

Technical Project Report - Android Module

PharmaNow

Subject: Computação Móvel

Date: Aveiro, 25/01/2022

Students: 88858: Pedro Valente Mateus
89222: Samuel Rodrigues Duarte

Project abstract: Esta aplicação tem como objetivo servir como um *hub* central para todas as farmácias de uma região colocarem os seus produtos à venda de forma rápida e acessível. Do outro lado, do lado do utilizador, este teria disponível os produtos colocados pelas farmácias podendo também consultar um mapa e ver as farmácias que estão mais perto de si e os seus respetivos produtos à venda. Sendo a compra feita na aplicação (compra não implementada) o utilizador apenas teria que se deslocar à farmácia para fazer recolha, esta compra é confirmada pelo funcionário com a simples leitura de QR Code.

Report contents:

[1 Application concept](#)

[2 Implemented solution](#)

[Architecture overview](#)

[Implemented interactions](#)

[Project Limitations](#)

[New features & changes after the project presentation](#)

[3 Conclusions and supporting resources](#)

[Lessons learned](#)

[Project resources](#)

[Reference materials](#)

1 Application concept

Esta aplicação tem o intuito de facilitar a venda e compra de medicamentos e produtos farmacêuticos. Para evitar as filas de espera numa farmácia o utilizador pode comprar os seus produtos desejados na aplicação podendo depois fazer a recolha a qualquer altura, para além disto as farmácias mais próximas de si são assinaladas no mapa e à distância de um click pode-se consultar os produtos das mesmas. Para as farmácias esta aplicação tem a utilidade de que podem expor os medicamentos que têm para venda evitando os momentos constrangedores de “Não temos esse produto disponível”. Por fim, de forma a facilitar a recolha e promover a poupança de tempo o utilizador após chegada à farmácia basta abrir o QR Code da sua compra (gerado pela aplicação) e apresentá-lo ao funcionário, o funcionário utiliza o leitor de QR Codes também integrado na aplicação e apenas acessível a *admins* (farmácias e desenvolvedores) para confirmar a compra e consultar os produtos a entregar.

2 Implemented solution

Architecture overview

Este projeto foi realizado tendo como base uma Fragment Navigation Drawer referida nos “Reference Materials”, sendo esta montada na MainActivity, a MainActivity cria de seguida uma instância do primeiro fragmento para este ser a primeira coisa que aparece no ecrã.

A MainActivity é a única Activity deste projeto para além das três utilizadas para fazer o Login do utilizador (sendo estas LoginActivity, SignupActivity e ResetPasswordActivity), de resto todas as páginas do projeto são Fragments. Fragments estes que são criados consoante a escolha do utilizador na NavigationDrawer, a navegação para cada um destes é de seguida feita através de *transactions*. Muitas das páginas da aplicação fazem uso de *RecyclerViews*, para o uso correto destas foram criados *Adapters* para cada tipo de dados de cada *RecyclerView*. Foi também nos *Adapters* que se fez a lógica do que acontecia ao se carregar num botão e algum tratamento de dados (como filtragem de dados) sendo que nos *Fragments* apenas se fez o requerimento dos dados à base de dados e população da lista usada para criar o *Adapter*. Alguns *Adapters* implementam a interface *Filterable* sendo esta usada para filtrar as listas com texto inserido pelo utilizador em tempo real.

Para tratamento de dados foram usadas classes próprias para cada tipo de dados a ser tratados, sendo estas:

- Product (Produto que está a ser disponibilizado pelas farmácias);
- Purchase (Compra feita por um utilizador);
- ShoppingCart (Carrinho de compras de um utilizador).

Não foi necessário criar um modelo para os *Users* pois foi utilizada a funcionalidade de *Authentication* da *Firebase*, esta providência uma instância da autenticação feita sendo possível aceder ao *email* ou *id* a qualquer momento, desta forma é possível associar os modelos anteriores a um *user* utilizando apenas um campo no modelo.

Já tendo sido criado um projeto na *Firebase* para utilizar a *Authentication*, utilizou-se esta mesma ferramenta para a persistência dos dados, através da *Realtime-Database*. Todos os

