COSC2675 Rapid Application Development

Computer Science and Information Technology
School of Science, RMIT

Assignment 2 – Semester 1 2018

Introduction

This assignment is worth **35%** towards your final grade. You are required to develop a web application as specified below. It is designed to:

- Practise your knowledge of Rails development
- Practise testing and debugging

This is a group assignment. A group should not have more than 2 members. Individual submission does not receive any extra marks. By default, both group members receive the same mark unless there is a significant discrepancy in their contributions.

Academic Integrity

The submitted assignment must be your own work. No marks will be awarded for any parts, which are not created by you. More on http://www.rmit.edu.au/academicintegrity.

Plagiarism is treated very seriously at RMIT. Plagiarism includes copying code directly from other students, internet or other resources without proper reference. Sometimes, students study and work on assignments together and submit similar files which may be regarded as plagiarism. Please note that you should always create your own assignment even if you have very similar ideas.

Plagiarism-detection tools may be used for all submissions. Penalties will be applied in cases of plagiarism.

Client Request

This assignment is developed based on the following fictitious message from the school admin team.

Dear Andy

Our school is thinking of building a news website to let everyone share relevant news. RAD students may be able to help as we've heard from many that they know how to quickly build a good website from scratch.

There is no formal specification yet. We just want a news-sharing site, which is easy to use and has good look-n-feel, for our staff and students. People with a proper RMIT email address should be able to register to be a user. Do you think we should only allow registered users to post, update and remove news? Viewing the news should not be a problem. Mmm... it would be nice that the site looks like Hacker News. Students and staff can comment on the news and see latest comments from fellow users.

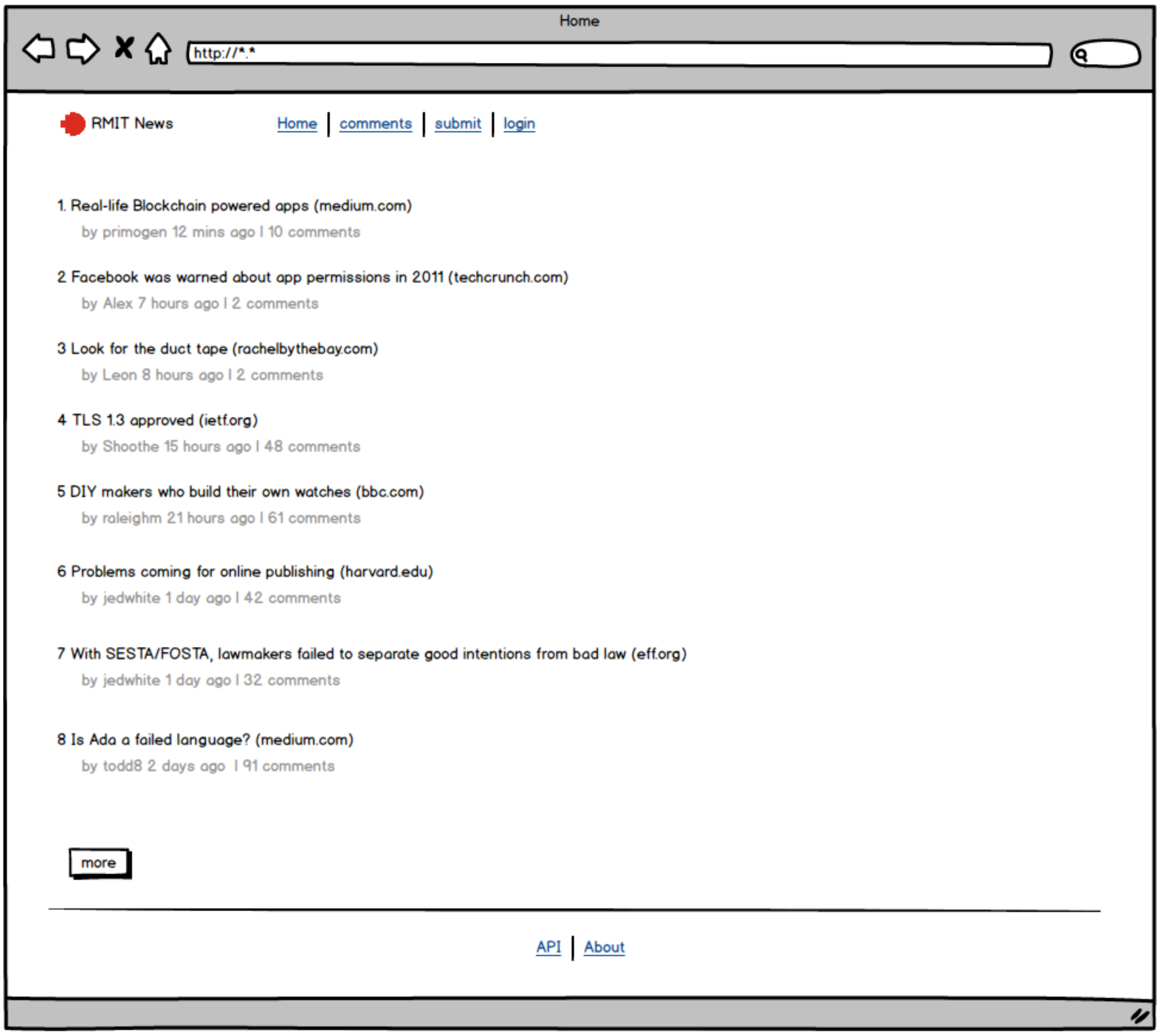
Thank you and talk soon.

