Entreprise Application

RSS Feed Aggregator

Koalab koala@epitech.eu

Abstract: The purpose of this project is to develop a server-based RSS feed aggregator, using either Java or .net technologies offered to you.

Contents

I Project **2**
I.1 Concept . . . 2
I.2 Requirements . . . 2

II Server **4**
II.1 Common . . . 4
II.1.1 Server . . . 4
II.1.2 Web . . . 4
II.1.3 Database . . . 4
II.2 .Net . . . 6
II.2.1 Deployment and usage . . . 6
II.2.2 Web client . . . 6
II.2.3 Database . . . 6
II.3 Java . . . 7
II.3.1 Deployment and usage . . . 7
II.3.2 Web client . . . 7
II.3.3 Database . . . 7

III Desktop Application **8**
III.1 Common . . . 8
III.2 .Net . . . 8
III.3 Java . . . 8

IV Mobile client **9**
IV.1 Common . . . 9
IV.2 .Net . . . 9
IV.3 Java . . . 9

V Bibliography **11**
V.1 Common . . . 11
V.1.1 Server - Web client . . . 11
V.2 Java . . . 12
V.2.1 Server - WebClient . . . 12
V.2.2 Desktop Application . . . 12
V.2.3 Mobile Application . . . 12
V.3 .Net . . . 13
V.3.1 Server - Web client . . . 13
V.3.2 Desktop Application . . . 13
V.3.3 Mobile Applications . . . 13

Chapter I

Project

I.1 Concept

Everyone knows and uses RSS feeds today. Be it for world news, blog articles or following the news on your favorites subjects, RSS feeds are the main way to receive news on the internet today.

RSS follows the RSS 2.0 Specification (can be found here).

The purpose of this project is to allow you to create your own suite of aggregator to use on your desktop, on the web on any computer, or on your mobile device.

I strongly recommend getting familiar with existing applications, to familiarize with how a rss aggregator is supposed to look and react, and to find the way you want your application to be

I.2 Requirements

Before you start the project, you have to choose the technology you'll use for the development. Your choices are:

- .net technologies (Windows)
- java

The project is divided in four parts, three clients and the server. All parts must be completed, working and usable for your project to be corrected and considered valid.

Depending on your choice of technology, you will have to use the following tools and technologies :

- If you choose to use .net :
  - .net Framework 4.0 minimum, 4.5 recommended
  - Visual Studio (2012-2013)
  - Azure technology to deploy your solution
  - ASP .net - C# - WPF
  - Windows Phone SDK

- If you choose to use java :
  - Java 7 minimum, 8 recommended
  - AWS technologies to deploy your solution
  - Java JSP - JEE
  - Android SDK

Chapter II

Server

II.1 Common

II.1.1 Server

Your server has to be administered correctly, incorporate this in your designs.

All development `MUST` inscorporate mocking and unit testing.

II.1.2 Web

Your interface must provide users a mean of authentification, as well as registering an account. The interface must be as user-friendly as possible, similar to what can be found elsewhere.

Your application must be rendered properly on every latest version of each popular browser.

You have to make full use of unit testing and mocking mechanisms.

Authentification APIs (like OAuth) is a welcome addition to your work.

```
Any framework or library not direcly authorised is subject to
caution, send us an e-mail asking to confirm it's usage before you
start using it.
```

II.1.3 Database

Users must be able to find their feed in the same state, whatever device they use or where they connect. They must also be able to manage their feeds (addition and removal). Again, use the existing feeds reader to inspire you and see what features are expected.

Your database must be correctly set up. A database with one table or 100 will greatly impact your final score.

Don't hesitate to drop by the lab to discuss your database architecture.

II.2 .Net

II.2.1 Deployment and usage

The server, workers and database will be deployed using the `Azure` technology. As such, make sure your work is scalable and that your Azure is configured correctly to allow you to do so.

You will use C# to develop the server and workers. Your server will have to provide data and administration through a WCF EndPoint. The EndPoint has to centralize information on streams and users.

II.2.2 Web client

All web applications (including the web client view) will be hosted by an Azure server and not the utility web server provided by Visual Studio.

You are required to use `ASP .Net` or `Silverlight` for your client.

II.2.3 Database

The choice of database is totally up to you.

II.3 Java

II.3.1 Deployment and usage

Your server, workers and database will be deployed using the `Amazon Web Services` (`AWS`) technologies. You will have to install correctly a tomcat server on the platform.

Obviously, you have to integrate mocking and unit testing in your project.

II.3.2 Web client

There are no imposed frameworks, however you have to separate clearly the layers of your project. MVC is a good start.

The whole interface must be developped using either J2EE or Ajax.

Your web application will be hosted on the Tomcat server deployed on your AWS.

II.3.3 Database

The choice of database is totally up to you.

Chapter III

Desktop Application

III.1 Common

You will provide users with a windowed application, allowing them to gather informations from the server, to browser their feeds, with minimum connectivity requirements.

Your application performance will be part of your evaluation. Downloading all the content of all the user's feeds is `NOT` a good feature. You have to find a balance.

The UI has to provide information in a consitent way, only displaying relevant summary data, with a way of seeing all available information for every item.

The "Read" state has to be updated automatically for a single item, a group of items and multiple feeds, saved back to the server.

Authentification must be provided at UI startup, with persistency over application restart.

All features you provide on your web client should be replicated on your desktop application, if possible.

Obviously, unit testing must be part of your development.

III.2 .Net

You will use C# and WPF. You must use `MVVM` pattern to build your application.

III.3 Java

You will implement a MVC architecture.

You are free to use an existing framework or use only the `JDK`

Chapter IV

Mobile client

IV.1 Common

All features available on the desktop and web applications should be available on your mobile device, if possible.

Your application must be able to perform without network connectivity, at least for a time.

Use of the MVVM pattern is strongly recommended.

Unit testing must be present on this part of the project as well.

IV.2 .Net

Your application should run on Windows Phone 8 as well as Windows Phone 8.1. To do so, you are free to use the whole Windows Phone SDK, as well as the latest features available with the 4.5 framework.
Your application must be in C#, but you are free to use any frameworks you like. (As long as they are in C#).

You can find frameworks that allow you to develop your application for all three plateforms (IOS, Android and Windows Phone. You are free to use them as well).

IV.3 Java

Your application will be developed using Java and the Android API. Any framework can be used, as long as they use the Android API and you are able to explain and use it alone.

You can find frameworks that allow you to develop your application for all three plateforms (IOS, Android and Windows Phone. You are free to use them as well).

Chapter V

Bibliography

V.1 Common

V.1.1 Server - Web client

OAuth

Selenium testing suite

V.2 Java

V.2.1 Server - WebClient

Installing Tomcat on a ec2 linux
- Grails framework
- Spring Web MVC framework

V.2.2 Desktop Application

Pure MVC framework
- JDK 7 Swing framework

V.2.3 Mobile Application

Android SDK
- Android quickstart
- Spring for Android
- MVVM for Android
- Codename One framework

V.3 .Net

V.3.1 Server - Web client

Creation and deployment of service in Azure
- Azure services
- ASP .net MVC
- NancyFX framework

V.3.2 Desktop Application

WPF apps with MVVM pattern

V.3.3 Mobile Applications

Start with Windows Phone 8 development
- MSDN tutorial to build Windows Phone application
- Xamarin Framework