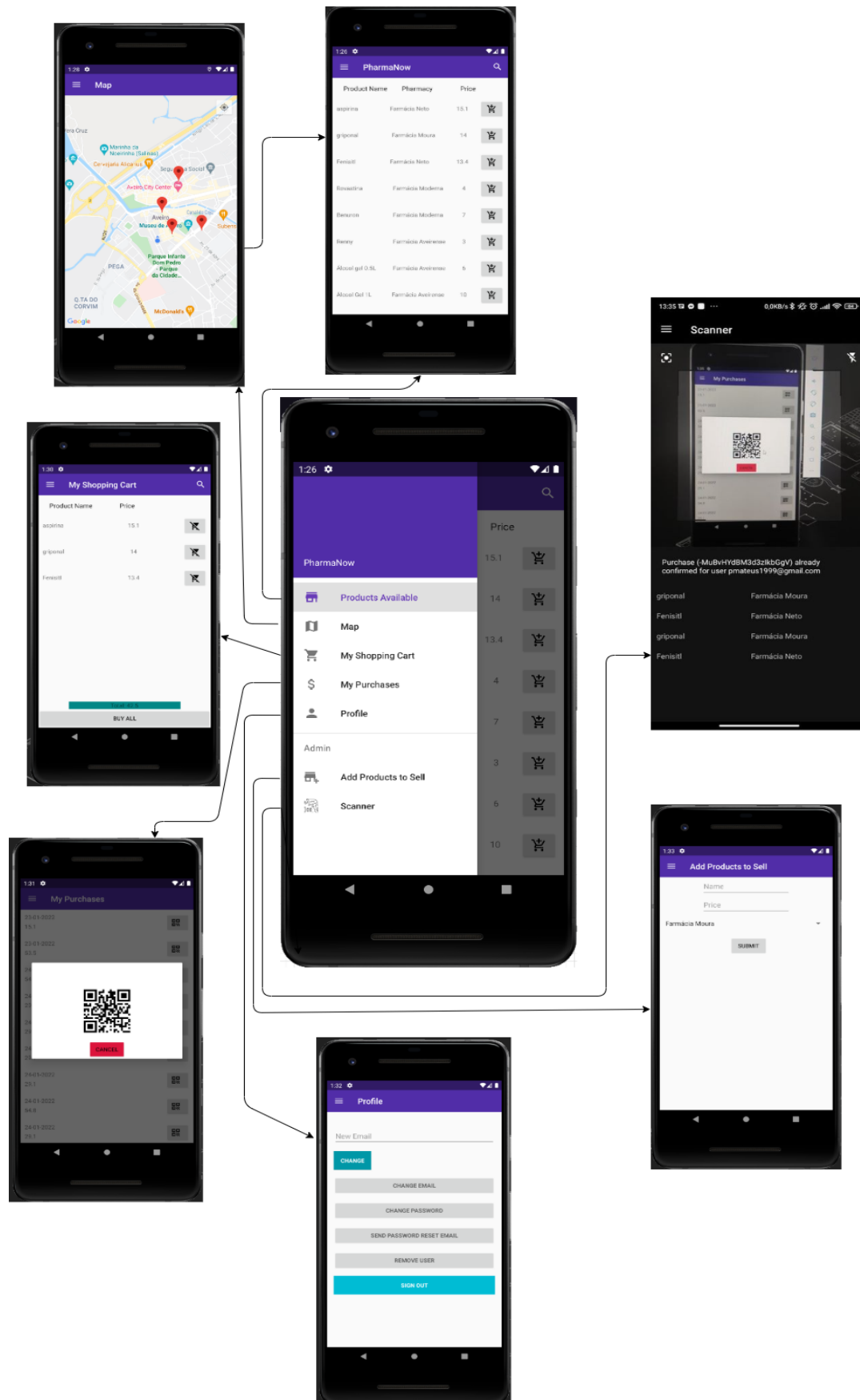
dados da aplicação vêm da base de dados, sendo que estes pedidos são feitos cada vez que o utilizador entra numa página garantindo assim que a aplicação está constantemente atualizada. É possível consultar, adicionar e remover dados da aplicação sendo as alterações feitas em *realtime* na base de dados, logo caso estejam dois utilizadores a consultar a aplicação os dados serão coerentes.

É possível consultar nos anexos um exemplo JSON de todos os tipos de dados guardados na base de dados, este JSON foi exportado diretamente da *Firebase* tendo sido apenas cortado alguns objetos para mais fácil consulta.

Por fim, fazendo uso do método *keepSynced* aplicável a uma referência de uma base de dados *Firebase* a aplicação faz *download* dos dados e mantém-nos sincronizados. Para além disto, enquanto a localização se mantém sincronizada não será removida da cache persistente do disco. Sendo assim, o utilizador apenas precisa de se ligar à internet uma vez após *login* (e para o primeiro *login*) para poder ter todos os dados da aplicação nesse momento, a partir daí poderá usar a aplicação como desejar, inclusive realizar compras sendo que estas serão depois registadas na base de dados quando este se ligar de novo à internet.

Foram criadas notificações para quando um novo produto é adicionado à base de dados, estas notificações usam um canal criado na *MainActivity* com o nome “M_CH_ID” com a sua importância definida para “HIGH”, cada vez que um produto é adicionado com sucesso o utilizador é notificado. Para além destas notificações, foi utilizado *Firebase Messaging* para enviar uma mensagem a todos os utilizadores às 18:10 a lembrar para verificar os produtos disponíveis, esta mensagem está programada para ser enviada todos os dias.

