

STACKOVERFLOW LIKE

Project Report for StackOverflow Like web site

Report

Elysée Jonas

Rabérin Alexandre

Summary

I - Work to be done	2
A – Subject	2
B – Particularities	2
C – Tools Used	2
II – Conception of the Application	3
A – UML Diagram	3
B – Mock-ups	4
III – Work Done	5
A – Architecture	5
Controllers	5
Services	5
B – Work to be done	5
Conclusion	6
Annex	7

I - Work to be done

A – Subject

In the "Rapid Application Development" cursus, we had to develop an application based on the same concept as the website www.StackOverflow.com

The Application should allow a user to ask questions on a forum; these questions will be able to be seen by all the users of the service.

A user will have to identify himself (therefore own an account) in order to ask a question, to answer a question or comment on another user's answer. Consequently, every registered user will have a profile which everyone will be able to see, and which will show information about the user such as the date at which he sign up or a summary of his activities.

A Tag system will be in place in order to organize questions around specific themes.

Questions and their answers will be able to be voted upon in order to show if the question itself is interesting or if the answer solves the problem brought up by the question. A reward system will exist in order to award users having performed certain actions.

B – Particularities

Certain constraints are added to the subject stated previously:

- The application will have to be internationalized, which means it will have to be able to adapt the displayed features' language to the user's system's language.
- The code and its commentaries will have to be in English
- The application will have to be tested, which means unit tests will have to be put in place in order to test the different functionalities implemented.

C – Tools Used

In order to develop this project, certain tools were made available to lighten the workload. In this cursus' scope, the web application development tool used was Grails: grails is based on the use of the programming language Groovy, used in order to code the application. IntelliJ was used as the development environment which supports the Grails framework.

II – Conception of the Application

A – UML Diagram

The first step of development of the project is to isolate the different entities we are going to use throughout the development and the operating phases of our application. For this reason, we have constructed the following UML Diagram which gives a general vision of our projects:

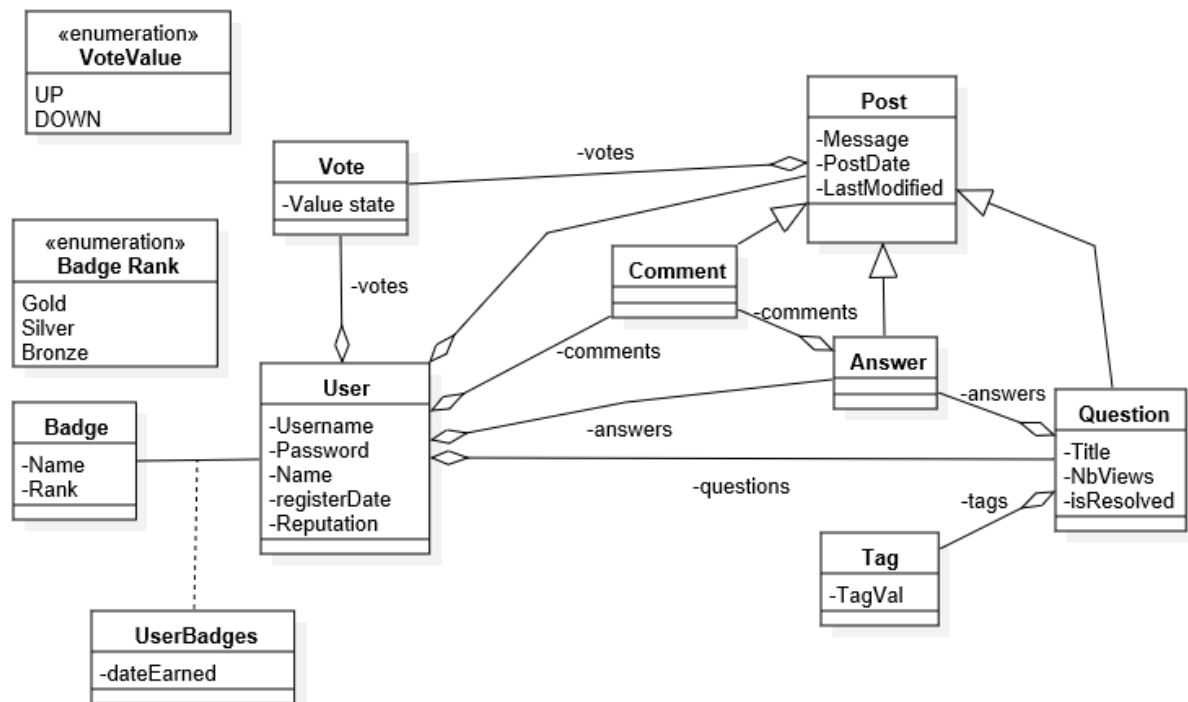


Figure: UML Diagram

We have a domain class which represents a user and all the information we have on him. A user handles posts which can be of various types: Questions, Answers or Comments. Questions can be tagged in order to be organized around similar themes. The community can cast Votes upon Posts in order to show their interest in it, whether it is positive or negative. Finally, a user is awarded badges upon completing certain actions. Every badge has a rank and can be obtained multiple times by a User upon completing specific tasks at different times.

B – Mock-ups

Knowing our application would be a web application, the graphical aspect of our project gains significant importance. The second step to take is clearly to be able to project what the final aspect of the application will be. In that event, we used Balsamiq in order to create different mock-ups of our application which allows to have a clear end goal when developing the graphical side of the application

