Flipboard Like HTML5 App – FrontEnd Assembly

1. Project Overview

Client wants to develop a Hybrid mobile application built in HTML5/JS/CSS, which will be deployed to iOS and Android - Phone & Tablet sized.
This mobile application will aggregate RSS data from external sources, social media content, and internal content served via REST API, similar as Flipboard.
For internal content, we'd like to seek an existing content management sytem that fits our client's requirement to reduce the development effort.

In this module assembly, we will build mobile application based on PhoneGap and HTML5.

2. Requirements

2.1. General Approach

Please see section 2 of the ADS.

2.2. Social Network Integration

2.2.1. Facebook

Facebook login will be done using the Javascript SDK Web Login process. It is described in good detail at https://developers.facebook.com/docs/facebook-login/login-flow-for-web/

The only thing we need beforehand for this process is the Facebook AppId which can be generated in the App Dashboard https://developers.facebook.com/apps/

Note that in the case of Facebook, we do not need to store the retrieved accessToken at our end. The JavaScript SDK takes care of storing the accessToken at their end.

Sharing article on FB will also be done using the JavaScript SDK. The FB.ui method will be used where method is 'feed'.
See the example at https://developers.facebook.com/docs/javascript/reference/FB.ui
The mapping between Article fields and FB.ui fields for sharing is quite obvious.

The current user's FB news feed is accessible by making a call to the /me/home endpoint. This can also be done using Javascript SDK using the FB.api call. See https://developers.facebook.com/docs/graph-api/reference/user/home/
Since the FB news feed is also displayed like articles, the mapping from FB response and Article fields is as follows:

FB JSON Response Field	Article
from.name	author
link	link
name	title

picture	image
message	description

2.2.2. Twitter

Twiiter OAuth login is described at https://dev.twitter.com/docs/auth/implementing-sign-twitter
Twitter does not provide any client side libraries for performing the login. So our app will have to do the OAuth dance ourselves. The steps provided in the above link are quite comprehensive though and can be followed by developer.

A description of doing general OAuth on PhoneGap is provided at http://phonegap-tips.com/articles/google-api-oauth-with-phonegaps-inappbrowser.html
Note the usage of the a localhost address for the callback url, which we just use to get the OAuth response data and then close the InAppBrowser.
The access-token received must be stored on the device in localstorage.

Twitter API calls must then be authenticated using the custom Authorization header used by Twitter. It is described at https://dev.twitter.com/docs/auth/authorizing-request. The header contains many fields; the Access-token is what we have stored on device and the other fields are all created on a per-request basis.

Twitter Share or Tweets can be performed either by using their in-built Tweet Button https://dev.twitter.com/docs/tweet-button or more generally using the statuses/update API endpoint. https://dev.twitter.com/docs/api/1.1/post/statuses/update. The former can be used if we are willing to use their Tweet Button UI and the latter must be used if we want to use our own custom Tweet Button UI.

Twitter Feeds can be read simply by using the statuses/user_timeline API endpoint. It is described at https://dev.twitter.com/docs/api/1.1/get/statuses/user_timeline
Since the Twitter news feed is also displayed like articles, the mapping from FB response and Article fields is as follows:

Twitter JSON Response Field	Article
user.name	author
entities.urls[0].expanded_url (if entities.urls has length > 0)	link
text	title
user.profile_image_url	image
text	description