Implemented interactions



Invés de fazer um *flow* de navegação de página para página esta aplicação faz uso de uma *Navigation Drawer* possibilitando o utilizador de aceder a todas as páginas em qualquer parte, desta forma o utilizador não tem que decorar qualquer tipo de *flow* complexo ou fazer determinada atividade para chegar a outra. Posto isto, a aplicação possui as seguintes páginas:

1. **Products Available:** Página inicial da aplicação, aqui o utilizador pode consultar todos os produtos disponíveis de todas as farmácias. Poderá ver o nome do produto, bem como o seu preço e a farmácia que o está a vender. Escolhido o produto o utilizador poderá carregar no botão de adicionar ao carrinho de compras, adicionando assim o produto ao carrinho de compras que poderá ver noutra página dedicada ao mesmo. Sabendo que pesquisar consultando uma lista pode não ser muito amigável quando um utilizador já sabe o que quer, foi implementado uma forma de o utilizador pesquisar por texto clicando na lupa do topo direito, desta forma a lista será automaticamente filtrada procurando a *query* do utilizador nos nomes dos produtos.
2. **Map:** Nesta página é possível visualizar-se um mapa (integração com o google maps) onde estão marcadas quatro farmácias aveirenses, estas farmácias estão marcadas de forma *hardcoded* e utilizando os *markers* do *Google Maps*. Tem como propósito permitir ao utilizador ver quais são as farmácias mais perto de si para uma compra mais rápida e eficiente. Para facilitar, os *markers* são clicáveis apresentando uma página com os produtos filtrados por farmácia, de acordo com a farmácia escolhida.
3. **My Shopping Cart:** Nesta página o utilizador pode consultar os seus itens adicionados ao carrinho de compras (como referido no ponto 1) bem como remover os mesmos caso assim o deseje. O preço total é atualizado consoante os itens presentes, mesmo após removido. O utilizador pode filtrar os produtos utilizando a lupa do canto superior direito para aceder mais facilmente ao produto que quer remover. Por fim, pode comprar todos os itens presentes no carrinho, um *prompt* de confirmação aparece caso o utilizador carregue no botão por engano (funcionalidade de compra não implementada). Após efetuar a compra esta pode ser consultada na página de compras do utilizador.
4. **My Purchases:** Nesta página o utilizador pode consultar todas as suas compras, sendo estas apresentadas com a data de compra e o custo total. Para além disto, tem também um botão que gera um *popup* com um QRCode, este deve ser utilizado no momento de recolha dos produtos na farmácia, apresentando-o para leitura por parte do funcionário para confirmação.
5. **Profile:** Nesta página o utilizador pode gerir a sua conta, no que consta a mudar o *email/password*, remover a sua conta da base de dados da aplicação ou proceder ao *sign out*.
6. **Add Products to Sell:** Página restrita a utilizadores com permissões de *admin*. Aqui é possível adicionar produtos, colocando o seu nome, preço e associando-lhes uma farmácia. Após criação com sucesso de um produto uma notificação *push* é enviada informando a criação do mesmo.
7. **Scanner:** Página restrita a utilizadores com permissões de *admin*. Aqui é possível fazer o *scan* dos QRCodes gerados pela página **My Purchases**, a leitura do código é feita em *runtime*, ou seja, podem ser lidos códigos a qualquer momento que a informação é atualizada instantaneamente. Finda a leitura é disponibilizada no ecrã

a seguinte informação:

- a. Id da compra;
- b. Se a compra já foi confirmada (entregue) ou não;
- c. Email do utilizador que fez a compra;
- d. Lista dos produtos comprados para o funcionário da farmácia saber quais tem que entregar.

Para além das páginas aqui referidas existem mais três, sendo todas atividades. São as páginas de Login, SignUp e ResetPasswordActivity. O propósito destas páginas é o de, como diz o nome, efetuar o login, signup e fazer reset da password. Estas páginas têm permissões para se iniciar no *launch* juntamente com a MainActivity.

Project Limitations

Como o projeto é bastante baseado em listas, poderíamos ter implementado mais formas de filtragem para facilitar a pesquisa pelas mesmas. No uso do mapa, os *markers* das farmácias foram colocados de forma *hardcoded*, como planos futuros gostaríamos de utilizar a API Places para identificar as farmácias de forma dinâmica, essa opção foi explorada mas não utilizada por ter um plano pago.

No início do projeto foi utilizado um tutorial para fazer a NavigationDrawer, este utiliza *transactions* para fazer a navegação entre *fragments*, tal metodologia impossibilita a utilização de *navigation* com o grafo de navegação e ações de navegação. Sendo assim, refazer a NavigationDrawer seria algo importante.