RMIT News

In this assignment, you will build a news sharing web application for RMIT. That is kind of a subset of famous hacker news (https://news.ycombinator.com/). For whom is new to Hacker News, it is a social news website focusing on computer science and entrepreneurship. It is run by Paul Graham's investment fund and startup incubator, Y Combinator. In general, news that can be submitted is defined as "anything that gratifies one's intellectual curiosity".

Part 1

Section 1 - Home Page



RMIT News   Home | comments | submit | login

1. Real-life Blockchain powered apps (medium.com)
   by primogen 12 mins ago | 10 comments

2 Facebook was warned about app permissions in 2011 (techcrunch.com)
   by Alex 7 hours ago | 2 comments

3 Look for the duct tape (rachelbythebay.com)
   by Leon 8 hours ago | 2 comments

4 TLS 1.3 approved (ietf.org)
   by Shoothe 15 hours ago | 48 comments

5 DIY makers who build their own watches (bbc.com)
   by raleighm 21 hours ago | 61 comments

6 Problems coming for online publishing (harvard.edu)
   by jedwhite 1 day ago | 42 comments

7 With SESTA/FOSTA, lawmakers failed to separate good intentions from bad law (eff.org)
   by jedwhite 1 day ago | 32 comments

8 Is Ada a failed language? (medium.com)
   by todd8 2 days ago | 91 comments

more

API | About

Your application should meet the following requirements

1. It has a navigation bar with RMIT logo and site name on it

2. The navigation bar must contains at least four hyperlinks
   a. Home – to show the home page
   b. Comments – to show the latest comments (of all news)
   c. Submit – to submit a news (login required)
   d. Login – login or create user account

3. The news content

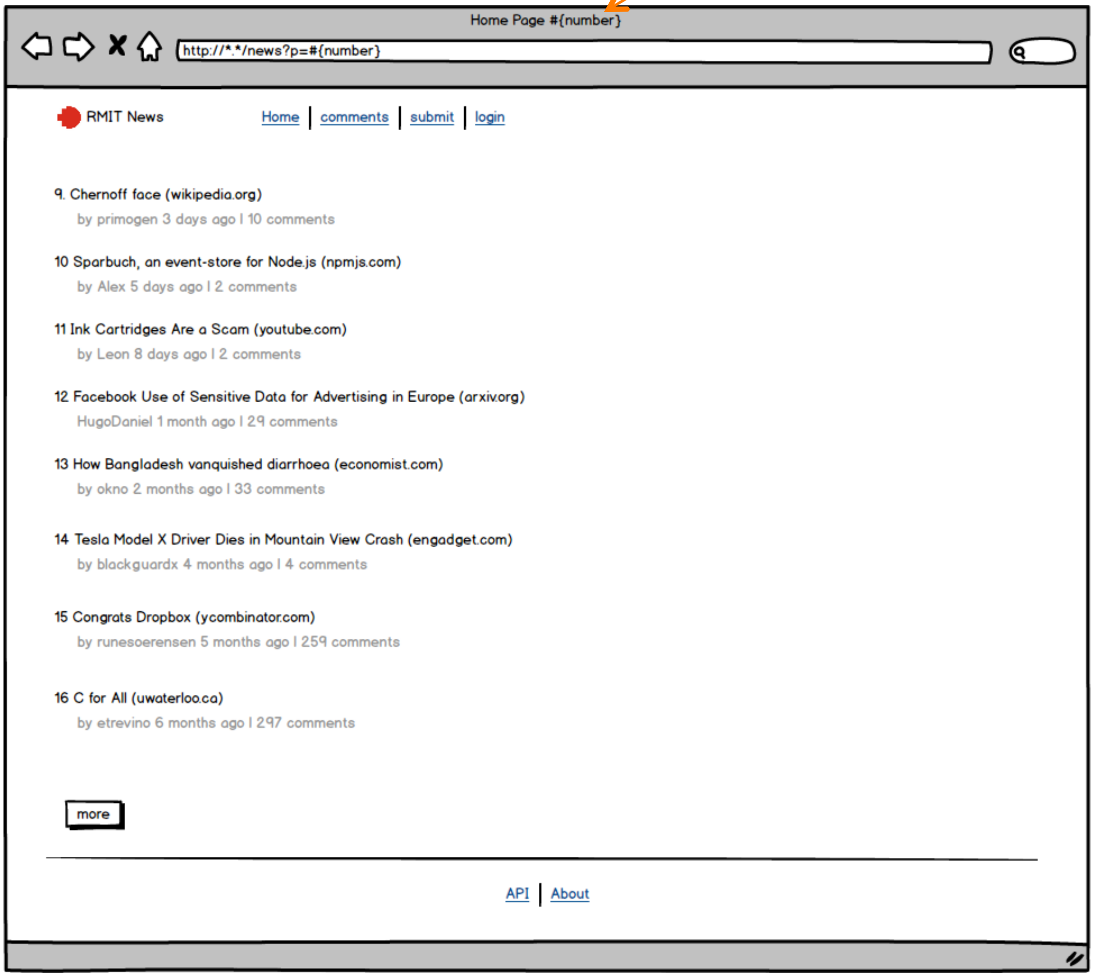
a. The news posts are sorted by the creation date using Unix timestamps.
b. The headline of a news contains the title and the source of the news (if exists)
    i. By clicking on the source, the user will be redirected to the source website.
    ii. The source news text is the domain name of that particular news website.
c. The meta information about each news contains:
    i. The author who submitted the news
    ii. The time when the news got submitted
    iii. The number of total comments on that news
    iv. The text e.g. “`25 comments`”, is clickable and will take the user to the comments page of that news. (see below)

