

Usuário do GitHub: denisrosas

Health Wizard

Descrição

Aplicativo que visa facilitar o agendamento de uma consulta médica levando em consideração a distância, próximo horário livre do médico e a avaliação feita pelos pacientes. A aplicação será feita somente para o usuário que quer agendar a consulta. Não permitirá aos médicos verem as consultas agendadas.

Uma pessoa que precisa realizar uma consulta com algum especialista, baixa o aplicativo e diz qual especialista ele quer marcar (ortopedista, cardiologista, etc), seleciona o dia e hora desejados.

O aplicativo irá mostrar todos os médicos daquela especialidade, e permitirá ao usuário agendar a consulta pelo próprio aplicativo.

Público-Alvo

Qualquer pessoa que necessite marcar uma consulta médica para si ou para um familiar, amigo. E quer facilmente filtrar por disponibilidade ou distancia ou mesmo saber um pouco a opinião de outros pacientes sobre um médico.

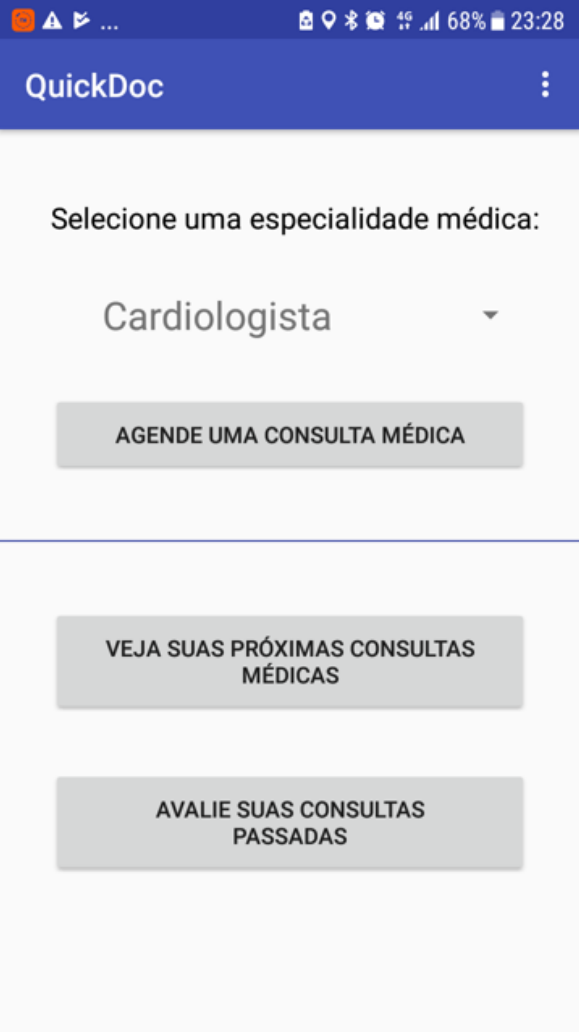
Funcionalidades

Principais funcionalidades do app:


- Exibe uma lista de médicos de uma determinada especialidade procurada
- Facilita a escolha do médico por suas avaliações, por disponibilidade ou distancia
- Faz o agendamento de consulta médica
- Permite ao paciente rever suas próximas consultas ou cancelar
- Permite aos pacientes avaliar as suas consultas médicas e publicar no app

Protótipo de Interfaces do Usuário


Tela 1

	<p>A primeira tela é onde o usuário escolhe o que ele quer fazer.</p> <ul style="list-style-type: none">- Agendar uma consulta médica- Ver próximas consultas agendadas- Avaliar o médico das consultas passadas
--	--

Tela 2



The screenshot shows the QuickDoc mobile app interface. At the top, there is a blue header with the app name "QuickDoc" and a menu icon. Below the header, the text "Cardiologista" is displayed. The main content area lists four cardiologists, each with a cartoon avatar, name, rating (4.4), distance (2.2 Km), and waiting time in days.

