

FundStarter

Sistemas Distribuídos 2015/2016

Entrega: 19 de Dezembro de 2015, 20:00

1 Objectivos do projecto

No final do projecto, o aluno deverá ter:

- Desenvolvido uma interface web para o FundStarter.
- Ter integrado a interface web com o projecto desenvolvido na meta 1.
- Dominado Struts2, JSPs, JavaBeans e Servlets.
- Seguido a arquitectura MVC para desenvolvimento web.
- Aplicado WebSockets para comunicar assincronamente com os clientes.
- Integrado a aplicação com um serviço REST externo.

2 Overview

Nesta segunda meta do projecto, irão criar um frontend web para a vossa aplicação. Esta nova interface irá possibilitar aos vossos utilizadores o acesso ao serviço a partir de quase qualquer dispositivo com Internet no planeta, sem necessitar de instalação de software do lado do cliente. Como a interoperabilidade é um requisito muito importante, utilizadores web deverão aceder à mesma informação que os utilizadores na aplicação desktop. Para tal, o servidor web deverá comunicar com o DataServer, anteriormente desenvolvido em RMI.

Os utilizadores deverão ter as mesmas funcionalidades, independentemente da interface que usem. Portanto a interface web deverá mostrar os projectos, recompensas e deverá permitir a criação de projectos e a doação de dinheiro. Como o aspecto interactivo é muito importante na web, o vosso projecto deverá mostrar alterações em tempo real, nomeadamente no dinheiro recolhido, e através de notificações quando é recebido dinheiro. Como os utilizadores estão cada vez mais exigentes, técnicas menos robustas como meta-refresh e iframes ocultas não serão aceites, fazendo-se uso de WebSockets.

Finalmente, as aplicações de hoje em dia não existem isoladas umas das outras. Através de APIs REST e OAuth, irão integrar a vossa aplicação com um serviço externo, o Tumblr. Ao criarem um projecto, deve ser criado um post no

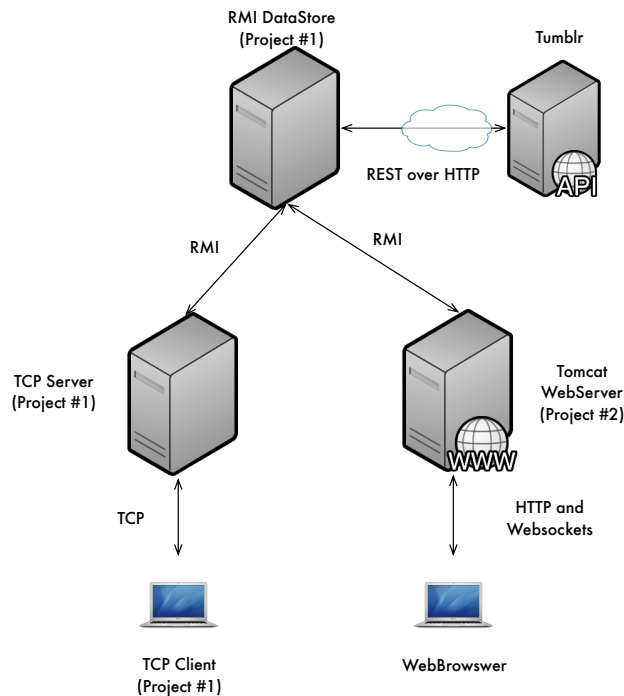


Fig. 1: O servidor web deverá integrar com a DataStore via RMI. A comunicação com o browser deverá ser via HTTP e via WebSockets).

tumblr/blog desse utilizador. Cada utilizador que dê dinheiro faz automaticamente um like nesse post.

3 Arquitectura

A Figura 1 mostra a arquitectura a seguir para o projecto. Devem implementar uma aplicação web que corra num servidor HTTP (Apache Tomcat) e que actue como um cliente RMI para com a DataStore. Os clientes irão usar browsers para se ligarem ao servidor web para pedirem páginas através de HTTP. Para melhorar a experiência de utilização podem ponderar fazer alguns dos pedidos via AJAX em vez de carregar a página toda.

A comunicação em tempo real para o browser deverá ser implementada usando WebSockets. Isto inclui notificações e alterações de valores em tempo real.

4 Interface Web

Usando uma arquitectura MVC, deverão implementar os seguintes requisitos funcionais usando Struts:

1. **Listar projectos actuais, com o seu progresso** - Deve ser possível de verificar para cada projecto se está perto do seu valor objectivo, ou não.
2. **Listar projectos antigos** - Deve ser possível consultar uma lista de projectos cuja data-final já terminou, vendo se foram aceites ou não chegaram ao objectivo.
3. **Consultar detalhes de um projecto** - Os detalhes incluem a data limite para investimento, os possíveis rewards, uma descrição do projecto e níveis extra.
4. **Registar conta** - Deve ser possível criar uma nova conta para um novo utilizador. A cada conta, deverá ser atribuído um saldo inicial de 100\$.
5. **Login** - Deverá ser possível fazer login, passando a ter acesso às seguintes funcionalidades.
6. **Consultar saldo** - Deverá ser possível a utilizadores autenticados consultar o crédito disponível na plataforma.
7. **Consultar recompensas** - Deverá ser possível a utilizadores autenticados consultar as recompensas a que têm direito, de doações feitas a projectos em curso e terminados com sucesso.
8. **Doar dinheiro ao projecto (Pledge)** - Os utilizadores poderão doar dinheiro correspondente a uma das recompensas, escolhendo uma das alternativas do projecto ao mesmo tempo. O dinheiro é retirado da conta do utilizador e fica retido pela plataforma. O voto na alternativa do projecto deverá ficar registado. (Exemplo: o utilizador João doa 60\$ à FitJump, escolhe a opção vermelho e fica com a recompensa de uma unidade.)
9. **Enviar mensagens para o projecto** - Os utilizadores registados poderão enviar mensagens para um projecto, de forma a esclarecer eventuais dúvidas.
10. **Criar um projecto** - Cada utilizador pode começar um ou vários projectos. Um projecto tem um nome, descrição, data limite, valor pedido e recompensas (cada recompensa tem um valor e descrição).
11. **Adicionar e Remover recompensas ao projecto** - O utilizador responsável pelo projecto poderá remover e adicionar as recompensas associadas ao projecto.
12. **Cancelar projecto** - O utilizador responsável pelo projecto poderá cancelá-lo, impossibilitando outros utilizadores de o visualizar. O dinheiro retido das recompensas pela plataforma deverá ser devolvido aos apoiantes.