4. The footer
    a. API: a link to the API documentation (https://github.com/pigfly/RMIT-CSIT-RAD/tree/master/2018_S1/assignment2)
    b. About: the makers of your web application
        i. The names of your group members
        ii. The student number of each member.
        iii. Your heroku deployed URL
        iv. Your assignment 2 bitbucket URL

5. Pagination
    a. The news home page will show eight news per page
    b. The pagination is triggered by pressing the more button

Section 2 - Home Page Pagination

2.3



9. Chernoff face (wikipedia.org)
by primogen 3 days ago | 10 comments

10 Sparbuch, an event-store for Node.js (npmjs.com)
by Alex 5 days ago | 2 comments

11 Ink Cartridges Are a Scam (youtube.com)
by Leon 8 days ago | 2 comments

12 Facebook Use of Sensitive Data for Advertising in Europe (arxiv.org)
HugoDaniel 1 month ago | 29 comments

13 How Bangladesh vanquished diarrhoea (economist.com)
by okno 2 months ago | 33 comments

14 Tesla Model X Driver Dies in Mountain View Crash (engadget.com)
by blackguardx 4 months ago | 4 comments

15 Congrats Dropbox (ycombinator.com)
by runesoerensen 5 months ago | 259 comments

16 C for All (uwaterloo.ca)
by etrevino 6 months ago | 297 comments

Requirements:

1. When the more button is clicked, the page will show the next eight news

2. The page's relative URL is `/news?p=#{number}` e.g. `/news?p=2`

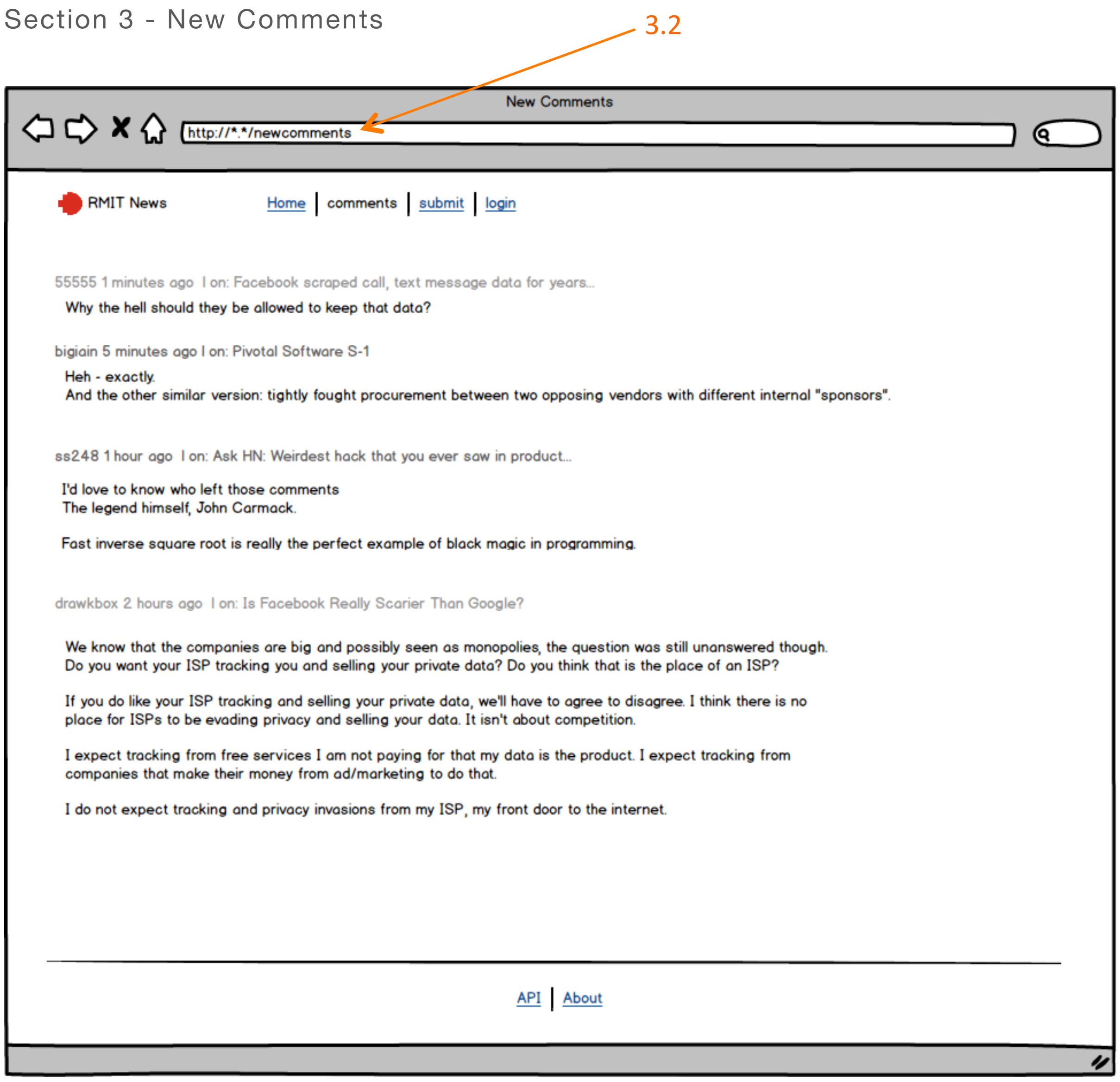
3. The page title should be changed into

