

Stalk Tracker

Dário Matos (89288) e Pedro Almeida (89205) - Grupo 2

Computação Móvel

Universidade de Aveiro

Dário Matos



Pedro Almeida



Motivação

A motivação para a desenvolvimento da Stalk Tracker provém da ambição de ter uma solução de segurança pública, tanto a nível de interação com estranhos como a nível de transmissão de doenças e vírus, adicionando funcionalidades e dando possibilidade aos utilizadores de analisar os dados acerca do ambiente que os rodeia. O objetivo foi de desenvolver uma aplicação intuitiva, com UI simples e interação básica, para que possa ser utilizada por uma grande variedade da população, de forma a garantir os benefícios a utilizadores de todas as idades.

Funcionalidades e requisitos

De forma a garantir os objetivos que deram motivação para o desenvolvimento, o funcionamento da aplicação baseia-se no tracking de dispositivos utilizando a tecnologia Bluetooth, guardando estes dados e sendo possível relacionar informação proveniente do utilizador da app.

Assim, listamos os seguintes requisitos funcionais:

- Alertar o utilizador que um determinado dispositivo/perfil pré-definido se encontra nas redondezas
- Fornecer dados gerais sobre os dispositivos que rodeiam o utilizador
- Pedir auxílio imediato através de um botão de emergência
- Consultar o histórico de proximidade com outros dispositivos
- Adicionar determinados contactos a uma lista de amigos ou, pelo contrário, a uma black list, consoante o facto de ser um contacto de confiança ou de perigo.

Embora tenha sido um objetivo pensado inicialmente, a deteção da proximidade de outros dispositivos não foi descrita acima, embora esteja presente na aplicação. Isto porque a biblioteca utilizada (*bluetooth serial*) retorna um valor de proximidade bastante variável (visto que depende da intensidade do sinal, e esta é frequentemente alterada

por pequenos pormenores no ambiente em redor). Após o período de desenvolvimento podemos afirmar que a aplicação cumpre os objetivos inicialmente desejados.

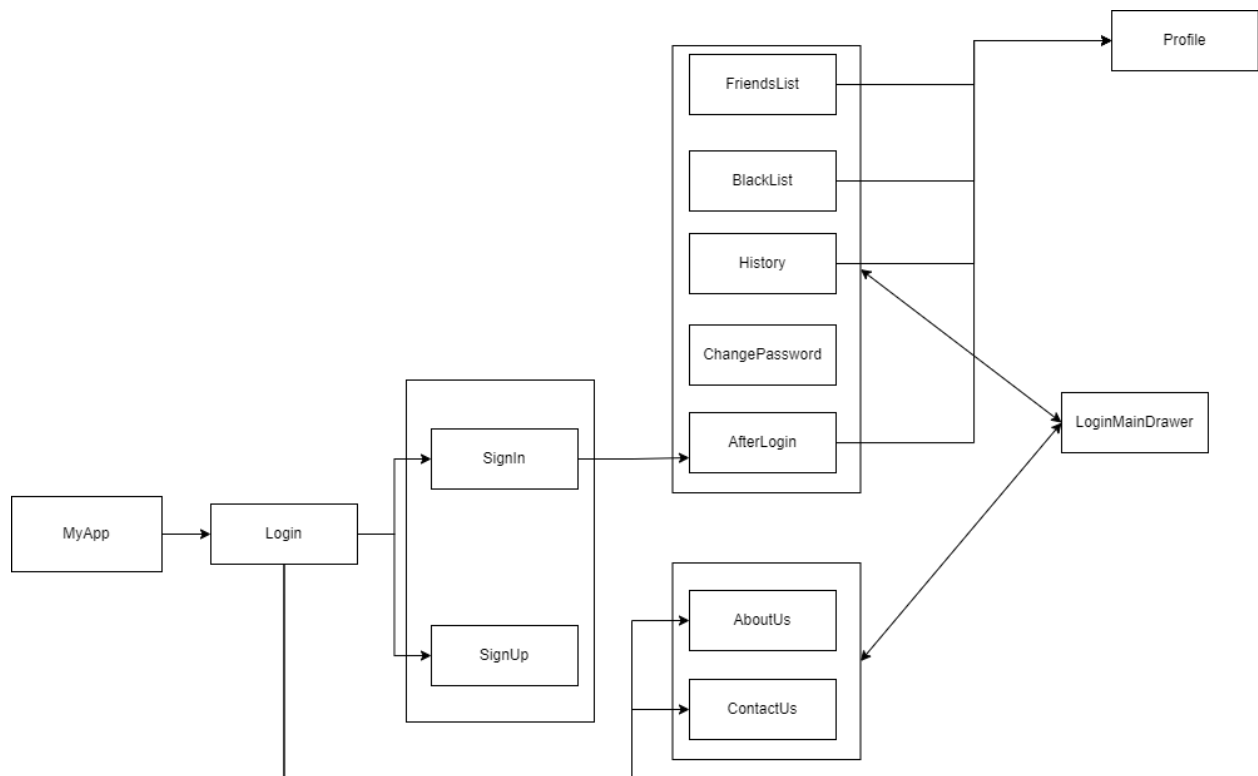
Para além do cumprimento de requisitos, a aplicação mostra-se de interação fácil, fornecendo feedback depois de utilizar cada botão, muito dele através de mensagens de texto, sugestões para correção do mau funcionamento (ex.: Sugerir ativar o *Bluetooth* do dispositivo após detetar que este se encontra desativo) e ações rápidas depois de pressionar elementos durante um intervalo de tempo.