2.2.3. LinkedIn

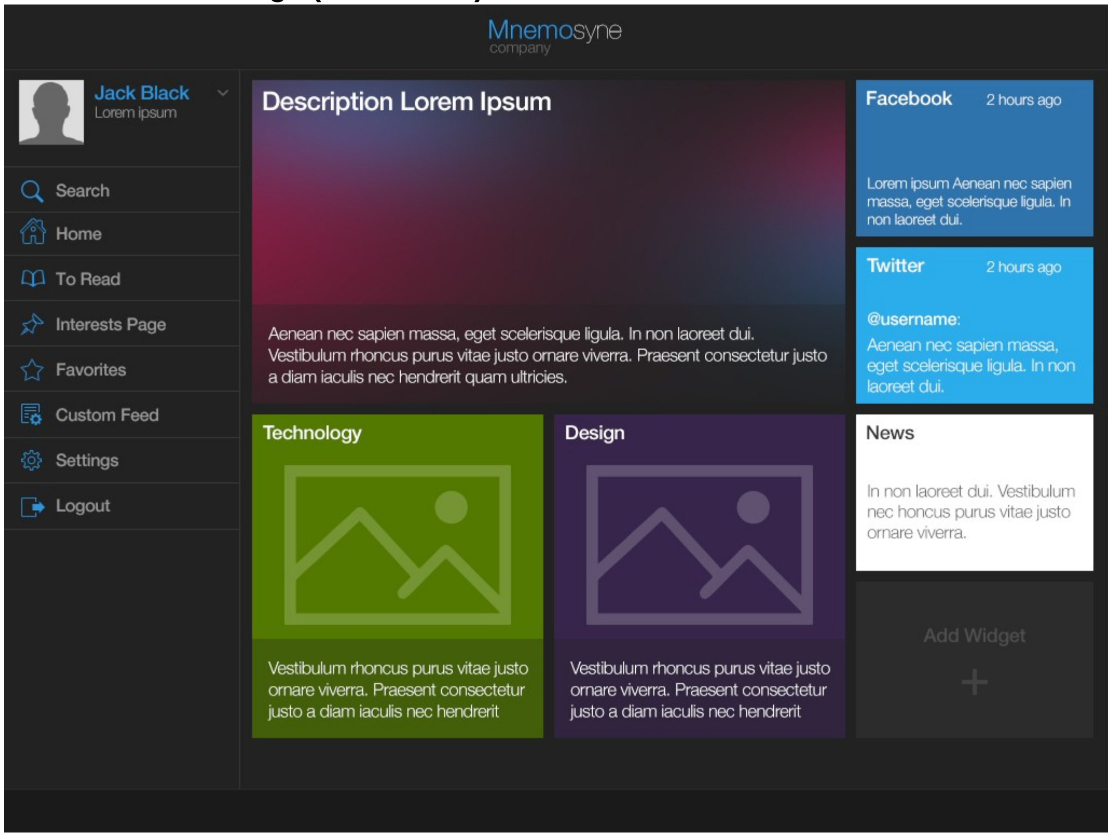
For LinkedIn, the OAuth login process is defined at https://developer.linkedin.com/documents/authentication. It is also a normal OAuth implementation almost exactly the same as the Twitter OAuth (see above).

For LinkedIn, we only need the Share functionality which is covered at https://developer.linkedin.com/documents/share-api

NOTE: To make HTTP calls to arbitrary domains, we need to whitelist them. See http://stackoverflow.com/questions/18263578/ajax-request-from-phonegap-android-fails

2.3. UI Mapping

2.3.1. Home Page (Dashboard)



Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui. Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis nec hendrerit quam ultricies.

Lorem ipsum Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui.

@username: Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui.