Implementar um sistema de entrega por estafeta, em que seria possível ver o estafeta no mapa após uma compra e simular o seu trajeto até à entrega, só acabando a entrega após confirmação com qr code;

New features & changes after the project presentation

Não foram feitas alterações ao código após a entrega para além de remoção de *warnings* e reordenamento de código.

3 Conclusions and supporting resources

Lessons learned

O tema escolhido mostrou-se complicado no sentido que não havia fontes de dados externas (APIs) que permitissem ter informações mais pertinentes para popular a base de dados (mais informação sobre medicamentos).

As funcionalidades de autenticação e sincronização da *Firebase* foram sem dúvida uma surpresa pela positiva. Poder aceder a dados como id e email do utilizador que está *logged in* em qualquer parte do código é uma ferramenta muito poderosa para poder associar produtos a um utilizador específico. Para além disto a funcionalidade de sincronizar os dados mostrou-se extremamente útil, pois com código simples foi possível fazer download dos dados para a cache e a app passou a poder ser utilizada offline.

Project resources

Resource:	Available from:
Code repository:	https://github.com/o2valente/PharmaNow
Ready-to-deploy APK:	https://github.com/o2valente/PharmaNow/blob/master/pharmaNow.apk
App Store page:	--
Demo video:	https://github.com/o2valente/PharmaNow/blob/master/demo_pharmaNow.mp4

Reference materials

Esta aplicação foi baseada numa aplicação que implementa uma *NavigationDrawer*, o tutorial para esta implementação pode ser encontrado em <https://guides.codepath.com/android/Fragment-Navigation-Drawer#getting-references-to-the-header>.

A autenticação é algo bastante comum a todas as aplicações então não houve necessidade em reinventar a roda, como tal, o tutorial para ter autenticação com *Firebase* pode ser encontrado em <https://www.androidhive.info/2016/06/android-getting-started-firebase-simple-login-registration-auth/>.

Utilizar a app offline é algo crucial em todas as aplicações, caso se esteja a utilizar *Firebase* este tutorial pode vir a ser muito útil para manter os dados sincronizados <https://firebase.google.com/docs/database/android/offline-capabilities>.

Attachments

JSON exportado da base de dados utilizada.

```
{
  "Purchases" : {
    "-Mu6EhpTKcq8RY9xtlf8" : {
      "confirmed" : false,
      "date" : "23-01-2022",
      "id" : "-Mu6EhpTKcq8RY9xtlf8",
      "productsBought" : [ {
        "price" : "15.1",
        "productId" : "-Mu1VoTqjP4cQ_lHkhhP",
        "productName" : "aspirina",
        "productPharma" : "Farmácia Neto"
      }, {
        "price" : "13.4",
        "productId" : "-Mu2LEy0PWCoUBkdb5nT",
        "productName" : "Fenisitl",
        "productPharma" : "Farmácia Neto"
      }, {
        "price" : "4",
        "productId" : "-Mu2RHKun8bdocqQLjoG",
        "productName" : "Rovastina",
        "productPharma" : "Farmácia Moderna"
      } ],
      "totalCost" : 32.5,
      "user" : "pedro.valente@ua.pt"
    },
  },
  "ShoppingCart" : {
    "sVvU8EKbjvf5vkTOIHEF2QtKQn11" : {
      "productsList" : {
        "-Mu8BfXgENAmdbVjrRBy" : {
          "price" : "15.1",
          "productId" : "-Mu1VoTqjP4cQ_lHkhhP",
          "productName" : "aspirina",
          "productPharma" : "Farmácia Neto"
        },
        "-Mu8BfcqZ_SyZp7I4dZL" : {
          "price" : "14",
          "productId" : "-Mu1VtEJQPqrWKXBSlK1",
          "productName" : "griponal",
          "productPharma" : "Farmácia Moura"
        }
      }
    }
  }
}
```



```

    },
    "userEmail" : "samuel.acert@gmail.com"
  }
},
"admin" : "pmateus1999@gmail.com",
"products" : {
  "-Mu1VoTqjP4cQ_lHkhhP" : {
    "price" : "15.1",
    "productId" : "-Mu1VoTqjP4cQ_lHkhhP",
    "productName" : "aspirina",
    "productPharma" : "Farmácia Neto"
  },
  "-Mu1VtEJQPqrWKBs1K1" : {
    "price" : "14",
    "productId" : "-Mu1VtEJQPqrWKBs1K1",
    "productName" : "griponal",
    "productPharma" : "Farmácia Moura"
  },
  "-Mu2LEy0PWCoUBkdb5nT" : {
    "price" : "13.4",
    "productId" : "-Mu2LEy0PWCoUBkdb5nT",
    "productName" : "Fenisitl",
    "productPharma" : "Farmácia Neto"
  },
}
}
}

```