Arquitetura

Widget Tree

A Widget Tree é iniciada por uma página inicial (*Home*) onde é possível efetuar o registo de conta (sujeito a confirmação por email) ou login, caso a conta já exista. A partir desta apenas são possíveis essas ações e visitar algumas páginas com informação do grupo , a partir da *drawer* lateral.

A partir do login realizado, a situação é semelhante. Na página inicial realiza-se o scan de dispositivos (que, por sua vez, levam a uma página individual com detalhes de cada um), ou consulta-se outras páginas a partir da *drawer* lateral também (*Friends List*, *Black List*, *History*, *Contact Us*, *About us*, *Change Password*).



Decoupled Scenario

Embora a aplicação deva correr normalmente em ambiente iOS, não garantimos o correto cumprimento de todas as funcionalidades devido a problemas de acesso a permissões (dados biométricos, localização, etc.).

Quanto à utilização da aplicação sem acesso à internet, esta não é aconselhada pois o processo de autenticação depende de conexão à internet. Contudo, após este processo, é possível a utilização sem conexão, visto que a sessão permanece guardada menos após o fecho da aplicação, e as *features* da aplicação são possíveis de realizar sem ligação à internet, apenas com o Bluetooth ligado.

Gestão de Dados

A gestão de dados é feita sobre 2 componentes: Firebase e Hive. A primeira serve unicamente para suportar autenticação de utilizadores e obter dados sobre novas

contas registadas e acessos. Através de Hive (utilizando a biblioteca *hive* para Flutter) são armazenados todos os dados úteis para um utilizador da aplicação, maioritariamente provenientes da análise de dispositivos em seu redor. São assim guardados objetos de uma classe *Contact*, contendo o nome do dispositivo, o endereço MAC, a intensidade do sinal, o carácter de pertencente a alguma lista (*Friend* ou *Black*) e fotografias que o utilizador da app pode associar a esse perfil. Também o histórico é assegurado no *Hive*, visto que associamos a cada *Contact* a data em que foi descoberto.

Fontes de dados

A *StalkTracker* não utiliza fontes de dados externas para consumo do utilizador. Contudo, utiliza *Firebase* como serviço de autenticação e, assim, correto funcionamento da aplicação.

No que toca a sensores, para cumprimento dos objetivos planeados, a aplicação utiliza:

- Bluetooth: de forma a obter os dados dos dispositivos em seu redor
- Localização: de forma a ajudar a obter a proximidade dos dispositivos
- Câmara: com o objetivo de guardar fotos associadas ao perfil de contactos
- Leitor de dados biométricos (Impressão Digital): com o intuito de limitar o acesso ao histórico apenas ao real utilizador do dispositivo com a *StalkTracker*

Para além destas virtudes, também é possível gerar Códigos QR a partir da página *Contact Us*, fornecendo mais informações acerca dos elementos do grupo do projeto.

Resultados Gerais

Objetivos e Problemas

Após testar a aplicação várias vezes, com diferentes utilizadores e em diferentes ambientes, podemos concluir que cumpre os objetivos inicialmente planeados.

Contudo, achamos que era possível desenvolver mais alguns setores:

- O *Hive* mostrou-se bastante útil e intuitivo, porém achamos bastante difícil de guardar dados complexos como imagens ou vídeos, tal como fazê-lo online.
- Também a biblioteca *bluetooth serial* forçou a refletir acerca de um trade-off produtividade/eficiência, visto ser das poucas bibliotecas que detetam todo o tipo de dispositivos *Bluetooth* e não só dispositivos *LE Bluetooth* (Gadgets do dia-a-dia como PCs e Smartphones não são *LE*), porém, despende bastante mais bateria.
- Seria também interessante relacionar os perfis encontrados pela StalkTracker com as ligações WiFi ou outras ligações de comunicação móvel, de forma a poder obter ainda mais informações sobre ainda mais dispositivos que rodeiam o utilizador