Avatar	Nome	Avaliação	Distância	Tempo de Espera
	Antonio Almeida	4.4	2.2 Km	55 dias
	Antonio Almeida	4.4	2.2 Km	47 dias
	Patrícia Cordeiro	4.4	2.2 Km	7 dias
	Antonio Almeida	4.4	2.2 Km	10 dias

Nesta tela o usuário vê todos os médicos da especialidade que ela procura (restrito a um raio de 25 quilômetros). No menu no lado direito o usuário pode escolher o tipo de ordenação:

- Mais próximos primeiro
- Mais bem avaliados primeiro
- Menos dias de espera para próximo horário livre

Tela 3



Se o usuário selecionar um médico, abrirá esta tela com todos mais detalhes.

- Foto maior
- Nome e especialidade
- Avaliação média geral
- Uma breve apresentação do médico
- Os planos que ele aceita
- Qual dia teria o próximo horário livre

E um botão “Agendar consulta com médico”

Tela 4

The screenshot shows the QuickDoc mobile app interface. At the top, there's a status bar with icons for signal, battery, and time (23:30). Below it, a blue header bar displays the app name "QuickDoc". The main content area features a large number "10" with left and right arrow icons, indicating the selected day. Below the number, the month "Maio" is displayed. A section titled "Selecione um dos horários disponíveis" (Select one of the available times) lists 16 time slots from 8:00 to 16:30, each with a radio button. The 10:00 slot is selected. At the bottom, there is a button labeled "AGENDAR CONSULTA COM MÉDICO". A modal dialog box is open at the bottom, asking for confirmation: "Confirmar consulta com Dr(a). Antonio Almeida - Cardiologista". It has two buttons: "NÃO" (No) and "SIM" (Yes).

QuickDoc

← 10 →

Maio

Selecione um dos horários disponíveis

- ☐ 8:00
- ☐ 8:30
- ☐ 9:00
- ☐ 9:30
- ☐ 10:00
- ☐ 10:30
- ☐ 11:00
- ☐ 11:30
- ☐ 13:00
- ☐ 13:30
- ☐ 14:00
- ☐ 14:30
- ☐ 15:00
- ☐ 15:30
- ☐ 16:00
- ☐ 16:30

AGENDAR CONSULTA COM MÉDICO

Confirmar consulta com Dr(a).
Antonio Almeida - Cardiologista

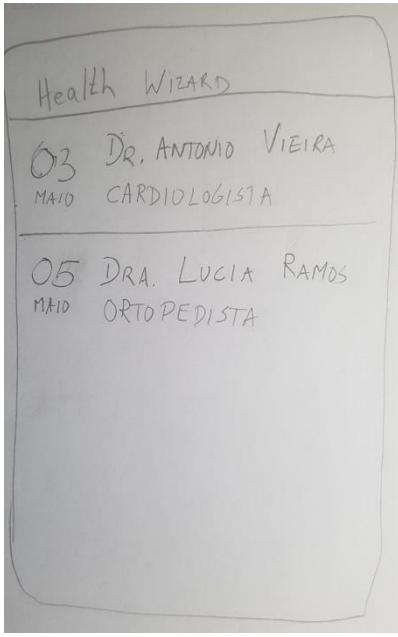
NÃO SIM

Nesta tela o usuário irá selecionar o dia e o horário de sua consulta. O primeiro dia com pelo menos um horário disponível será apresentado ao usuário. Tocando nas setas para esquerda e direita ele irá para o dia anterior ou próximo para visualizar os horários livres.

Abaixo da data aparecerão somente os horários disponíveis. O primeiro horário em aberto já estará selecionado.

Ao clicar em agendar um pop-up de confirmação será exibido para o usuário confirmar o agendamento.

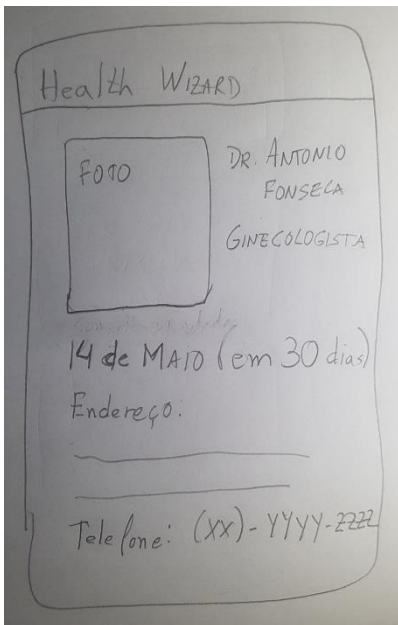
Tela 5



Esta tela é exibida quando o usuário na tela 1 pressiona o botão “Ver Suas Próximas Consultas”.