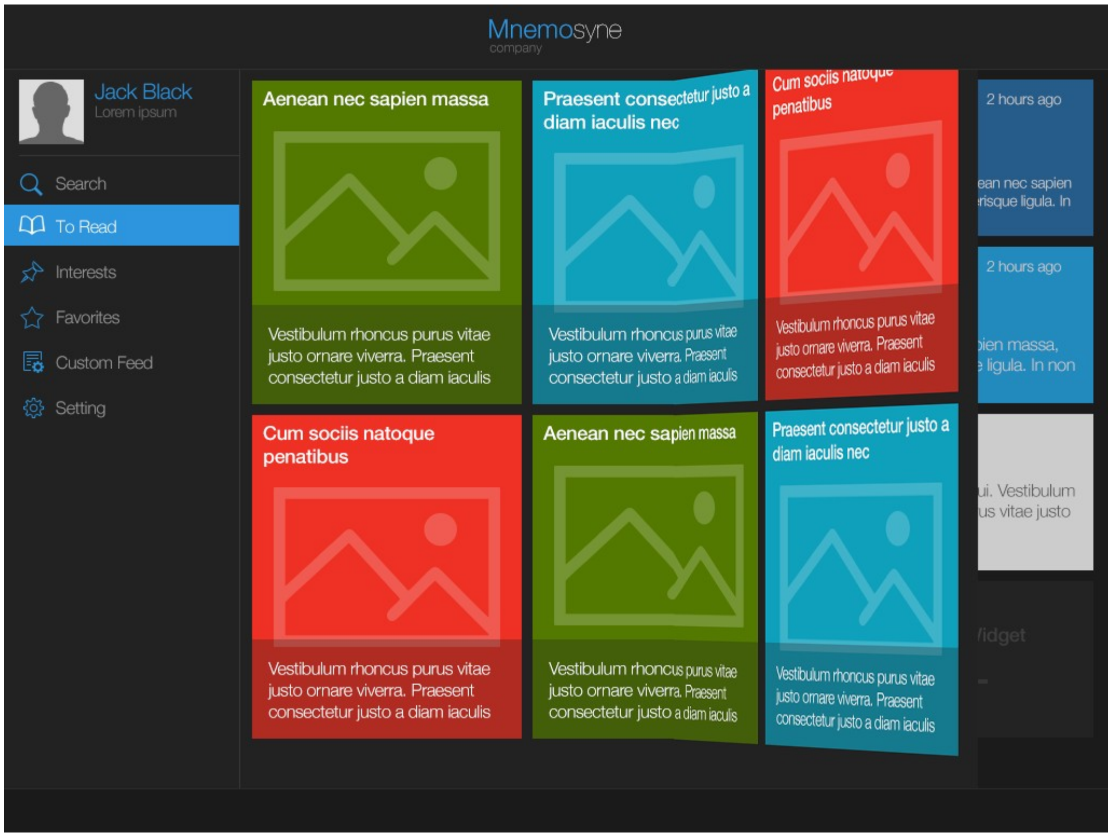
In non laoreet dui. Vestibulum nec honcus purus vitae justo ornare viverra.

Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis nec hendrerit

Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis nec hendrerit

1. Use the getCategories API to get all the categories.
2. Use the getInterests API to get the categories of interest for the user (if logged in).
3. Get first 2 interests (or categories if not logged in) and show them in the bottom 2 tiles. (in the image above, Technology and Design). The image to be shown is the category.image and the text below the image is the category.description.
4. The Facebook and Twitter feeds are described above.
5. The main tile can be shown by calling getArticlesForCategory for a random category and binding the article title and description to the tile. (In Flipboard this tile is for Flipboard picks, but we do not have any such concept in our system for now)
6. On flip, the next n interests can be shown on tiles just like above.

2.3.2. To Read



Aenean nec sapien massa
Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis

Praesent consectetur justo a diam iaculis nec
Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis

Cum sociis natoque penatibus
Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis

ean nec sapien
risque ligula. In

pien massa,
ligula. In non

Cum sociis natoque penatibus
Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis

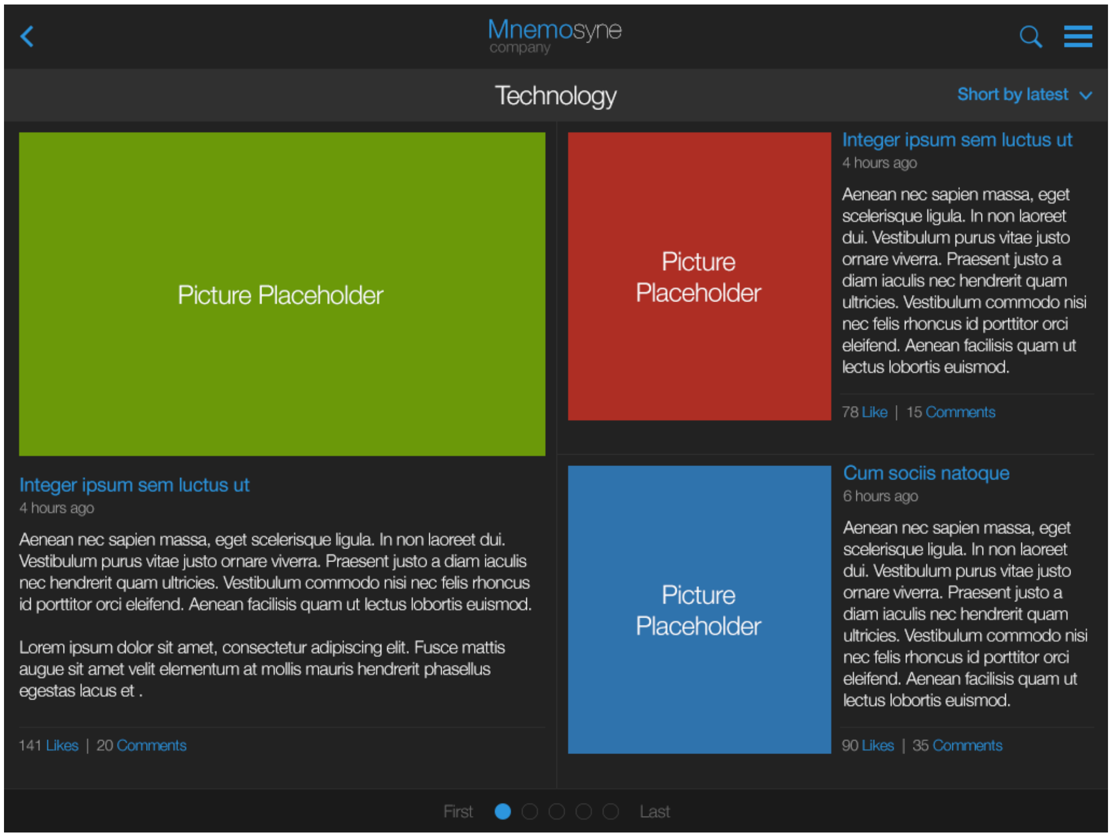
Aenean nec sapien massa
Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis

Praesent consectetur justo a diam iaculis nec
Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis

ui. Vestibulum
us vitae justo

1. Use the getToRead API to get all the to read articles. Use the paging feature.
2. Each article can be displayed using the article.title as the header and the first 200 characters of article.description as the text.
   Note that articles do not have photos. Even in Flipboard most articles are shown without photos. Showing images would require the custom conversion process of each article page. This is an open item at http://apps.topcoder.com/forums/?module=Thread&threadID=816845&start=0
3. On page flip we can use the pagination feature of the getToRead API to show the next page of tiles.

2.3.3. View Category



Integer ipsum sem luctus ut
4 hours ago

Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui. Vestibulum purus vitae justo ornare viverra. Praesent justo a diam iaculis nec hendrerit quam ultricies. Vestibulum commodo nisi nec felis rhoncus id porttitor orci eleifend. Aenean facilisis quam ut lectus lobortis euismod.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce mattis augue sit amet velit elementum at mollis mauris hendrerit phasellus egestas lacus et .

141 Likes | 20 Comments

Integer ipsum sem luctus ut
4 hours ago

Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui. Vestibulum purus vitae justo ornare viverra. Praesent justo a diam iaculis nec hendrerit quam ultricies. Vestibulum commodo nisi nec felis rhoncus id porttitor orci eleifend. Aenean facilisis quam ut lectus lobortis euismod.

