

SUR+

Introduction

The goal of my project will be to fully test and show to my peers the skills I have acquired over the past months doing the HKUST Full Stack Web Development Specialization.

The project will be a website and Ionic App that will potentially allow users to put up advertisements to sell buy/sell surplus materials that end-users (companies and/or individuals) often have after a building project.

I tried to use everything that we learned in the Full Stack Web Development Course provided by HKUST in my project.

It used: HTML, CSS, JavaScript, JQuery, Bootstrap 4.0, AngularJS, NodeJS, MongoDB, RESTful API, Git, NPM, FontAwesome, Bower, AxureRP Prototyping

An expected list of features include:

- The ability to upload advertisements without the need to Log In and/or register.
- Search Engine that allows refining based on the types of materials.
- Custom advertisements with their own unique webpage including a description, uploadable pictures, telephone number, cost, unique advert reference number, links to social media, upload date, username, etc.-
- An Ionic mobile app that supports most if not all the web-based features.

On a more technical level, features would include:

- Ionic based mobile app (maybe not in the final release)
- built on Node.js and Angular.js
- Axure RP and Bootstrap for the website front-end
- Most likely be developed and hosted using IBM Bluemix.
- REST API and Mongo DB.
- Etc.

Design and Implementation

For me the trickiest part was converting the design I had made in the prototype into a well-coded website. This was because I thought the prototyping software I was using was also a sort of web design software that could be used as the template to my website. After finishing the beautiful prototype and moving onto the code itself, it quickly became apparent that Prototyping software does not allow anything more than showing a barebones design of the look and feel of the website. Hooking the client to the server on a prototype seemed near to impossible, so I quickly after gave up and started from scratch. This time going back to basics: Bootstrap. The rest was pretty straight-forward. I just used the ConFusion project as a

template for my website and go from there. A views, fonts, images, styles, and scripts folder essentially. From there, the project evolved.

My biggest challenge was time. Though I managed to finish it, there were various controllers I wished to implement into the site but could not due to having work and study to do aside from HKUST's FSWD course. Among which was integrating Ionic mobile application.

My project is the foundations to a major website which can easily include a suggestions section, featured section, Google maps integrated into the contact details section, most especially e-mail server integration, where users would be capable of editing their posts via a link that would be sent to their e-mail addresses; an admin control panel to review posts, increased methods of security & authentication, Google ads implementation, even users that can sign up if they so desire. Giving them more flexibility in managing the items long term.

I used as stated in the introduction, HTML, CSS, JavaScript, JQuery, Bootstrap 4.0, AngularJS, NodeJS, MongoDB, RESTful API, Git, NPM, FontAwesome, Bower, AxureRP prototyping fundamentally; then adding my own custom css, sass and less files to give a better and more modern looking bootstrap website.

As for modules, I have fundamentally two. One for users, and one for items assigned to users/email addresses. Though the user module was never used in the final version.

SUR+Items

Upload!

Upload an Item!

Item info...

Item Title

Title

Category

Admixture

Weight (kg.)

Dimensions (cm.)


Description

Description [Max 1000 characters]...

Price

Image URL

Image URL



Contact info...

Full Name



SUR+

Place offers on your surplus materials now! Quick and easy. No need to register or log in.

About us

SUR+ was founded in 2017 with the intention of providing people whom have surplus materials the opportunity to sell them without even needing to register a username.

We, Hornamental Co were founded also this year with SUR+ being our first successful development. A major thank you for being a part of our endeavours.



Location

13-L, 57th Street, Yougangsbridge,
09145, Hummermach,
Lougenburg.



Contact us via...

Telephone: +0800 101 101
Fax: +555 4954 4954
e-mail: sampletext@randomaddress.com



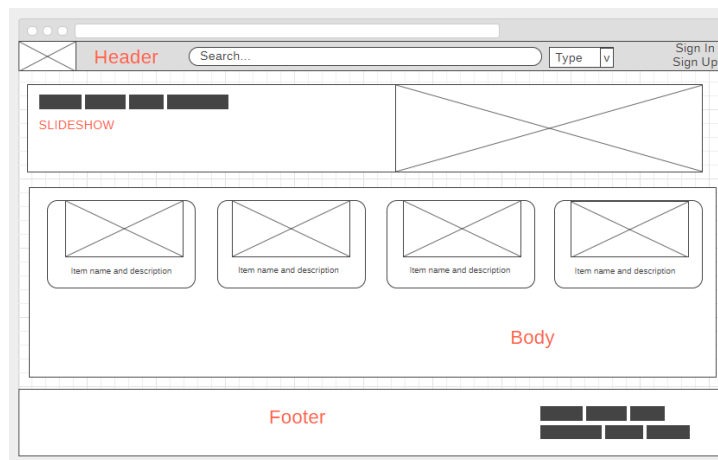
Front-end Architecture Design

The website will be fundamentally divided into the homepage, the user profile and registration [maybe, if implemented in the Project Implementation and Final Report] the listed item's dedicated webpage, the search results page, the create-an-item page, the header and the footer, etc. Each page will have 3 different views (small/mobile, medium/tablet and large/desktop) and 3 fundamental parts (header, body and footer).

