

Technical Project Report - Android Module

Pawesome Pets

Subject: Introdução à Computação Móvel

Date: Aveiro, 19/06/2024

Students: 108067: Mariana Figueiredo Perna
108782: Rafaela Espírito Santo Dias

Project abstract: Uma aplicação para gerenciar o estilo e qualidade de vida dos seus animais domésticos.

Report contents:

[1 Application concept](#)

[2 Implemented solution](#)

[Architecture overview \(technical design\)](#)

[Implemented interactions](#)

[Project Limitations](#)

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

[3 Conclusions and supporting resources](#)

[Lessons learned](#)

[Work distribution within the team](#)

[Project resources](#)

[Reference materials](#)

1 Application concept

A PawesomePets foi criada para ajudar todos os donos de animais de quatro patas a garantir a segurança e o bem-estar dos seus pets. A aplicação permite associar um identificador QR Code e NFC ao animal, para que, caso seja encontrado sozinho na rua, as suas informações pessoais estejam sempre acessíveis a qualquer utilizador da nossa plataforma. Os nossos utilizadores-alvo são todos os que possuem animais de estimação em casa e desejam proporcionar-lhes uma vida mais feliz e segura.

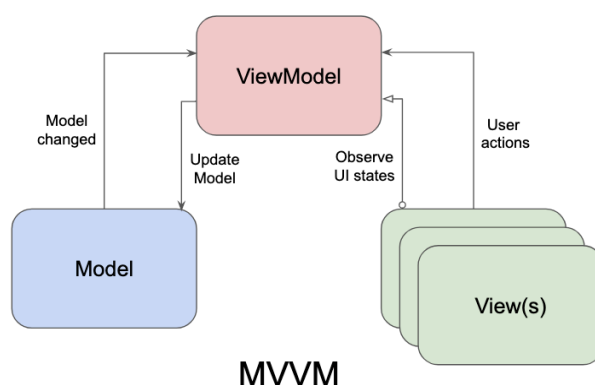
A nível dos benefícios, podemos dizer que a aplicação será bastante útil para garantir a segurança do seu pet, uma vez que cada animal possui um identificador QR Code e NFC. Isto assegura que, em caso de perda, as informações do pet sejam facilmente acessíveis, facilitando o seu regresso a casa. Além disso, a PawesomePets permite armazenar informações detalhadas de cada animal, incluindo fotografias, dados de saúde e histórico de vacinas. A aplicação oferece ainda acesso a um mapa onde os utilizadores podem marcar os percursos realizados com os seus cães, incentivando-os a sair mais frequentemente para passear com os seus animais.

A PawesomePets é, portanto, uma ferramenta essencial para todos os donos de animais que desejam assegurar o bem-estar dos seus companheiros de quatro patas e viver uma vida mais feliz ao lado deles.

2 Implemented solution

Architecture overview (technical design)

A arquitetura da aplicação PawesomePets segue o padrão MVVM (Model-View-ViewModel) com integração de diversos componentes para assegurar uma gestão eficiente de dados e funcionalidades. Abaixo está uma descrição detalhada das camadas/componentes utilizados, a interação entre os módulos e outras funcionalidades implementadas.



Camadas e Componentes Utilizados

1. Camada de Dados:

- **Modelos:** Representação das entidades da aplicação, como **Pet**, **Breed**, e **PetList**.
- **Room Database:** Utilizada para armazenamento local de dados com suporte para operações reativas através do uso de Flow e LiveData.
- **Repositórios:** Implementação do padrão Repository para abstrair a origem dos dados (local ou remoto).

2. Camada de Interação com API:

- **Retrofit:** Usado para chamadas a APIs remotas e integração com serviços externos.
- **Firebase:** Integração com Firebase Auth para autenticação, Firestore para armazenamento de dados na nuvem, e Storage para armazenamento de imagens.

3. Camada de Negócio:

- **ViewModel:** Utiliza LiveData e Flow para fornecer dados reativos às Views. Implementa a lógica de negócio e coordena a interação entre a View e o Repositório.
- **Koin:** Utilizado para injeção de dependências, garantindo a modularidade e testabilidade do código.

4. Camada de Apresentação:

- **Compose:** Utilizado para a construção da UI, promovendo um design declarativo e moderno.
- **Navegação:** Implementada com Navigation Compose para gerir a navegação entre as telas.