78 Like | 15 Comments

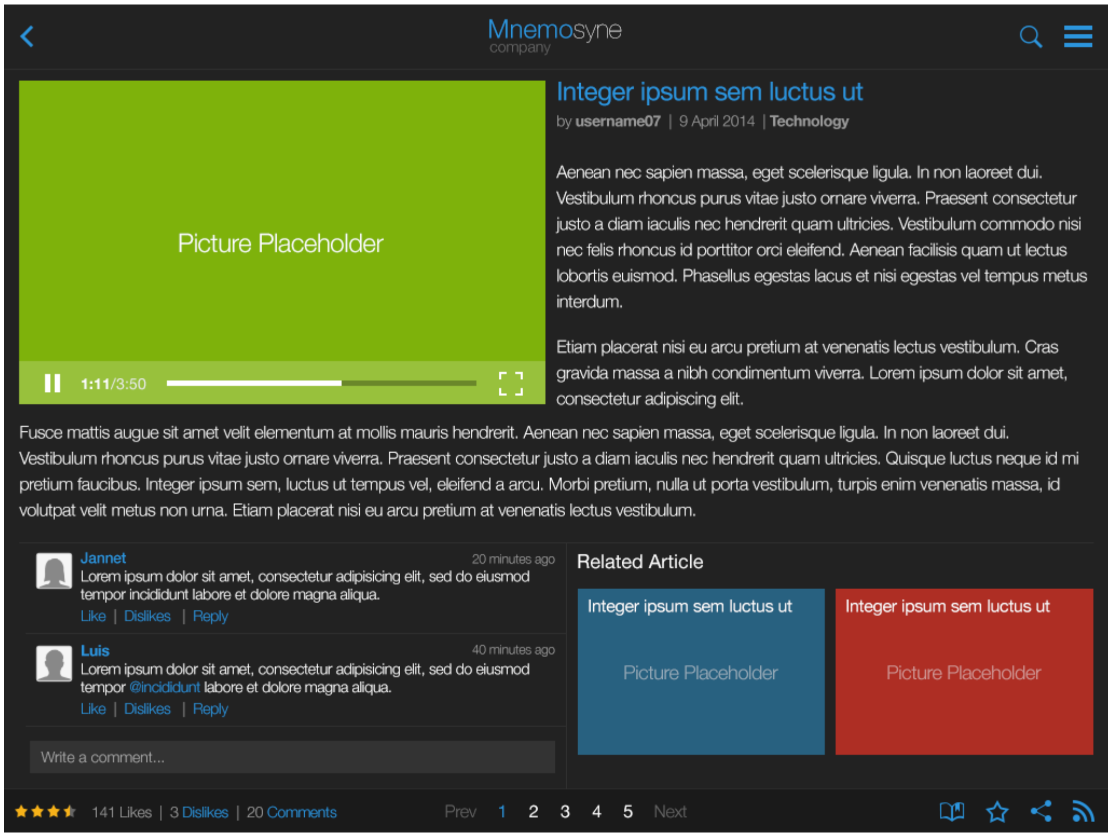
Cum sociis natoque
6 hours ago

Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui. Vestibulum purus vitae justo ornare viverra. Praesent justo a diam iaculis nec hendrerit quam ultricies. Vestibulum commodo nisi nec felis rhoncus id porttitor orci eleifend. Aenean facilisis quam ut lectus lobortis euismod.

90 Likes | 35 Comments

1. Use the getArticlesForCategory with the paging parameters set and optionally the sorting parameters.
2. Each article can be displayed using the article.title as the header, the first 200 characters of article.description as the text, the article.pubDate used to display the recency of the article and the total_likes and total_comments for the namesake fields.
   Note that articles do not have photos. Even in Flipboard most articles are shown without photos.
3. On page flip we can use the pagination feature of the getArticlesForCategory API to show the next page of tiles.

2.3.4. View Article



Mnemosyne
company

Integer ipsum sem luctus ut

by **username07** | 9 April 2014 | **Technology**

Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui. Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis nec hendrerit quam ultricies. Vestibulum commodo nisi nec felis rhoncus id porttitor orci eleifend. Aenean facilisis quam ut lectus lobortis euismod. Phasellus egestas lacus et nisi egestas vel tempus metus interdum.