I used a *wireframe.cc* to sketch the basic structure of *SUR+*. In case of not being able to use the links to the wireframed small, medium and large views of the index.html and the general structure of the UI, then I included some screenshots down below, which have also been hyperlinked:

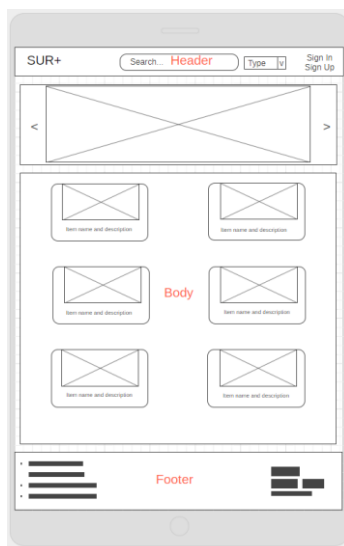
Desktop view

<https://wireframe.cc/4iRuh6>



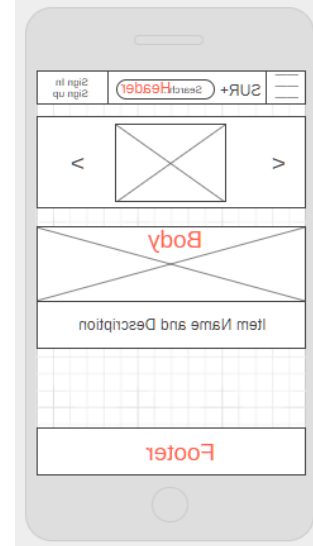
Tablet view

<https://wireframe.cc/SR7iHz>



Mobile view

<https://wireframe.cc/ndiV1X>



As said before, the intention of the website is to allow users to upload advertisements of their surplus building materials in the easiest and fastest way possible.

The homepage (or landing page) will be the index.html. From there users will be able to either log in/sign up, directly via the login/register form [again, if implemented in the final version], input and search for listed and refined items, select to create an item, view the “featured items”, and some randomly generated items under “Suggested items”, etc.

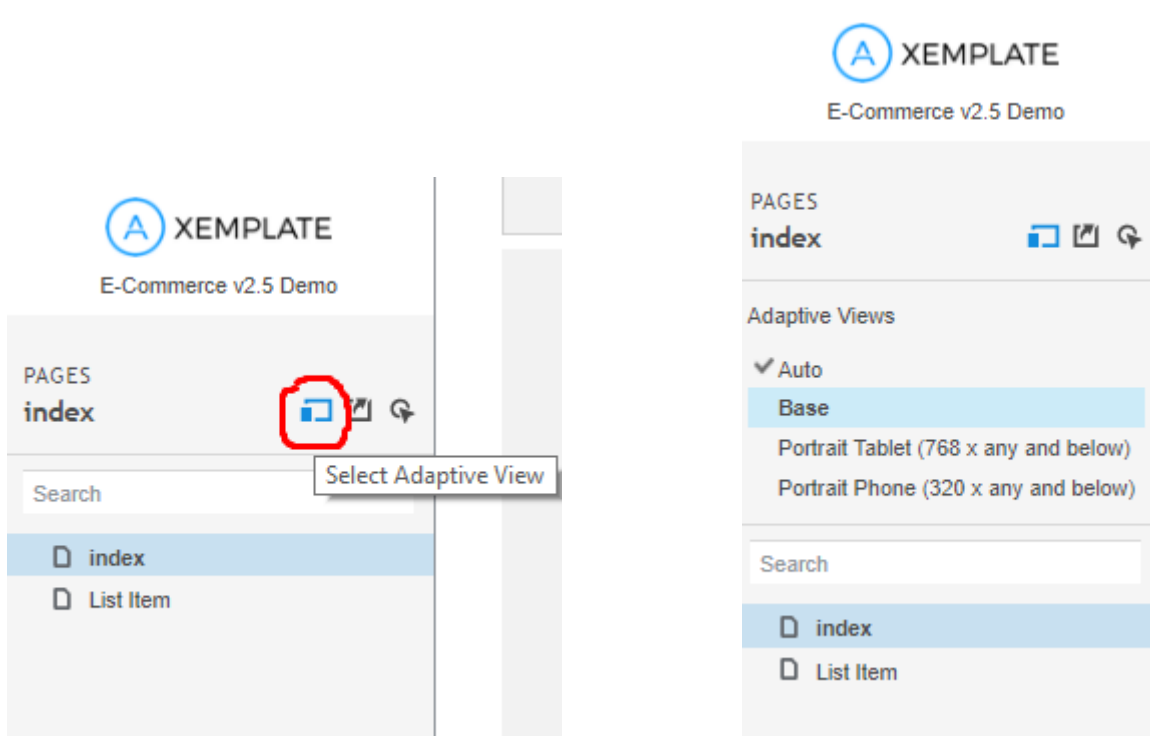
The website prototype has been made using *Axure* and a live preview of it is available here:

<http://3es50a.axshare.com/#g=1&p=index>

If asked for a password, input: **antidote123**

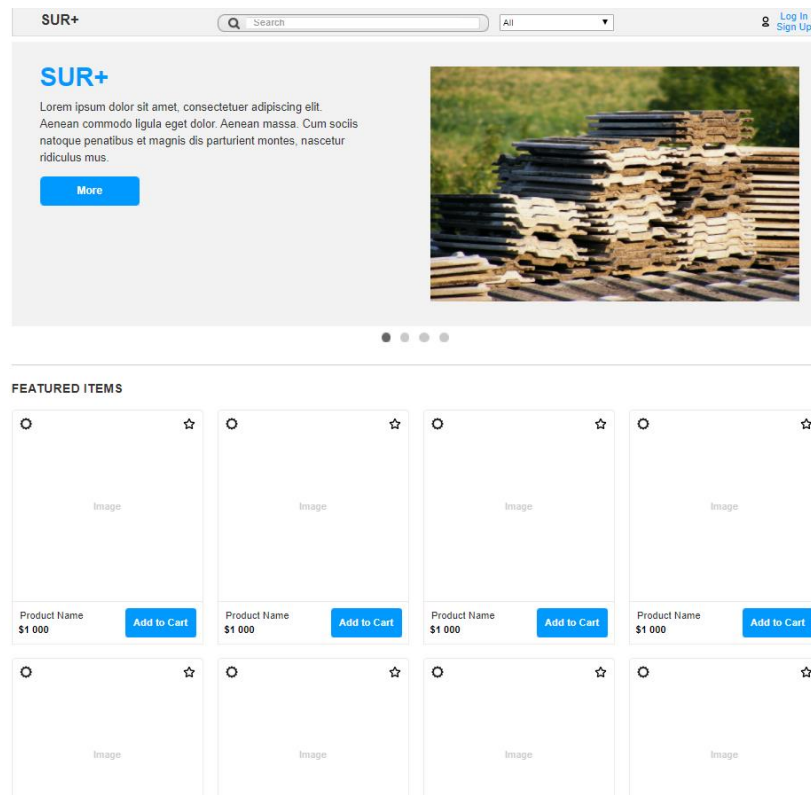
Select Adaptive View on the toolbar on the left (circled in red), and choose the view you wish to see.

Also select either “index” or “list item” to view the two currently prototyped webpages.