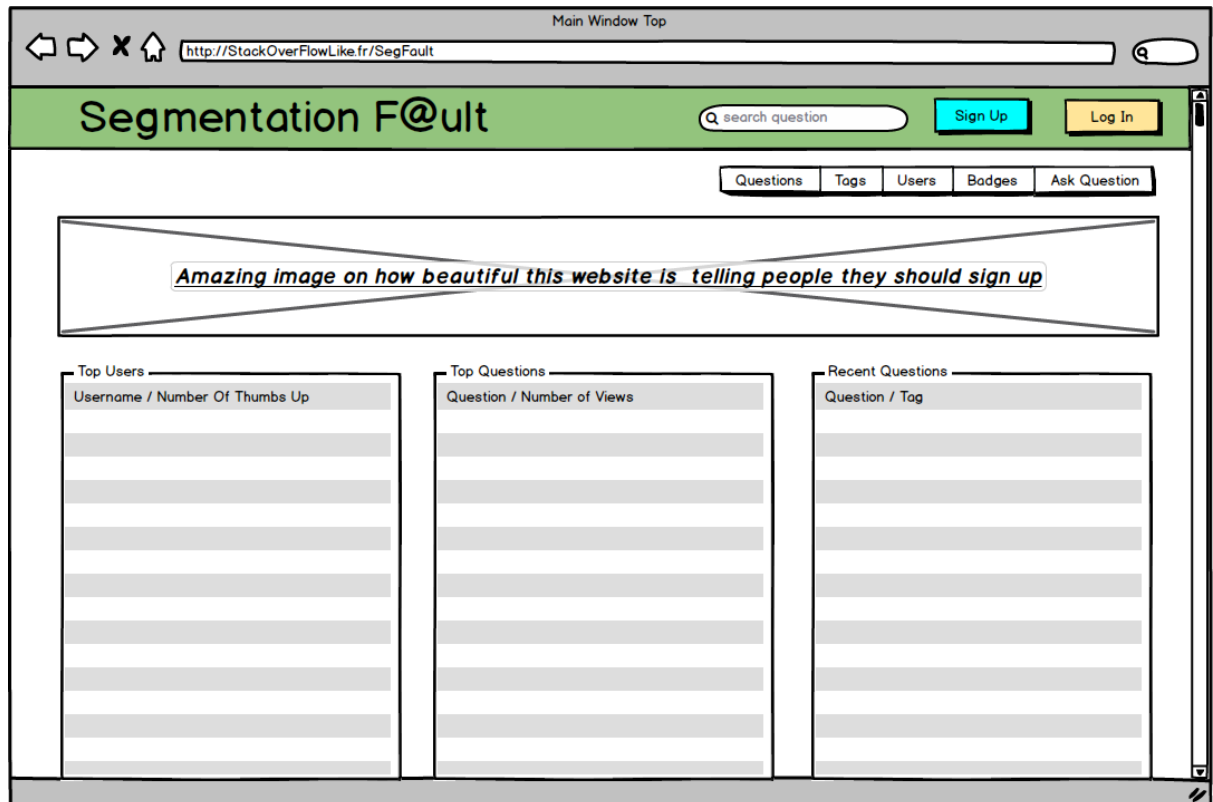


Figure: Home Window Mock-up

III – Work Done

A – Architecture

Different controllers are in charge of different aspects of the application. Each controller, depending on the web site's needs, calls services which in turn are in charge of more complexed procedures such as the logic behind the reputation system. This explains why different services can communicate with one another. In addition, this approach allows us to group the different business logics of our application and delegate the routing logic to the controllers.

The application relies on a privilege system which allows us to create different interfaces for the users depending on the privileges which they have been attributed. Features using this system include the ability to ask a question only if a user is logged in.

A series of test have been put in place in order to test the different domain classes and controllers we have created.

Controllers

Controllers are in charge of four actions on the domain classes to which they are affected (see UML Diagram). These four actions, also known as CRUD actions, allow them to create a new instance of the domain class in the database, read and access an instance of the domain class in the database, update one and, finally, remove an instance from the database.

The controllers are also responsible for the redirection of pages of our website, also referred to as the internal routing system. Depending on various data, such as answers from services or requests from the user, the controller will choose the correct page to display.

Services

Services take on more complex tasks than controllers. In order to separate business treatments and routing operations, services solely know isolated mechanisms such as attributing the correct reputation to a user. The QuestionService, when a user posts a question, will take care of calling other services involved in order to track if a user is rewarded with a badge at the end of the post, for example.

A Service' mechanism is launched by a Controller or another Service.

B – Work to be done

From a functional point of view, our application possesses most of the functionalities awaited for an imitation of the website StackOverflow.

However, the visual aspect of the product we are delivering requires a lot of work in order to be polished. A graphical chart for the website is needed in order for the project to be complete. The current version of the project relies on different bootstrap sources in order to be visually appealing. Nonetheless, it is clear that this solution is unacceptable for a final product but is enough presently given the allotted time.

Conclusion

This project was an opportunity for us to discover new ways to develop web applications thanks to the Grails framework. The complexity of the project covered the necessary aspects needed for us to truly gain the knowledge to be acquired in this cursus.

However, it is clear to us that the time granted in order to fulfill this project was insufficient. Some aspects of the project, mainly the visual aspect, have been heavily affected by this.

Annex

Balsamiq Mockups

A web browser window titled "Sign up page" with the URL "http://SegFault.fr/SignUp". The main content is a light blue box titled "Sign Up" with the heading "Segmentation F@ult". The form includes:

- Your Name:** Two input fields for "firstname" and "lastname".
- Username:** A single input field with a placeholder "Enter your username".
- Your Email:** Two input fields for "Enter your e-mail" and "Confirm your e-mail".
- Your Password:** Two input fields for "Enter your password" and "Confirm password".
- Register:** A single button at the bottom.

Annex: Sign Up Window

A web browser window titled "Sign up page" with the URL "http://SegFault.fr/SignUp". The main content is a light blue box titled "Log In" with the heading "Segmentation F@ult". The form includes:

- Username:** A single input field with a placeholder "Enter your username".
- Password:** A single input field with a placeholder "Enter your password".
- Sign In:** A single button.
- E-Mail:** A single input field with a placeholder "Enter your e-mail".
- Password:** A single input field with a placeholder "Enter your password".
- Sign In:** A single button.

