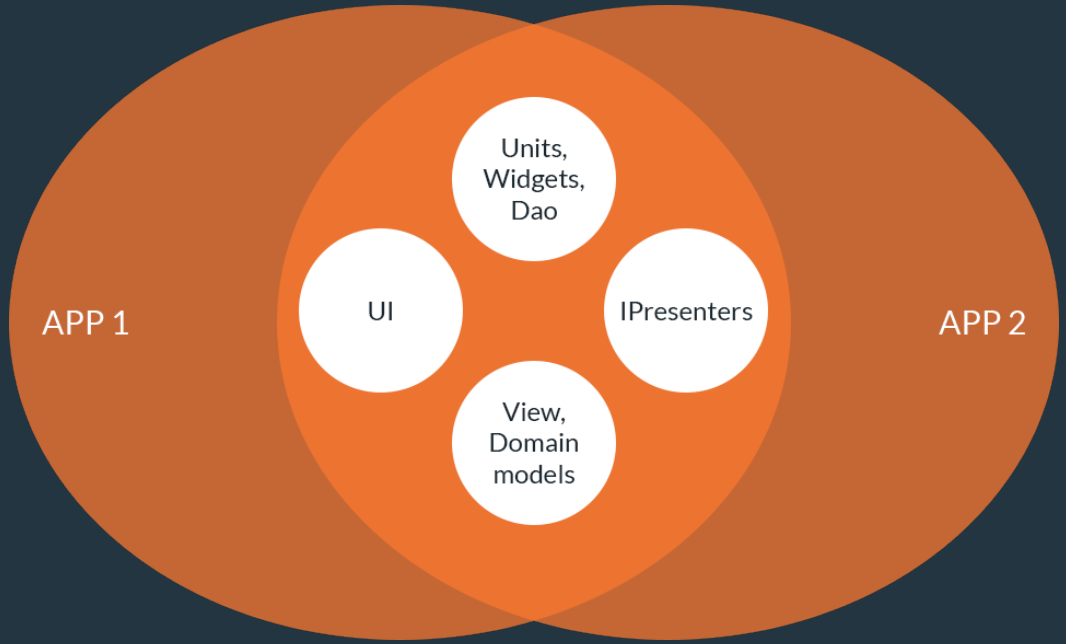


UI

Fragmentek

Activityk

Modulok



App

- Dependecy Graph felépítése
- Feature Set összeállítása
- Navigáció összeállítása
- API kommunikáció (Repository)
- Interaktorok
- Lokalizáció
- App specifikus feature, screen-flow

Feature Flagging

- Interfész, minden feature-ről

Feature Flagging

- Interfész, minden feature-ről
- Minden app-ben, ott van az összes feature, összes kód

Feature Flagging

- Interfész, minden feature-ről
- Minden app-ben, ott van az összes feature, összes kód
- A UI rétegben kezeljük, és főleg komponensek megjelenésének, viselkedésének szabályozására

Feature Flagging

- Interfész, minden feature-ről
- Minden app-ben, ott van az összes feature, összes kód
- A UI rétegben kezeljük, és főleg komponensek megjelenésének, viselkedésének szabályozására
- Példa: olyan FeatureFlagger implementáció ami push notification-re bekapcsol egy feature-t a felhasználók X %-nál

Packages

- Package by layer
  - Rétegek szerint vannak csoportosítva az egymástól kvázi független elemek
  - Fejlesztés közben ugrálni kell a package-k között
  - Ha egy feature-t ki kell venni, azt kb lehetetlen egyetlen művelettel
  - Gyenge modularitás

Packages

- Package by feature
  - A feature-k packagek szerint vannak csoportosítva
  - Minimalizálja a szkópot
  - Könnyű fejlesztés közben navigálni
  - Erős modularitás
  - Feature team-ek

Packages

- Package by feature
  - A feature-k packagek szerint vannak csoportosítva
  - Minimalizálja a szkópot
  - Könnyű fejlesztés közben navigálni
  - Erős modularitás
  - Feature team-ek

SOLID

- Single Responsibility
  - A class should have only one reason to change.
- Open Close
  - Classes, modules and fuctions should be open for extension, but closed for modifications.
- Liskov's Substitution
  - We must make sure that new derived classes are extending the base classes without changing their behavior.
- Interface Segregation
  - Clients should not be forced to depend upon interfaces that they don't use.
- Dependency Inversion
  - High-level modules should not depend on low-level modules. Both should depend on abstractions. Abstractions should not depend on details. Details should depend on abstractions.

Dependency Inversion

```
class Worker {
    public void work(){
        //dolgozik
    }
}
class Manager {
    Worker worker;
    public void setWorker(Worker w){ worker = w; }
    public void manage(){ worker.work; }
}
```

Dependency Inversion

```
class Worker {
    public void work(){
        //dolgozik
    }
}
class Manager {
    Worker worker;
    public void setWorker(Worker w){ worker = w; }
    public void manage(){ worker.work; }
}

class SuperchargeWorker {
    public void work(){
        //sokat dolgozik
    }
}
```

Dependency Inversion

```
interface IWorker {
     public void work();
}

class Worker implements IWorker {
     public void work(){
          //dolgozik
     }
}
class Manager {
     IWorker worker;
     public void setWorker(IWorker w){ worker = w; }
     public void manage(){ worker.work; }
}


class SuperchargeWorker implements IWorker {
     public void work(){
          //sokat dolgozik
     }
}
```

Dependency Inversion

```
interface IWorker {
     public void work();
}

class Worker implements IWorker {
     public void work(){
          //dolgozik
     }
}
class Manager {
     IWorker worker;
     public void setWorker(IWorker w){ worker = w; }
     public void manage(){ worker.work; }
}


class SuperchargeWorker implements IWorker {
     public void work(){
          //sokat dolgozik
     }
}
```

Dependency Inversion

- Expect interface, provide implementation
- A DI megoldás:
  - Navigációra
  - Lokalizációra (Pénznem, Dátum)
  - Presenterekre

  …stb.

  DI nélkül ne is kezdj el fejleszteni manapság!

Tesztelés

- API: MockWebServer + Robolectric
- UI: Calabash, Robotium
- Util: JUnit

Dependency Injection a tesztekben is!
Újrafelhasználható tesztek.

Debug

- Timber
  - https://github.com/JakeWharton/timber
  - Logolni, logolni, logolni

- Stetho
  - https://github.com/facebook/stetho
  - Chrome Developer Tools-ban tudod nézni a működést
  - Hátrány: nagyon “fogja” az app-et

Continuous Integration

- Fejlesztjük a Core modult
- Merge Request ha kész egy feature
- Jenkins megnézi az MR-t hogy mergelhető-e
- Statikus kód elemzés Sonarqube-al és Android Lint-el
- Jenkins buildel, deployol a Core-ból egy Snapshot-ot a Nexus Maven snapshot reposity-ba
- Ez triggereli az összes app buildelését, hogy megnézzük az új snapshot-ban vannak-e breaking change-k

Release esetén release repository-ba deploy és Fabric test release deploy

Összefoglaló

- SOLID elvek tisztelete
- Pragmatikusság
- Cserkész szabály, Clean Code elvek
- Dependency Injection tool nélkül nem érdemes elkezdeni
- CI összelövésére időt kell szánni

Köszönöm a figyelmet!

We are hiring!

Richard Radics

Supercharge

Android Developer

richard.radics@supercharge.io // www.supercharge.io

Supercharge