

# Milestone 2: Web Interface

## IN705 Databases Three

### Introduction

Our API server is running and we've seen how we can interact with it using JavaScript, it's time to develop our web based user interface. Web interfaces have become the typical way to present database applications, even in corporate IT settings. They can be developed more quickly and are simpler to deploy than desktop applications. We will develop our user interface as a *single page application* using AngularJS.

In order to build our web interface we will need to do some UI design and planning to sort out the desired user interaction and associated interface layout. Implementing the interface using AngularJS is relatively easy, but you will need to dig into some of AngularJS's directives to see how to accomplish this if you have not done so already. You will need to write some controllers to manage the data objects and work with the API server. You may also choose to modify the API server somewhat to better align with your UI plan. We are agile, after all.

### 1 Web interface requirements

Your user interface shall support the following user actions:

- A new user shall be able to create an account.
- An existing user shall be able to “sign in”.
- An existing user shall be able to edit her name and blurb fields.
- An existing user shall be able to compose and submit a new splatt.
- An existing user shall be able to see his splatts feed.
- A user, whether signed in or not, shall be able to view other users' information and splatts.
- An existing user shall be able to follow and unfollow other users.
- An existing user shall be able to delete her account.

**Challenge problem:** Implement a search function or functions that allow a user to search for other users or to search for splatts by keyword.

### 2 Submission requirements

You will submit your source code by *tagging*<sup>1</sup> a commit to your GitHub repository with the tag “M2”. This commit must be dated no later than Friday, 12 September at 18:00 NZST. Late submissions will be penalised

---

<sup>1</sup>See <http://git-scm.com/book/Git-Basics-Tagging>. Use a lightweight or annotated tag labeled “M2”.

10% per day beginning at 18:00:01 each day beginning on 12 September. Improper submissions will not be marked.

You will also deploy a running instance of your web interface on your server. To mark your submission, the lecturer will test and evaluate your deployed instance and will pull your source code from GitHub to evaluate the code quality.

- correct and complete implementation of the requirements: 50%
- A functional, responsive, and appealing interface: 25%
- quality of source code: 25%