   `Home page #{your_current_pagination_number}`

   The page number for the above example is 1.

   The page number is zero indexed, meaning that number for the Home page is 0.

Section 3 - New Comments



55555 1 minutes ago I on: Facebook scraped call, text message data for years...

Why the hell should they be allowed to keep that data?

bigiain 5 minutes ago I on: Pivotal Software S-1

Heh - exactly.
And the other similar version: tightly fought procurement between two opposing vendors with different internal "sponsors".

ss248 1 hour ago I on: Ask HN: Weirdest hack that you ever saw in product...

I'd love to know who left those comments
The legend himself, John Carmack.

Fast inverse square root is really the perfect example of black magic in programming.

drawkbox 2 hours ago I on: Is Facebook Really Scarier Than Google?

We know that the companies are big and possibly seen as monopolies, the question was still unanswered though.
Do you want your ISP tracking you and selling your private data? Do you think that is the place of an ISP?

If you do like your ISP tracking and selling your private data, we'll have to agree to disagree. I think there is no place for ISPs to be evading privacy and selling your data. It isn't about competition.

I expect tracking from free services I am not paying for that my data is the product. I expect tracking from companies that make their money from ad/marketing to do that.

I do not expect tracking and privacy invasions from my ISP, my front door to the internet.

Requirements:

1. The New Comments page shows the latest comments of all news posts.

2. The New Comments page's relative URL path is `/newcomments`

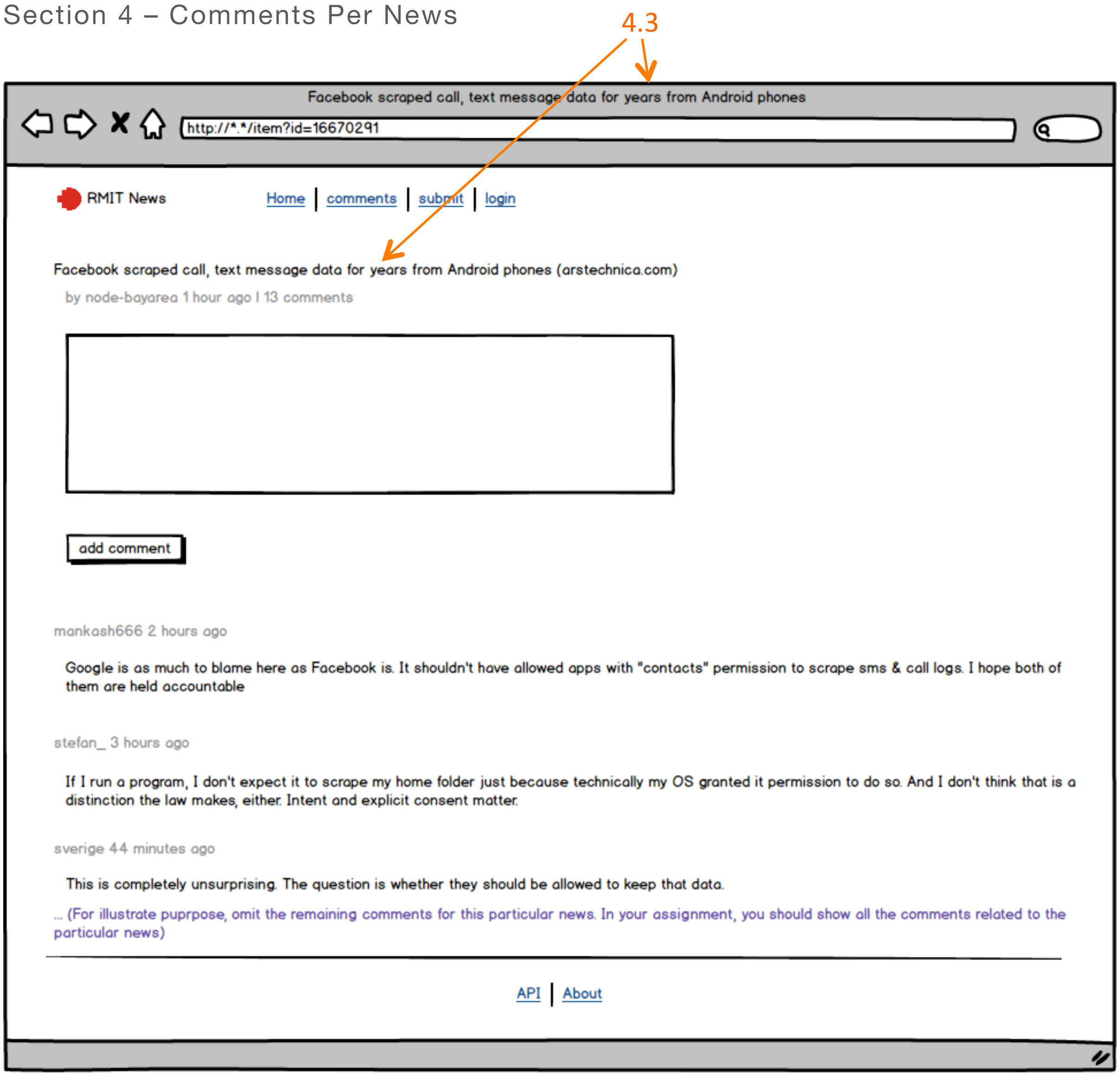
3. The New Comments page is sorted by the creation date & time