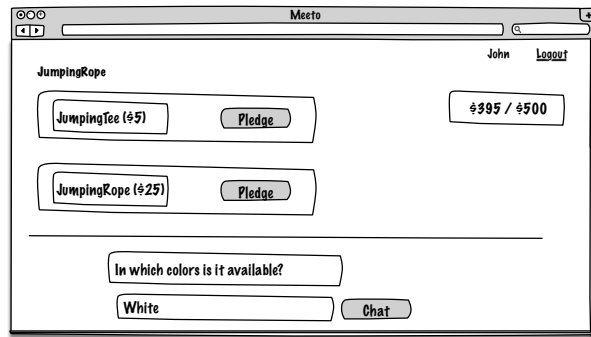


Fig. 2: Exemplo de uma interface possível

13. **Responder a mensagens de apoiantes** - As mensagens enviadas pelos apoiantes merecem resposta, e cabe apenas ao responsável do projecto responder-lhes.
14. **Fim do projecto** - Quando se chega à data-hora limite de um projecto, é verificado se o conjunto do dinheiro doado chega ao limite pedido. Se não chegar, o dinheiro é restituído aos apoiantes e o projecto automaticamente cancelado. Se o projecto tiver recolhido dinheiro suficiente, este é enviado para a conta do responsável pelo projecto. Os apoiantes devem também receber indicação das recompensas a que têm direito.

A Figura 2 mostra um exemplo da interface possível para o projecto.

Grupos de 3 elementos deverão implementar os seguintes requisitos:

15. Os projectos podem ter vários administradores que podem alterar e cancelar o projecto, bem como responder a mensagens de utilizadores.
16. As recompensas poderão ser atribuídas a um segundo utilizador, como oferta.
17. Podem ser adicionados e removidos níveis de recompensas extra, caso o projecto ultrapasse certos valores altos.

5 Notificações em Tempo Real

De forma a que a vossa aplicação seja responsiva, deverão usar WebSockets para fazer push de informação para o cliente assim que possível. Deverão usar WebSockets para:

18. Actualizar automaticamente os valores recolhidos pelo projecto na lista de projectos e na página de projecto.

19. Notificação para os administradores dos projectos online quando é feita uma doação.
20. Notificação para os administradores dos projectos online sempre que for enviada uma mensagem para o projecto.

6 Integração com o Tumblr

O Tumblr é uma rede social multifacetada. Pode ser usada como um blog (como o Blogger), uma plataforma de microblogs (como o Twitter), como plataforma de fotografia (como o Instagram) ou como uma mistura entre todas.

Neste projecto irão integrar com o Tumblr de forma a que a actividade no FundStarter seja reflectida no Tumblr. Para garantir que se usam as contas dos utilizadores correctamente, a API do Tumblr está protegida por OAuth, um standard para autorização de serviços a aplicações externas que alterem dados dos utilizadores. Para a negociação OAuth, poderão usar a biblioteca Scribe. Para a execução de pedidos REST não poderão ser usadas bibliotecas que comuniquem com a API do Tumblr.

Para este projecto deverão:

20. Associar a conta de um utilizador a uma conta do Tumblr.
21. Quando um projecto é criado, deve ser feito um post no tumblr do utilizador.
22. Sempre que um utilizador com tumblr associado faz uma doação, deve ser feito automaticamente um like no post do projecto.
23. Um utilizador no Tumblr deve poder fazer login no FundStarter com a conta do Tumblr.

7 Relatório

O relatório deverá conter informação sobre as seguintes secções

- Introdução
- Arquitectura (Detalhes de implementação)
- Integração com o Projecto #1
- Integração com WebSockets
- Integração REST
- Especificação de Testes

8 O que irão aprender

Este projecto presuppõe que os alunos aprendam competências práticas relativas a:

- Programar para a Web em Java
- Arquitectura MVC com Struts2
- Integração de Servidor web com RMI.
- Uso de WebSockets
- Utilizar APIs REST
- Autenticar usando OAuth

9 Entrega do projecto

O projecto deverá ser entregue num ficheiro ZIP. Esse ficheiro deverá conter um ficheiro README com toda a informação necessária para instalar e executar o projecto sem a presença dos alunos. Projectos sem este ficheiro, ou sem informações suficientes **não serão considerados**. Projectos que não executem correctamente também não serão avaliados.

Dentro do ficheiro ZIP deverá também estar um PDF com o relatório. O relatório deve seguir a estrutura fornecida, dado que a avaliação irá incidir sobre cada um dos pontos.

Também no ficheiro ZIP deverá existir o ficheiro WAR com a aplicação Web pronta a executar, bem como os deliverables da meta anterior prontos a correr.

Finalmente, o ficheiro ZIP deverá ter também uma pasta com o código fonte do projecto.

`http://infoforestudante.uc.pt`