Nela são exibidas todas as próximas consultas agendadas pelo usuário, mostrando a data, o médico e a especialidade.

Tela 6

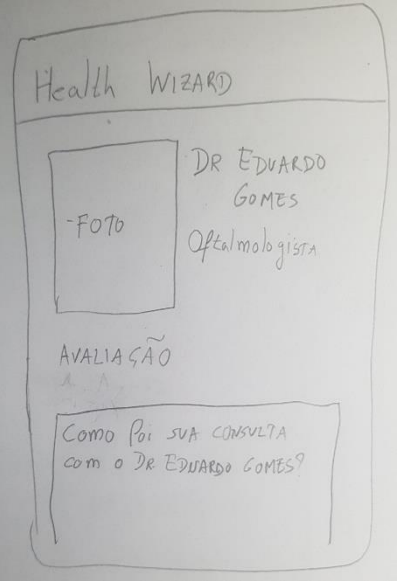


Esta tela é exibida quando o usuário na tela 5 seleciona uma das suas consultas agendadas.


Esta tela exibe os detalhes do agendamento da consulta médica exibindo:

- Foto
- Nome e especialidade do médico
- Endereço
- Data e hora da consulta
- Telefone

Tela 7

	<p>Esta é a tela para avaliação de uma consulta que já aconteceu, onde o usuário pode detalhar como foi sua experiência com o médico e publicar para os demais.</p>
---	---

Widget

	<p>O Widget terá as 3 próximas consultas médicas. Caso o usuário selecione uma das consultas, será aberta a tela 6 com os detalhes da consulta.</p>
--	---

Considerações Chave/Key Considerations

Como seu app vai tratar a persistência de dados?

Os dados dos médicos (foto, especialidade, planos que atende, endereço, telefone de contato), as consultas médicas, as avaliações dos médicos serão salvas em uma estrutura JSON do Firebase.

Como eu já criei uma base de dados no Firebase, eu anexeí ela no Github como exemplo. **Por favor olhe o arquivo quick json example.json.**

Descreva qualquer caso de uso específico (“corner case”) da experiência do Usuário (UX).

Fluxo de tela para agendamento de consulta. Quando o usuário confirmar o agendamento ele retornará para Tela 1.

Tela 1 → Tela 2 → Tela 3 → Tela 4 → Tela 1

Fluxo de tela para ver próximas consultas agendadas. Após chegar na tela de agendamento de consulta, só é possível voltar.

Tela 1 → Tela 5 → Tela 6

Fluxo de tela para avaliação de consulta. Ao concluir a avaliação, o usuário retorna a Tela 1.

Tela 1 → Tela 7 → Tela 1

Descreva quais bibliotecas você utilizará e compartilhe a razão de incluí-las.

Farei uso da biblioteca Picasso para exibir a imagem das fotos dos médicos nos ImageView. A biblioteca Picasso facilita o uso de imagens que não estão locais no dispositivo, fazendo cache das mesmas e tirando a responsabilidade do desenvolvedor de gerenciar a conexão e download.

Descreva como você implementará o Google Play Services.

Farei uso do service “Google Location and Activity Recognition” para obter a localização do usuário e medir a distância atual entre o usuário e o médico. Não pretendo usar mais nenhum service.

Próximos Passos: Tarefas Necessárias

Tarefa 1: Criação do projeto no Firebase e definição de regras de acesso

1. Criar projeto Firebase
2. Definir permissões para que cada usuário tenha acesso somente suas próprias consultas.

Tarefa 2: Implementar todas as interfaces

1. Definir o código XML de todas as interfaces XML
2. A tela 2, 4 e 5 (consulta de médicos, lista de horários disponíveis, e lista de consultas agendadas) contém listas que serão RecyclerViews. Será necessário desenhar os itens de cada RecyclerView.

Tarefa 3: Implementar autenticação pela conta Google

1. Ao iniciar o app na Main Activity, verificar se o usuário está autenticado ao Firebase
2. Se não estiver, pedir para se autenticar usando a conta Google.