4. Each comment has two parts:
    a. Headline: include the user who commented on that particular news, the date that the comment was created, and the title of the news.

    b. Body: the comment message body

5. The New comments page will show and only show three comments sorted by the creation date.

Section 4 – Comments Per News



4.3
Facebook scraped call, text message data for years from Android phones
http://*.*/item?id=16670291
RMIT News
Home
comments
submit
login
add comment
API
About

Facebook scraped call, text message data for years from Android phones (arstechnica.com)
by node-bayarea 1 hour ago | 13 comments

mankash666 2 hours ago

Google is as much to blame here as Facebook is. It shouldn't have allowed apps with "contacts" permission to scrape sms & call logs. I hope both of them are held accountable

stefan_ 3 hours ago

If I run a program, I don't expect it to scrape my home folder just because technically my OS granted it permission to do so. And I don't think that is a distinction the law makes, either. Intent and explicit consent matter.

sverige 44 minutes ago

This is completely unsurprising. The question is whether they should be allowed to keep that data.

... (For illustrate pupprose, omit the remaining comments for this particular news. In your assignment, you should show all the comments related to the particular news)

Requirements:
1. The Comments page of each news shows a text area for commenting and all the comments related to that particular news. Comments are sorted by date and time.

2. The Comment Per News page's relative URL path is `/item?id=#{id_number}`

3. The page title is the same title of the news, which is to be commented on.

4. The comment section has four parts:
    a. Headline: the news title

    b. Sub header: the creator of the news, the news creation date/time and the total number of comments at present
    c. A text area for the commenting
    d. A submit button - “add comment”

5. Validation rules for comment messages
    a. Must not be blank
    b. Must be at least 3 characters **excluding** white spaces, new lines and tabs
    c. Must be less than 1000 characters

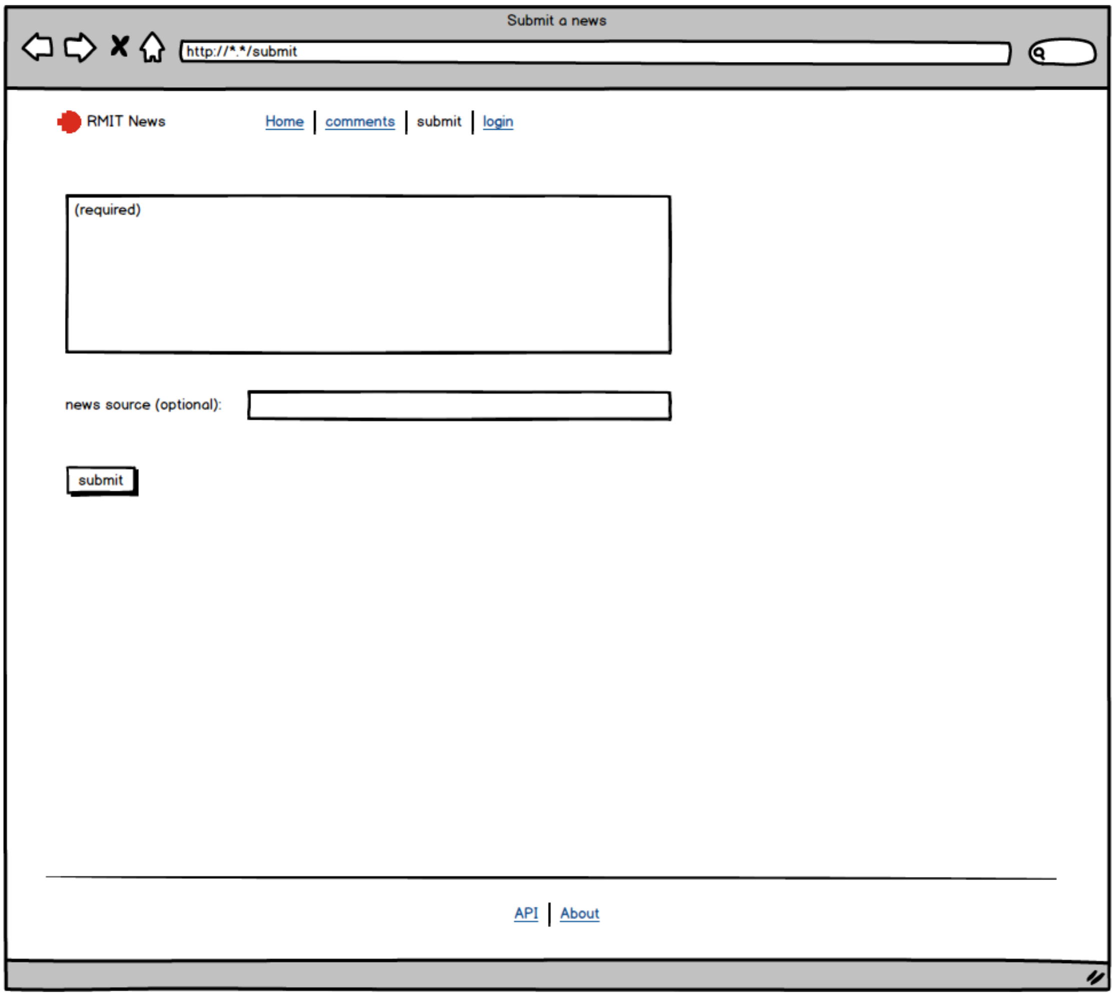
6. Banner messages:
    a. If the comment **doesn’t** pass the validation, show a warning banner message (see banner message page in Section 6).