Annex: Log In Window

Sign up page
http://SegFault.fr/SeeQuestion

Segmentation F@ult
search question
François Dubridge
Questions Tags Users Badges Ask Question

Why does it not work?

Asked 12 Dec. 2012 14:12:35 by François Dubridge

3

Lorem ipsum is a pseudo-Latin text used in web design, typography, layout, and printing in place of English to emphasise design elements over content. It's also called placeholder (or filler) text. It's a convenient tool for mock-ups. It helps to outline the visual elements of a document or presentation, eg typography, font, or layout. Lorem ipsum is mostly a part of a Latin text by the classical author and philosopher Cicero. Its words and letters have been changed by addition or removal, so to deliberately render its content nonsensical, it's not genuine, correct, or comprehensible Latin anymore. While lorem ipsum's still resembles classical Latin, it actually has no meaning whatsoever. As Cicero's text doesn't contain the letters K, W, or Z, alien to latin, these, and others are often inserted randomly to mimic the typographic appearance of European languages, as are digraphs not to be

Java Grails Bootstrap
Answer

Answers

Answered 12 Dec. 2012 13:12:35 by Messaiah BlueGiven

42

Just wait until tomorrow

Comment

Commented 13 Dec. 2012 11:11:11 by Mayen Johakin

It works great!! thanks!!!

Answered 12 Dec. 2012 13:12:35 by Dumbut Ignit

-12

First

Comment

Annex: Display Question Details

Sign up page
http://SegFault.fr/PostQuestion

Title Of Your Question:
Why does it not work?

Description of your problem:

Lorem ipsum is a pseudo-Latin text used in web design, typography, layout, and printing in place of English to emphasise design elements over content. It's also called placeholder (or filler) text. It's a convenient tool for mock-ups. It helps to outline the visual elements of a document or presentation, eg typography, font, or layout. Lorem ipsum is mostly a part of a Latin text by the classical author and philosopher Cicero. Its words and letters have been changed by addition or removal, so to deliberately render its content nonsensical, it's not genuine, correct, or comprehensible Latin anymore. While lorem ipsum's still resembles classical Latin, it actually has no meaning whatsoever. As Cicero's text doesn't contain the letters K, W, or Z, alien to latin, these, and others are often inserted randomly to mimic the typographic appearance of European languages, as are digraphs not to be found in the original.