Tarefa 4: Definição de toda a árvore JSON

A definição da árvore JSON é fundamental para minimizar o fluxo de dados na obtenção de informações do Banco de dados.

1. Definir como serão armazenados os dados dos médicos
2. Definir como serão armazenados os dados dos usuarios
3. Definir como serão armazenadas as consultas médicas
4. Definir como serão armazenadas as avaliações dos médicos

Lembrando que eu precisarei filtrar as consultas médicas tanto pelo paciente quanto pelo médico. Uma vez definida a estrutura, irei preencher com informações (falsas) de médicos para poder usar simular o uso do aplicativo de casos reais.

Tarefa 5: Definição das classes para armazenar o conteúdo JSON

Será necessário criar classes para ler e escrever os dados direto no JSON.

- Classe com os dados do médico
- Classe com os dados de uma consulta médica.
- Classe com os dados de uma avaliação de um médico

Tarefa 6: Implementação da MainActivity

A MainActivity irá

1. Verificar que o usuário está autenticado
2. Conectar com o projeto Firebase
3. Implementar o funcionamento dos botões
 - a. Agendar consulta
 - b. Ver próximas consultas
 - c. Avaliar médico de consultas passadas

Tarefa 7: Implementação da Activity que lista médicos de uma especialidade

1. Verificar que o usuário está autenticado
2. Obter dados dos médicos de uma determinada especialidade
3. Preencher recyclerview com foto, nome, avaliação, distância e tempo até próxima consulta do médico
4. Implementar funcionamento do botão (verificar agenda do médico)

Tarefa 8: Implementação da Activity de detalhes do médico escolhido

1. Verificar que o usuário está autenticado
2. Implementar exibição de Foto que será obtida do Firebase Storage
3. Obter detalhes do médico do firebase
4. Alterar os textviews com as informações obtidas do médico:
 - a. Nome do médico
 - b. Especialidade
 - c. Avaliação média
 - d. Planos de saúde que ele aceita
 - e. Texto com breve apresentação do médico
 - f. Telefone para contato
 - g. Endereço
 - h. Tempo em dias até próximo horário disponível
5. Implementar funcionamento do botão “realizar agendamento”

Tarefa 9: Implementação da Activity de seleção de Data e horário da consulta

1. Verificar que o usuário está autenticado
2. Implementar Funcionamento das setas para exibir o dia anterior ou posterior
3. Exibir lista de horários disponíveis no dia em uma RecyclerView
4. Implementar pop-up de confirmação para agendamento com o nome do médico, especialidade, dia e horário.
5. Armazenar corretamente no Firebase a consulta médica.

Tarefa 10: Implementação da Activity “Próximas Consultas Agendadas”

1. Verificar que usuário está autenticado
2. Pegar da base de dados, todas as consultas para o usuário logado
3. Implementar RecyclerView que exibe data, hora, nome do médico e a especialidade
4. Implementar que ao tocar em um item da RecyclerView, chama a Activity para detalhes da consulta (Tela 6)

Tarefa 11: Implementação da Activity que contém os detalhes da consulta médica

1. Verificar que o usuário está autenticado
2. Pegar dados do Firebase da Consulta Médica
3. Colocar a foto do médico no ImageView a esquerda
4. Preencher todos os textviews com os dados da consulta

Tarefa 12: Implementação da Activity para listar as consultas médicas que já passaram e não foram avaliadas

1. Verificar que o usuário está autenticado
2. Obter dados de todas as consultas médicas feitas pelo usuário logado, que o horário da consulta já passou e que não foram avaliadas ainda
3. Preencher RecyclerView com estes dados

4. Implementar a chamada de outra activity quando o usuário selecionar uma das consultas passadas

Tarefa 13: Implementação da Activity para avaliar consulta médica

1. Verificar que o usuário está autenticado
2. Exibir foto, nome e especialidade do médico
3. Exibir EditText em bom tamanho para que usuário escreva sua avaliação.
4. Implementar funcionamento do botão “Avaliar Consulta Médica”

Tarefa 14: Testar o aplicativo em diferentes dispositivos

1. Fazer diversos testes com o aplicativo para confirmar seu correto funcionamento com três ou mais usuários.