    b. If comment **does** pass the validation, show a success banner message (see banner message page in Section 6) and redirect the user to `/newcomments`

    c. You can decide the text for both types of banner messages yourself.

7. The list of all comments
    a. It should show all the comments that belongs to this particular news
    b. All the comments that belongs to this particular news should be sorted by creation date/time
    c. It should show no comments if there is no comment for that news

Section 5 - Submit



Requirements:

1. The Submit Page allows the user to submit a news

2. The Submit Page's relative URL path is `/submit`

3. If the user is not logged in, redirect the user to the login page (`/login`)

4. The navigation title of this page should be "Submit a news"

5. The submit section has three parts
    a. The text area for the submitted news title (required)
    b. The source of the news (optional)
    c. The submit button

6. Validation rules for the submit text area:
    a. Must not be blank
    b. Must be at least 10 characters (**including** white spaces, new lines and tabs)
    c. Must be less than 200 characters

    Note: The text area is for entering the news title only. There is no news body.

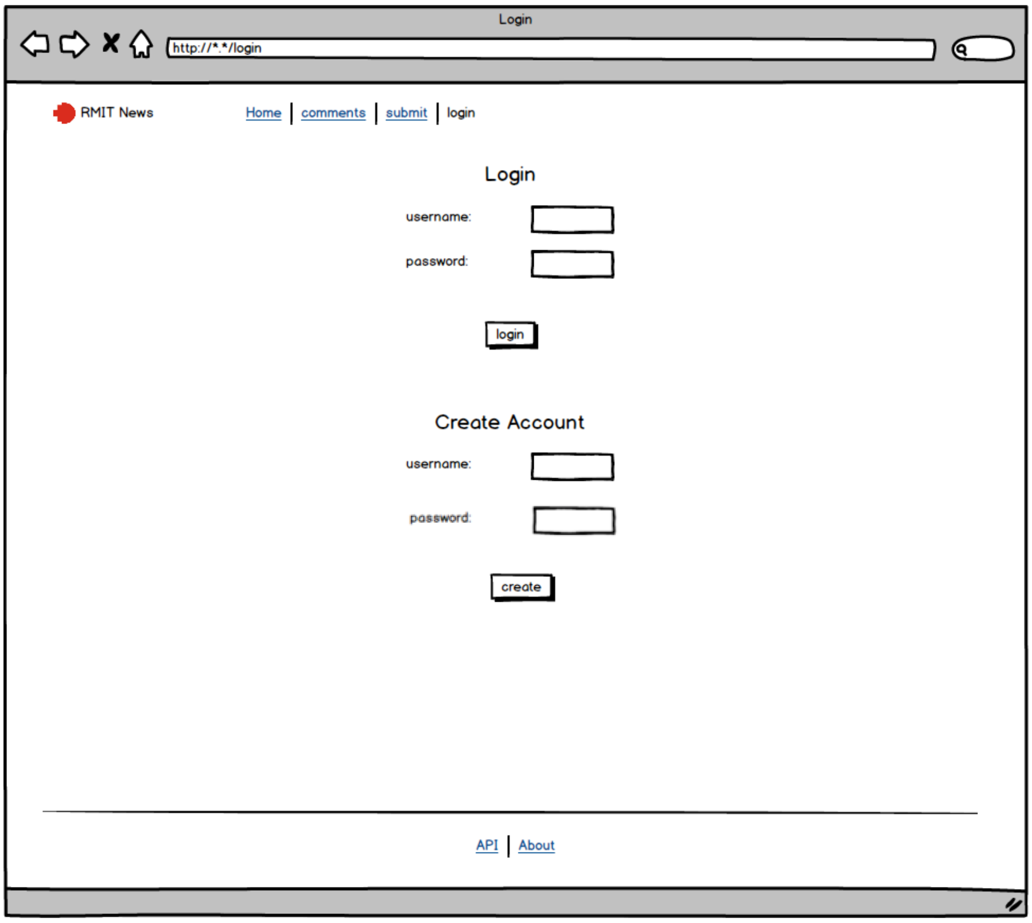
7. Validation rule for the source of the news
    a. Must be a valid URL if the URL is not empty

8. Banner messages
    a. If a submission **doesn't** pass the validation, show a warning banner message (see banner message page in Section 6).

    b. If comment **does** pass the validation, show a success banner message (see banner message page in Section 6) and redirect the user to **the home page.**

    You can decide the text for both types of banner message yourself.