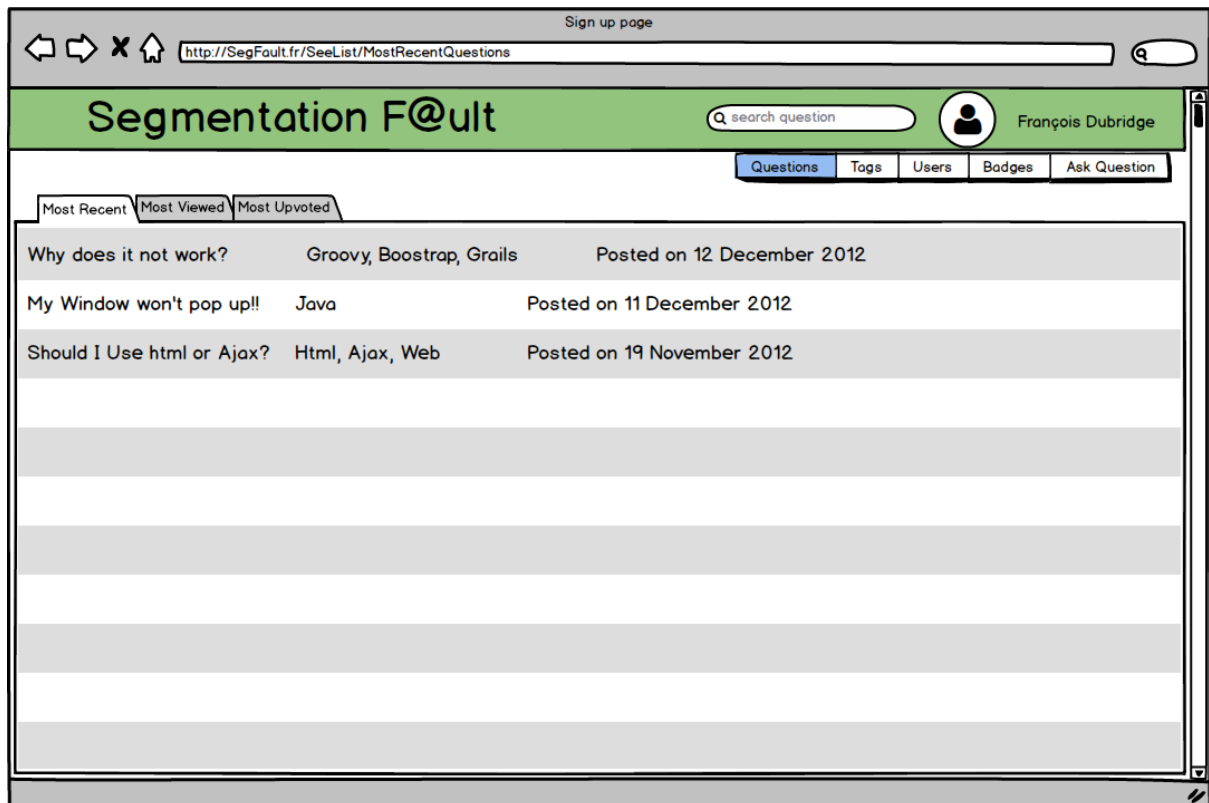
Code:
{

Topics Associated:

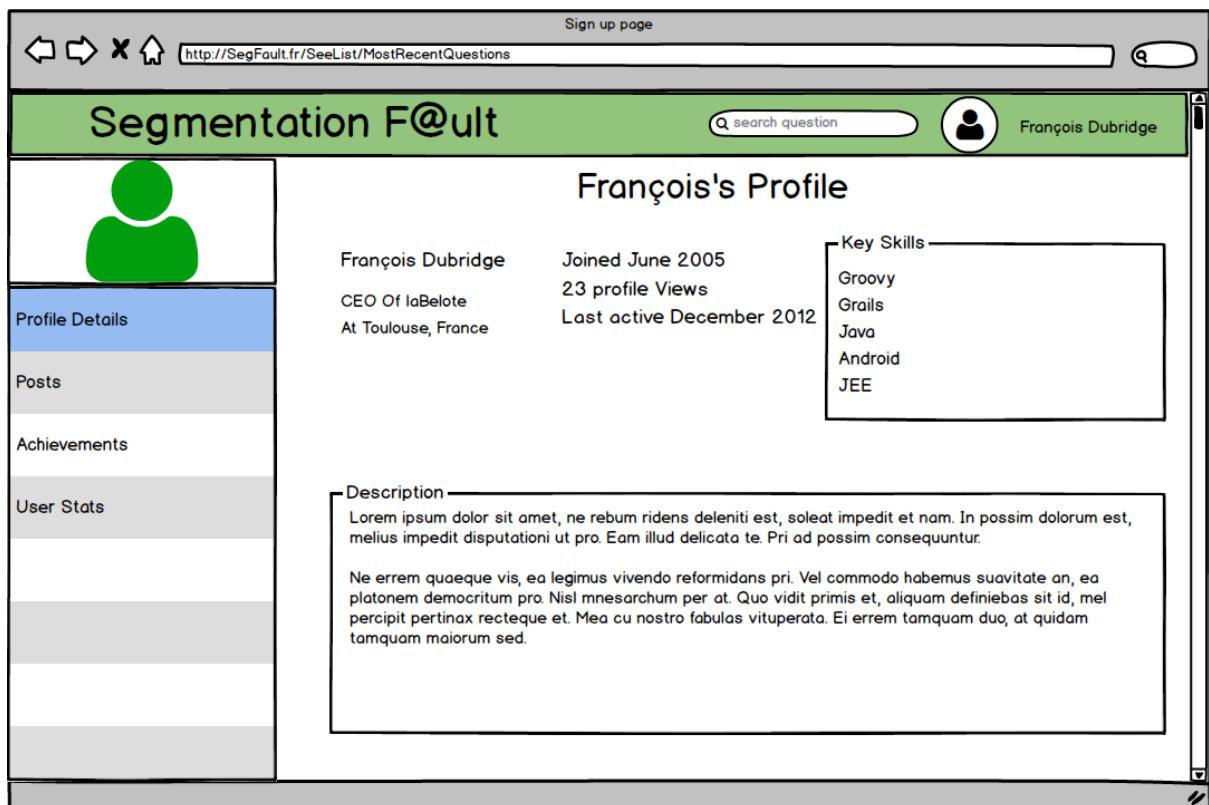
Tags List	Tags Chosen
C	Java
C#	Grails
Cpp	Bootstrap
Lua	

