

Projeto e Seminário - Ano Letivo 2019/2020

Proposta de Projeto

Daniel dos Santos Cabral – 40569 – LI61N

40569@alunos.isel.ipl.pt – 91 204 62 84

Orientador: Eng. Carlos Martins

cmartins@cc.isel.ipl.pt

Título

Projeto IQueue

Introdução

Este projeto consiste na implementação de um sistema de gestão de senhas virtuais para atendimento em diversos serviços.

Envolve o desenvolvimento de três 3 componentes fundamentais:

- API REST Servidora
- Aplicação Android
- Aplicação Web

Aplicação Android

- Terá um sistema de registo e de autenticação. Após registo, o utilizador poderá usar a aplicação para tirar senhas virtuais para os serviços dos operadores aderentes.
- Após *login*, quando entra no raio de alcance de um transmissor *beacon* pertencente a um operador aderente, abre *pop-up* perguntando se pretende “tirar senha”.
- Ao confirmar, é desencadeado um pedido HTTPS ao servidor.
- Se responder “Não”, o *pop-up* só volta a abrir quando sair e voltar a entrar no raio de alcance do *beacon*.
- Após tirar a senha:
 - Fica disponível a informação atualizada da senha que está a ser atendida no momento.
 - Tem como opção desistir da espera.
- Quando sai do alcance do transmissor *beacon*, é aberto *pop-up* a questionar se pretende desistir ou continuar à espera.
- Quando faltarem N senhas para ser atendido, recebe uma notificação.
- Quando chegar a sua vez, o utilizador recebe uma notificação (com o balcão onde será atendido, caso tenha sido configurado um). A notificação pode ser repetida até ser considerado desistência e nesse caso, recebe a notificação de desistência.
- Depois de ser atendido, o utilizador recebe um pedido para classificar o atendimento e, opcionalmente, para dar sugestões sobre o mesmo.
- Para tirar senha fora do raio de alcance do *beacon*, o utilizador:
 - Após *login*, tem uma opção para visualizar os operadores aderentes assim como os serviços que disponibilizam com a opção de tirar senha antecipadamente.
 - Quando entra no raio de alcance de um *beacon* pertencente a esse operador, já não será despoletado o *pop-up* para tirar senha.
 - Pode ver quantas senhas faltam para ser atendido mesmo antes de entrar no raio de alcance do *beacon*.
 - Recebe notificações quando faltarem N senhas para ser atendido.
 - Se ainda não estiver no raio de alcance do *beacon* quando chegar a sua vez, é despoletado um *pop-up* a questionar se pretende tirar nova senha ou desistir.

REST API e Aplicação WEB

- Deve suportar a configuração de vários operadores, cada um com filas de espera personalizadas.
- Deve suportar três perfis de utilizadores:
 - Um perfil de administrador *master*, que pode configurar os vários operadores, definir utilizadores com perfil de gestão para cada operador e aceder ao *log* de erros do sistema. Quando é configurado um utilizador de gestão para um operador, este recebe um *email* com a respectiva confirmação e credenciais de acesso.
 - Um perfil para utilizadores de atendimento, que só pode “chamar” as senhas, marcar o início e fim do atendimento assim como informar da desistência.
 - Um perfil para gestão do sistema, que pode:
 - Definir várias filas de espera para vários serviços, podendo definir um limite de senhas por dia para o serviço.
 - Adicionar/remover utilizadores com perfil de atendimento. Quando é configurado um utilizador de atendimento, recebe um email com a respectiva confirmação e credenciais de acesso.
 - Obter relatórios de: número de pessoas atendidas; tempo médio de espera; tempo médio de atendimento; número de pessoas que desistiram; classificações dos serviços, e; sugestões feitas pelos utilizadores.
 - Eliminar informação de clientes, visando cumprir o Regulamento Geral de Proteção de Dados Pessoais (RGPD).
- O sistema de *log* terá um automatismo para envio de *email* para o administrador *master* quando ocorrerem erros severos no sistema (e.g. *status code* na gama 500).

Funcionalidades Gerais

- Ambas as *user interfaces* (Android e Web) devem suportar internacionalização, estando disponíveis pelo menos em dois idiomas (Português e Inglês).
- Usar *Transport Layer Security* (TLS) nas comunicações com o servidor.
- *Deployment* do servidor na *cloud* e da aplicação Web com suporte para balanceamento de carga.

Tecnologias a Utilizar

A aplicação Android será implementada utilizando a linguagem *Java* e tirando partido das APIs *Nearby* e *Proximity Beacon* da *Google*.

O servidor será igualmente implementado em *Java*, utilizando os *frameworks Spring* e *Hibernate*.

Será utilizada a base de dados relacional *SQL Server*.

A aplicação Web será implementada utilizando o *framework Angular*.

Serão adquiridos pelo menos três transmissores *beacon*, para permitir um cenário de desenvolvimento e de teste o mais realista possível.

Etapas intermédias

Versão *beta* da REST API – 5 de abril de 2020

Versão *beta* da aplicação Web – 3 de maio de 2020

Versão *beta* da aplicação Android – 31 de maio de 2020

Planeamento

Actividades	Semana 1 09/3 - 15/3	Semana 2 16/3 - 22/3	Semana 3 23/3 - 29/3	Semana 4 30/3 - 05/4	Semana 5 06/4 - 12/4	Semana 6 13/4 - 19/4	Semana 7 20/4 - 26/4	Semana 8 27/4 - 03/5	Semana 9 04/5 - 10/5	Semana 10 11/5 - 17/5	Semana 11 18/5 - 24/5	Semana 12 25/5 - 31/5	Semana 13 01/6 - 07/6	Semana 14 08/6 - 14/6	Semana 15 15/6 - 21/6	Semana 16 22/6 - 28/6	Semana 17 29/6 - 05/7	Semana 18 06/7 - 12/7	Semana 19 13/7 - 19/7	Semana 20 20/7 - 25/7
Elaboração da proposta do projeto																				
Definição do modelo de dados e implementação da Base de Dados																				
Implementação da camada de acesso a dados e <i>logger</i> e definição dos URIs da API																				
Implementação dos <i>Controllers</i> e do serviço de envio de e-mails e definição das respostas JSON																				
Definição das páginas a implementar e dos seus componentes																				
Implementação das páginas e dos seus componentes																				
Definição das <i>Activities</i> e componentes a implementar																				
Implementação das <i>Activities</i> e outros componentes																				
Implementação do serviço de reconhecimento de Beacons																				
Implementação do sistema de autenticação - API																				
Implementação do sistema de autenticação - Web																				
Implementação do sistema de autenticação - Android																				
Optimização - API																				
Optimização - Web																				
Optimização - Android																				
Revisão Global Final																				
Elaboração do Relatório																				
Legenda	Plano	API	Web	Android	Relatório	Geral														

Objetivos

- Aprimorar o desenvolvimento nas tecnologias envolvidas.
- Desenvolver competências na implementação de *software* em comunicação com transmissores *beacon* e adquirir prática na utilização das APIs Nearby e Proximity Beacon da Google.
- Desenvolver competências na implementação de APIs seguras através da configuração do protocolo TLS.