Section 5 - Login



Requirements:

1. The login page shows the login and creates an account

2. The login page's relative URL is `/login`

3. The validation rule for user name
    a. Must not be blank
    b. Can only contain letters, digits, dashes and underscores.
    c. Should be between 2 and 15 characters.
    d. Must be a unique username in the database.

4. The validation rule for password
    a. Must not be blank
    b. Must contain at least one uppercase letter, one special character, one number and one lowercase letter
    c. Must contain at least 10 characters

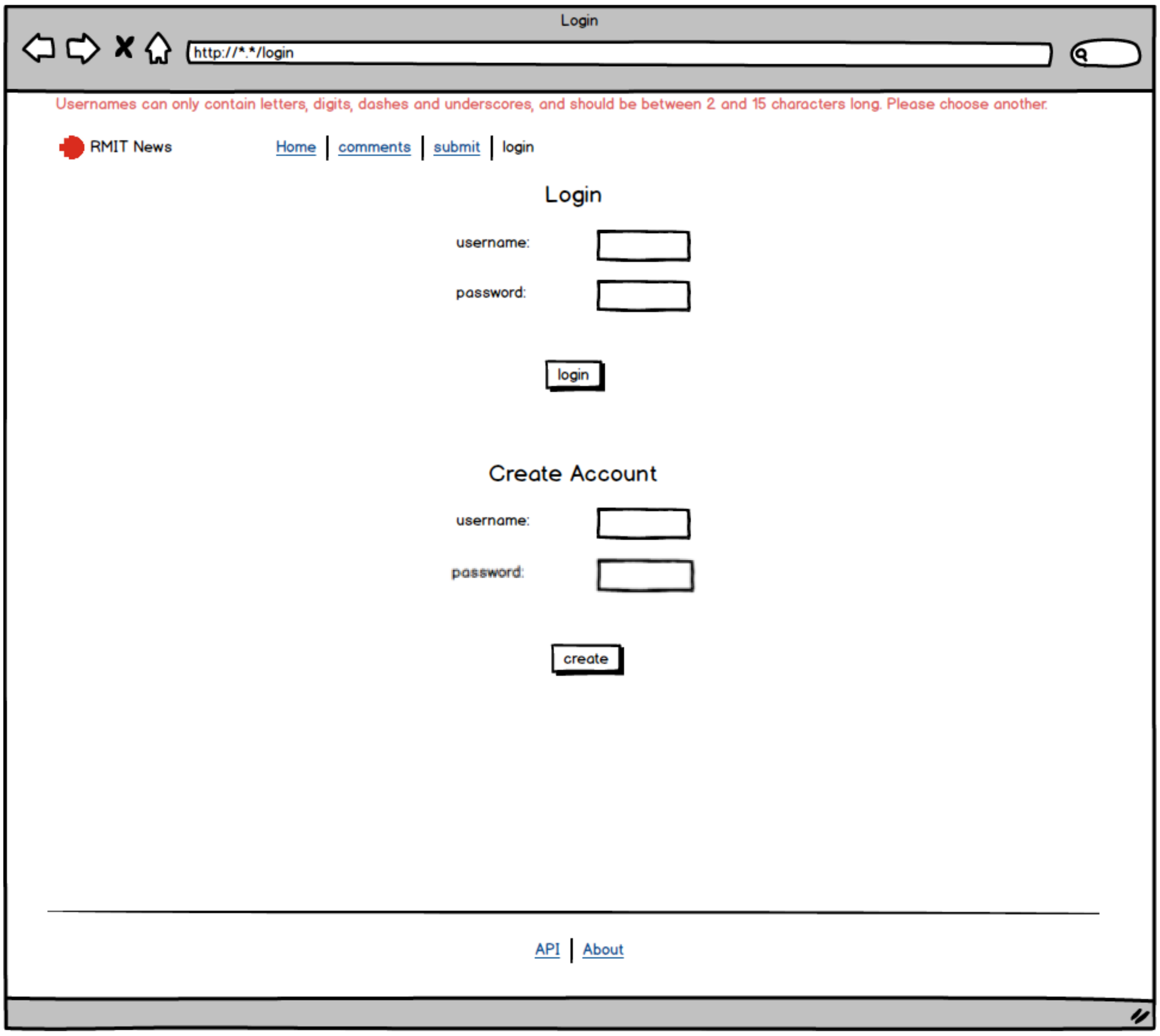
5. Both username and password must be valid in order to pass the validation

6. Banner messages
    a. If a submission **doesn't** pass the validation, show a warning banner message (see banner message page in Section 6).

    b. If comment **does** pass the validation, show a success banner message (see banner message page in Section 6) and redirect the user to **the home page.**

       You can decide the text for both types of banner message yourself.