Contribuições

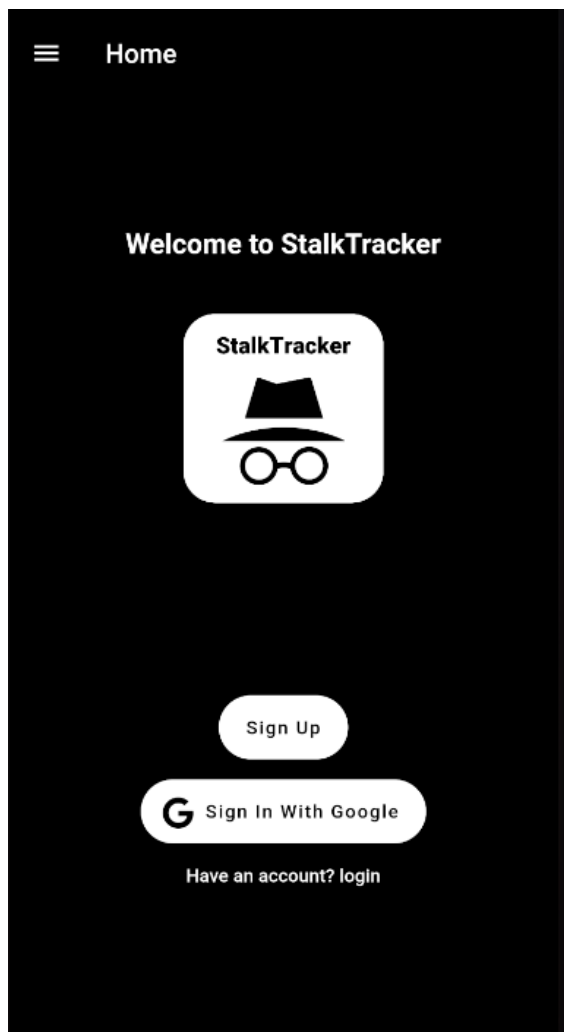
No que toca a contribuições, consideramos que ambos os membros do grupo desenvolveram 50% da aplicação, mais especificamente da seguinte forma:

- ☐ **Dário Matos** - Implementação de mecanismos de Persistence (*Hive*), da biblioteca capaz de ligações *Bluetooth (bluetooth serial)*, implementação da capacidade de utilização da câmara e do botão de emergência

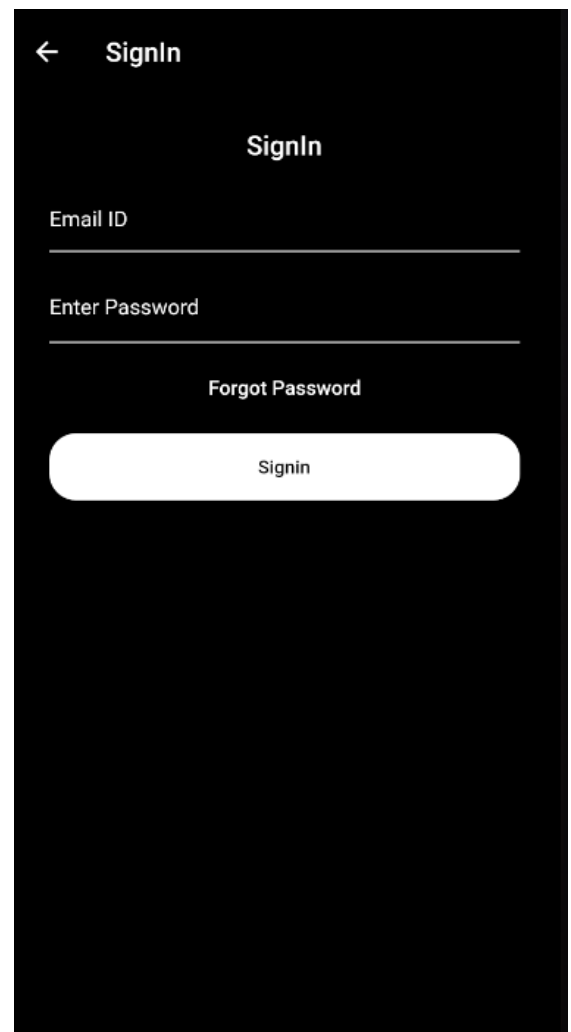
- **Pedro Almeida** - Implementação do uso de sensores biométricos, implementação da geração de QR Codes, desenvolvimento de UX e apresentação de dados dinâmicos

Ambos os membros trabalharam no design dos widgets, nas mecânicas de passagem de informação entre páginas, nos alertas e feedback da UI, tal como no desenvolvimento do relatório.