5. Funcionalidades Avançadas:

- **NFC e QR Code:** Leitura e geração de QR Codes e NFC para identificação dos pets.
- **CameraX:** Utilizado para capturar fotos dos pets.
- **Google Maps:** Integração com Google Maps para traçar percursos e encontrar serviços locais para pets.
- **Mapas e Localização:** Utilização de serviços de mapas e localização para funcionalidades de rastreamento e visualização de trajetos.

Implementação das Diretrizes de Arquitetura Android

1. Model-View-ViewModel (MVVM):

- **Model:** Definição das entidades e gestão dos dados.
- **View:** Composables do Jetpack Compose para a criação das interfaces.
- **ViewModel:** Gestão de estados e lógica de negócio, utilizando LiveData e Flow para reatividade.

2. LiveData/Flows:

- Utilização de LiveData para fornecer dados à UI de forma observável e Flow para operações assíncronas e reativas.

3. Room:

- Utilizado para armazenamento local, com DAOs definidos para operações de

CRUD (Create, Read, Update, Delete).

Estratégias de Persistência de Dados e Atualização de Conteúdo

1. Firebase Firestore:

- Armazenamento e sincronização de dados em tempo real, permitindo acesso offline com sincronização automática quando a conexão é restaurada. (Não ficou bem implementado)

2. Room Database:

- Armazenamento local para dados críticos, com integração ao Firestore para manter a consistência dos dados.

Estratégias de Design Avançadas

1. Integração com Serviços Externos:

- Retrofit para chamadas a APIs.
- Firebase para autenticação, armazenamento e notificações push.

2. Leitura de Sensores:

- Utilização de NFC para leitura de tags e identificação dos pets.

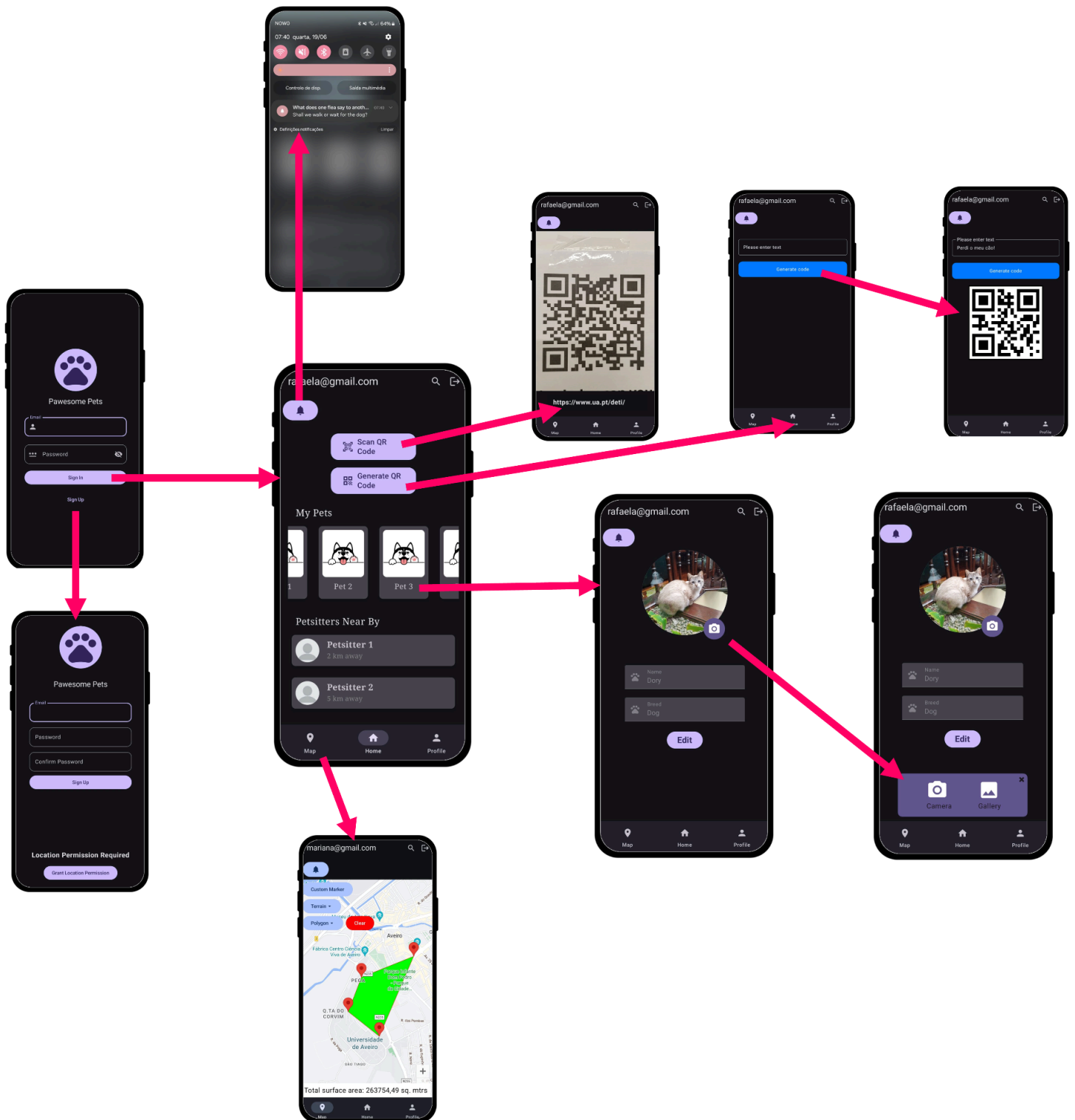
3. Mapeamento e Localização:

- Google Maps e serviços de localização para rastreamento de percursos e localização de serviços.

4. Notificações Locais:

- Usado para enviar notificações após clicar num botão que envia uma piada ao utilizador para o deixar de bom humor.

Implemented interactions [DIAGRAM]



Project Limitations

Escrita pelo NFC: Não conseguimos implementar a funcionalidade de escrita pelo NFC. Embora a leitura de tags NFC esteja funcional, a capacidade de escrever dados nas tags NFC ainda não foi desenvolvida.

Visualização dos Dados na Home Page: Não conseguimos implementar a visualização dos dados dos cães na página inicial. Apesar de os dados serem corretamente criados na UI e armazenados na base de dados Room, a exibição desses dados na página inicial ainda não está funcional.

Cloud Messaging: Não conseguimos implementar o Cloud Messaging para permitir que os utilizadores possam conversar entre si e receber notificações dessas conversas. A integração com o sistema de mensagens em tempo real e notificações push está planeada, mas ainda não foi concluída.

3 Conclusions and supporting resources

Principais Desafios:

Durante o processo de implementação, deparámo-nos com alguns problemas complexos. Um dos maiores desafios foi a navegação entre páginas. A criação de grafos de fluxo para gerir a navegação revelou-se um grande obstáculo. A documentação sobre a navegação e a implementação prática dessa funcionalidade apresentaram-se mais complexas do que inicialmente esperado. Para superar este problema, dedicámos tempo extra ao estudo das melhores práticas e ao entendimento detalhado dos componentes de navegação do Jetpack Compose. Além disso, a divisão do trabalho em pequenas tarefas e a colaboração constante dentro da equipa permitiram-nos resolver este desafio de forma eficaz mas trabalhosa.

Surpresas no Desenvolvimento Android:

Por outro lado, ficámos positivamente surpreendidas com a facilidade de integração da API do Google Maps. Esperávamos que a implementação e a personalização dos mapas fossem complexas, mas a documentação clara e as ferramentas fornecidas pela API tornaram este processo bastante direto e intuitivo. Conseguimos rapidamente adicionar funcionalidades de mapeamento e rastreamento de percursos, o que acrescentou um valor significativo à nossa aplicação.

Sugestões para Futuras Edições do Curso "Computação Móvel":

Da nossa experiência, gostaríamos de sugerir a inclusão do ensino de React Native no curso "Introdução à Computação Móvel". Acreditamos que React Native, sendo uma tecnologia amplamente utilizada para desenvolvimento de aplicações móveis multiplataforma, pode ser uma adição valiosa ao currículo. Não gostámos particularmente

da experiência com o Android Compose, pois considerámos que a sua complexidade é desnecessária para muitas aplicações. React Native, por outro lado, poderia oferecer uma curva de aprendizagem mais suave e uma experiência de desenvolvimento mais agradável.

Work distribution within the team

Tendo em consideração o desenvolvimento geral do projeto, a contribuição de cada membro da equipa foi distribuída da seguinte forma: Mariana Perna fez 50% do trabalho e Rafaela Dias contribuiu com 50%.

Project resources

Resource:	Available at:
Code repository:	<code>git@github.com:MarianaPerna07/PawesomePets.git</code>
Ready-to-deploy APK:	https://www.transfERNOW.net/dl/20240619Zlymdnai
App Store page:	<put URL, only if applicable>
Demo video:	<optional: link to a video demonstration of the app. Consider make a video demo if your project requires specific/complex setup , not easily replicable.>