Section 6 - Banner Message



Login

http://*.*/login

Usernames can only contain letters, digits, dashes and underscores, and should be between 2 and 15 characters long. Please choose another.

RMIT News

Home | comments | submit | login

Login

username:

password:

login

Create Account

username:

password:

create

API | About

Guidelines

- Your web **application** should be mobile friendly
- You can use your favourite CSS/HTML style as you wish

Part 2 API

In this part, you will implement your API endpoint for RMIT news.

Overview

This is sample documentation for RMIT News API (RAD course)

Please post your relevant thoughts in canvas if found anything unclear.

URI and Versioning

The first version will have URIs prefixed with `http://*.*/v0/` and is structured as described below.

Design

The v0 API is essentially a dump of your database.

It's not the ideal public API, but it's the one you can get a taste how RESTful API works internally.

Items

News, comments are just items. They're identified by their ids, which are unique integers, and live under /v0/item/<id>.

All items have some of the following properties, with required properties in bold:

Field	Description
id	The item's unique id.
type	The type of item. One of "news", "comment"
by	The username of the item's author.
time	Creation date of the item, in Unix Time.
text	The comment text

Field	Description
url	The source URL of the news
title	The title of the news

For example, a news: http://yourownurl/v0/item/8863.json

```
{
  "by" : "dhouston",
  "id" : 8863,
  "time" : 1175714200,
  "title" : "My YC app: Dropbox - Throw away your USB drive",
  "type" : "news",
  "url" : "http://www.getdropbox.com/u/2/screencast.html"
}
```

comment: http://yourownurl/v0/item/2921983.json

```
{
  "by" : "norvig",
  "id" : 2921983,
  "text" : "Aw shucks, guys ... you make me blush with your compliments.<p>Tell you what, Ill make a deal:
I'll keep writing if you keep reading. K?",
  "time" : 1314211127,
  "type" : "comment"
}
```

another news: http://yourownurl/v0/item/121003.json

```
{
  "by" : "tel",
  "id" : 121003,
  "text" : "<i>or</i> HN: the Next Iteration<p>I get the impression that with Arc being released a lot of
people who never had time for HN before are suddenly dropping in more often. (PG: what are the numbers
on this? I'm envisioning a spike.)<p>Not to say that isn't great, but I'm wary of Diggification. Between links
comparing programming to sex and a flurry of gratuitous, ostentatious  adjectives in the headlines it's a bit
concerning.<p>80% of the stuff that makes the front page is still pretty awesome, but what's in place to
keep the signal/noise ratio high? Does the HN model still work as the community scales? What's in store
for (++ HN)?",
  "time" : 1203647620,
  "title" : "The Arc Effect",
  "type" : "news",
  "url" : ""
}
```

Endpoint to implement:

- /v0/item/<id>
  - Request Type: GET
  - Authentication: None
  - Return status code: 200 for OK, 400 for Errors

- /v0/item/create
    - Request Type: POST
    - Request Body:

```
{
  "username" : "alex",
  "text" : "Create a sample news",
  "type" : "news",
  "new source" : "https://www.rmit.edu.au"
}
```

    - Authentication: None
    - Return status code: 200 for OK, 400 for Errors

Additional Notes

1: Your application is expected to be production ready. Proper test and data validation should be embedded in your application.

2: There is no need for your application to look and behave exactly like the above example. You can make necessary adjustment to improve the design. However the aforementioned functionalities are the minimum requirement and should be well supported.

3: Only the following GEMS should be used in your application.

```
gem 'rails'
gem 'bootstrap-sass'
gem 'puma'
gem 'sass-rails'
gem 'uglifier'
gem 'coffee-rails'
gem 'jquery-rails'
gem 'turbolinks'
gem 'jbuilder'
gem "font-awesome-rails"
gem 'bcrypt'
```

Submission

- This assignment will also be using bitbucket to host all your Rails files

  https://bitbucket.org/

  - Create a **private repo** named as `RAD18_Assign_2_group_{studentNumber_small}_{studentNumber_large}`

ONLY ONE submission for each group is needed. For duplicate group submissions, we will randomly select one to mark.

  - Make a branch called "***submission***"
  - Invite the following people as default reviewers to your project
    - junliang.jiang@rmit.edu.au
    - anto.dominic@rmit.edu.au
    - andy.song@rmit.edu.au
  - Make a pull request to merge your branch ***submission*** to branch ***master*** and include the following default reviewers (to ensure our access)
    - junliang.jiang@rmit.edu.au
    - anto.dominic@rmit.edu.au
    - andy.song@rmit.edu.au
  - **The due date: 11:59pm of the 20th of May.**
  - **A final submission before 15th May 11:59pm will receive 10% extra as the bonus.**
  - **This assignment will attract additional marks for individual lab demonstration.**
  - **For one professional developer working alone, the expected completion time is less than 5 full days.**
    - It is a good habit to keep track of the hours that you spent on a project. For that purpose, we recommend toggl.com, a free online time tracker.

- Warning
  - You will receive ZERO mark if you don't host your assignment in bitbucket before the due day (see details later).
  - You will receive ZERO mark if your application fails to run (see details later).