Post Your Question Cancel

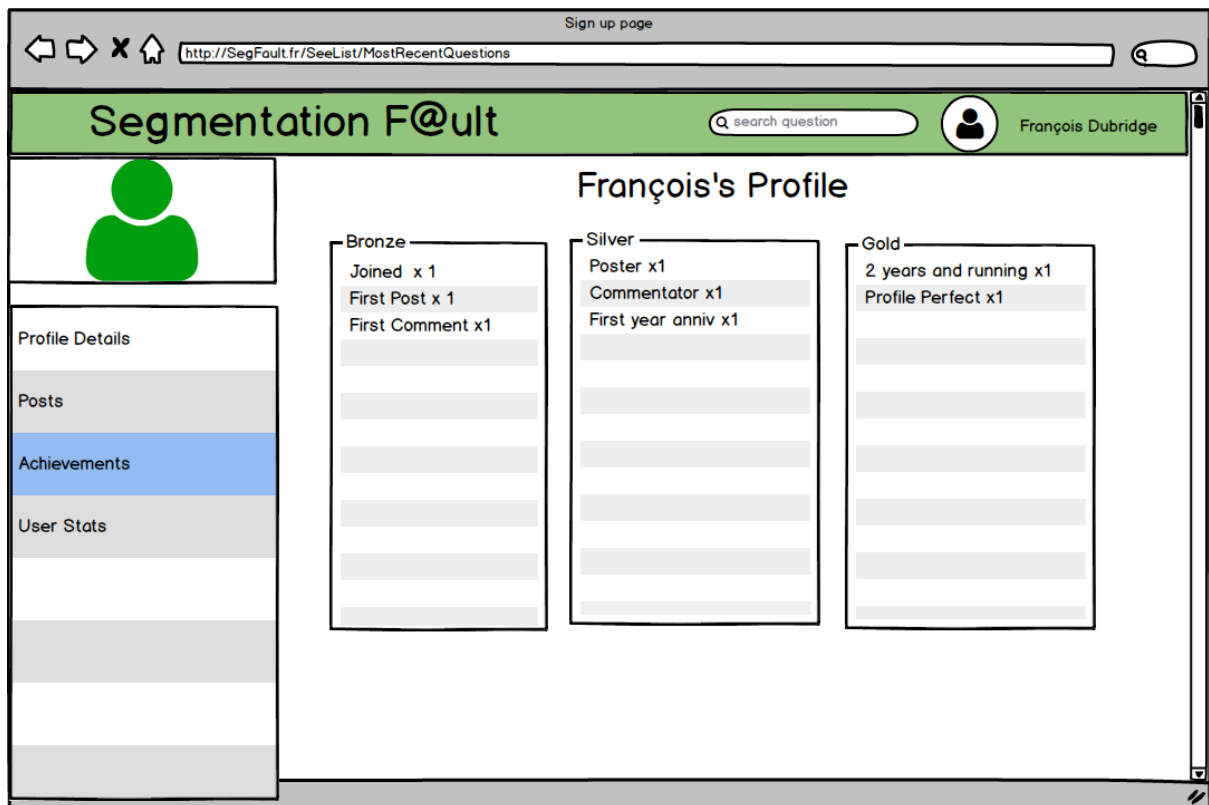
Annex: New Question Form



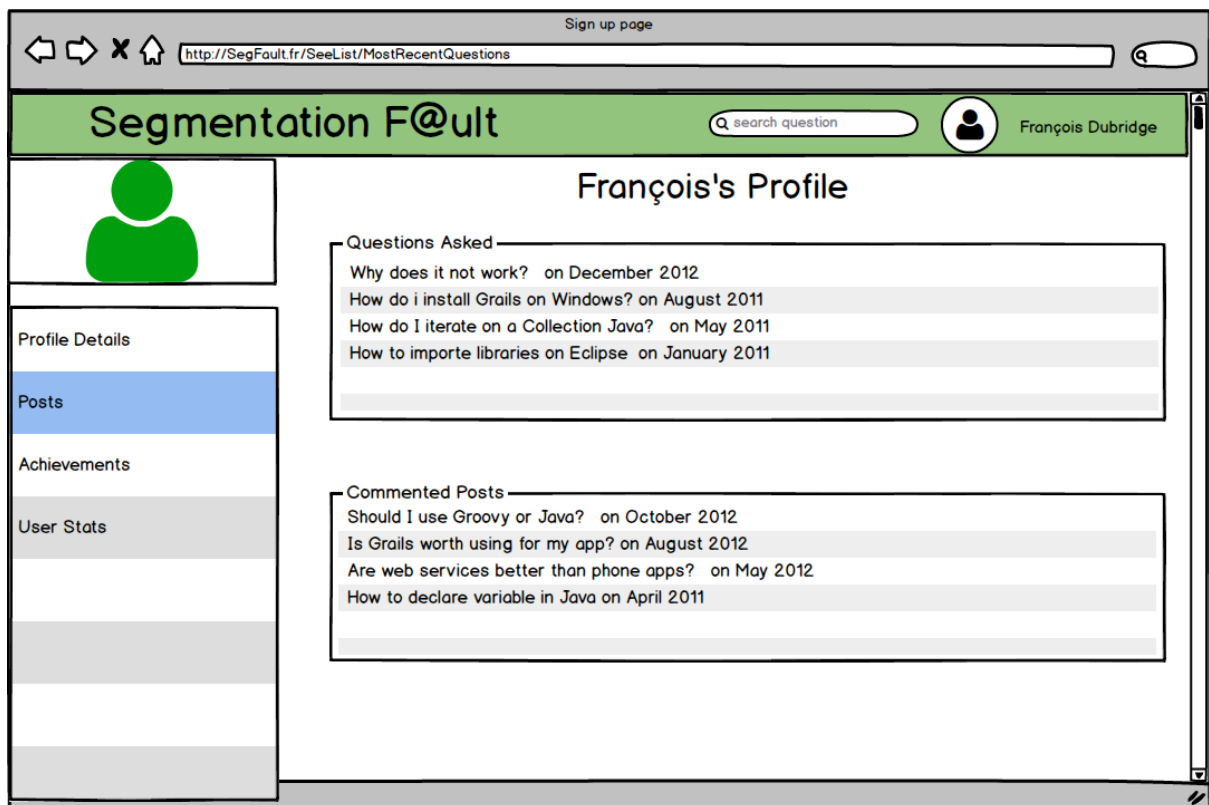
Annex: Display Item List



User Profile



User Achievements



User Activity