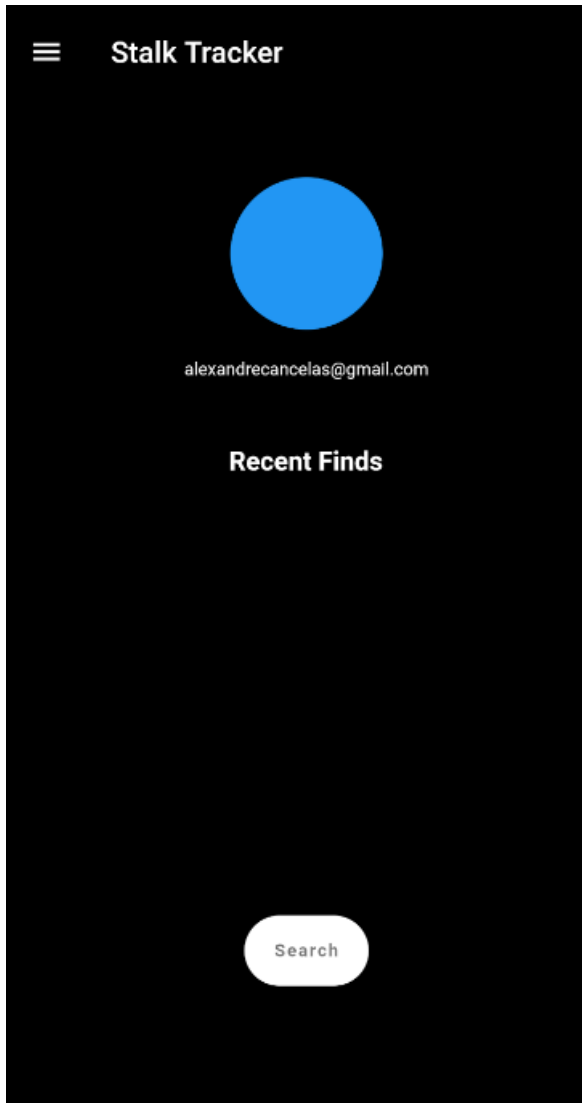
Manual de utilização (caso exemplo)



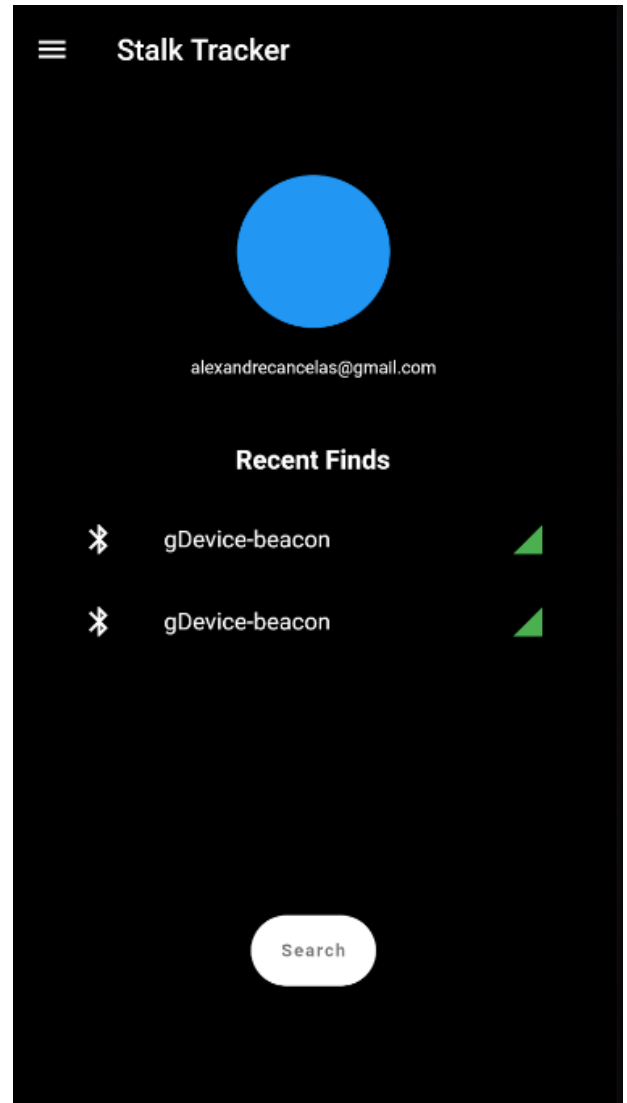
1 - Ecrã principal



2 - Página de Login, após precionar "Signin"



3 - Página inicial de
conta pessoal



4 - Após pressionar "Search",
o display dos dispositivos
encontrados