Etiam placerat nisi eu arcu pretium at venenatis lectus vestibulum. Cras gravida massa a nibh condimentum viverra. Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Fusce mattis augue sit amet velit elementum at mollis mauris hendrerit. Aenean nec sapien massa, eget scelerisque ligula. In non laoreet dui. Vestibulum rhoncus purus vitae justo ornare viverra. Praesent consectetur justo a diam iaculis nec hendrerit quam ultricies. Quisque luctus neque id mi pretium faucibus. Integer ipsum sem, luctus ut tempus vel, eleifend a arcu. Morbi pretium, nulla ut porta vestibulum, turpis enim venenatis massa, id volutpat velit metus non urna. Etiam placerat nisi eu arcu pretium at venenatis lectus vestibulum.

Jannet — 20 minutes ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt labore et dolore magna aliqua.
Like | Dislikes | Reply

Luis — 40 minutes ago
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor @incididunt labore et dolore magna aliqua.
Like | Dislikes | Reply

Write a comment...

1. Use the getArticle call for getting all the details of the article.
2. For displaying the article content itself, the entire article will be loaded into an iframe with the src set to articleHeader.link
   If however the articleHeader.is_video is true, we will use the HTML5 video tag to load the video at articleHeader.link.
   Note that articles do not have photos. Even in Flipboard most articles are shown without photos.
3. The mapping of the remaining Article fields to this page is straightforward.

2.3.5. Favorites page

The user's favorite articles can be retrieved using the getFavorites API call. The flipping can again be done using the paging calls. The mapping of the ArticleHeader fields and the page is straightforward. The favorites can be set and deleted using the saveFavorite and deleteFavorite calls.

2.3.6. Interests page

The user's interest categories can be retrieved using the getInterests API call. The mapping of the Category fields and the page is straightforward. The intrests can be set and deleted using the saveInterest and deleteInterest calls.

2.3.7. Search page

There is no search page in the storyboards. But a search page similar to that of Flipboard can be implemented by making call to the search API call.

2.3.8. Login and Logout

Login and Logout pages will be done using the namesake REST API calls.
The SessionId received after the login API call should be stored in the localStorage.
The Access-tokens received after the OAuth login to social networks must also be stored on th device in localStorage.
For logout, one or all of these values must be removed from localStorage.

2.4. Sharing by email

We will use the Cordova plugin email composer for sharing by email feature.
https://github.com/katzer/cordova-plugin-email-composer
We will use the window.plugin.email.open method with the article information in the body and the subject, like so:

```
window.plugin.email.open({
   subject: article.title,
   body:  article.link
});
```

We must check if email service is available and if not then show modal error popup.

```
window.plugin.email.isServiceAvailable(
   function (isAvailable) {
      // alert('Service is not available') unless isAvailable;
   }
);
```

2.5. Other Requirements

See ADS section 2 for cross cutting requirements like logging, configuration, exception handling, internationalization etc.

2.6. Submission Deliverables

- Source Code
- Deployment Guide to verify the submission.

2.7. Technology Overview

2.7.1. Application Technologies

- HTML5
- PhoneGap 3
- BootStrap 3.1
- jQuery 2.0
- XHR
- JavaScript
- CSS
- localstorage
- Web SQL

3. Project Dependencies

3.1. Assemblies

Flipboard-Like HTML5 App – REST API Assembly

3.2. Components

None

3.3. Third Party Libraries

See section 1.4.1

4. Project Deliverable Details

4.1. Source code setup

Standard TopCoder Assembly Contest source code setup

4.2. Build Setup

Standard TopCoder Assembly Contest ant based build setup

5. Final Submission

- For each member, the final submission should be uploaded to the Online Review Tool.
- The final submission will be reviewed using the standard Online Review Assembly Scorecard.