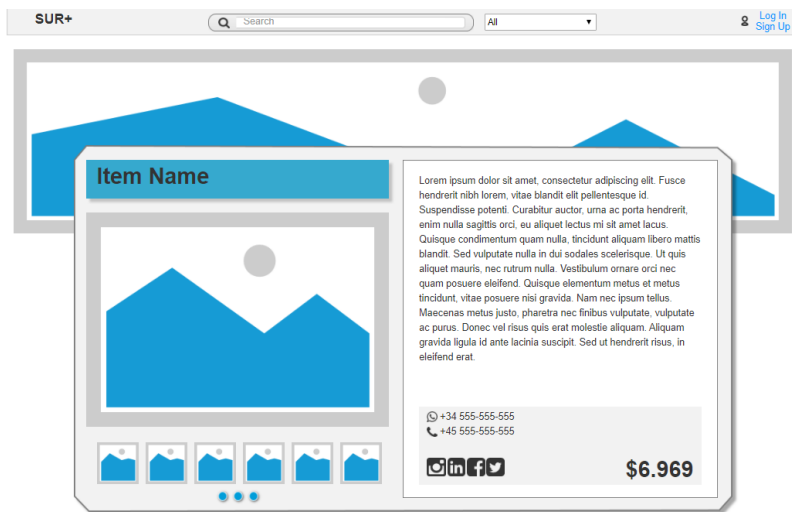


DESKTOP VIEW OF INDEX

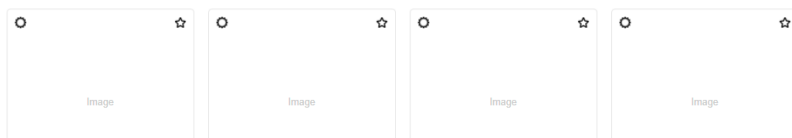
(OLD) FIRST PROTOTYPE



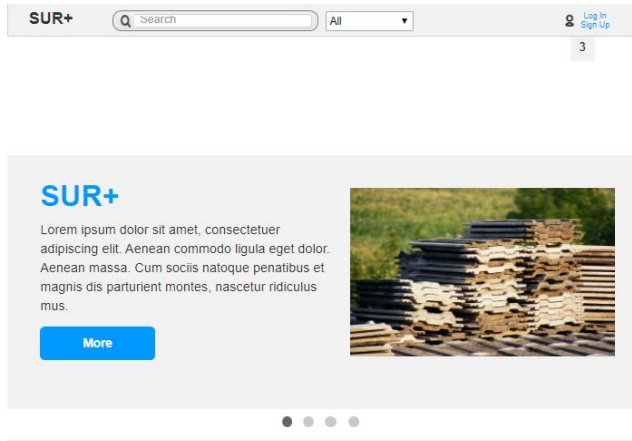
DESKTOP VIEW OF LIST ITEM



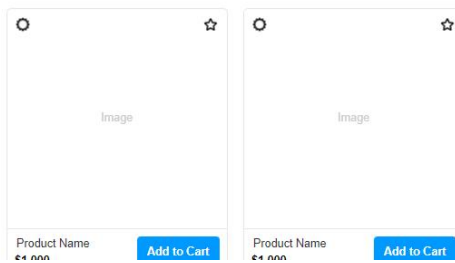
RELATED PRODUCTS



TABLET VIEW OF INDEX



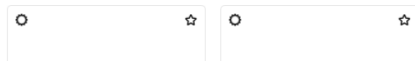
FEATURED ITEMS



TABLET VIEW OF LIST ITEM



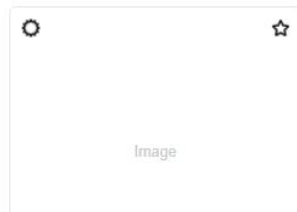
RELATED PRODUCTS



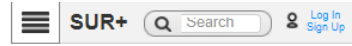
MOBILE VIEW OF INDEX



FEATURED ITEMS

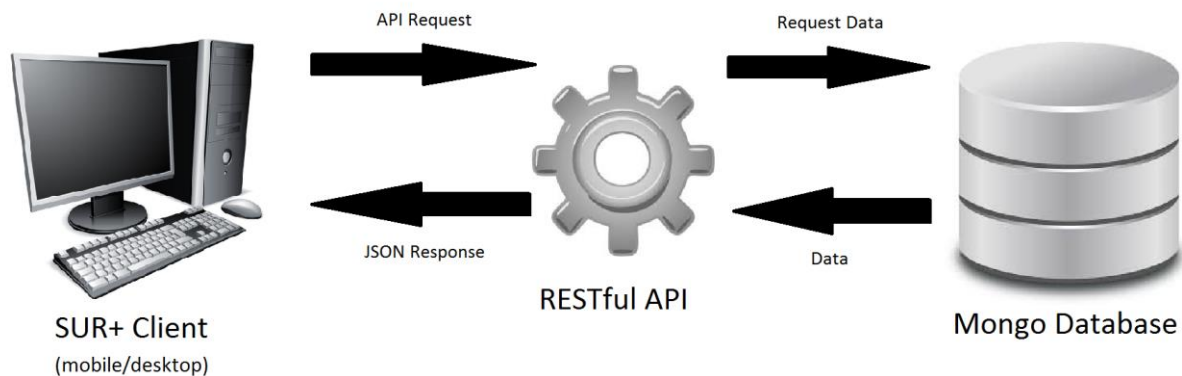


MOBILE VIEW OF LIST ITEM



Database Schemas, Design and Structure

The “items” title (product name), featured (if featured or not), category (the type of item it may be – helps with refined search), weight, dimensions X Y and Z, telephone number, mobile number, image_url and price.



Communication

Users will make a request to receive information from the Mongo Database. Requests can be made in the forms of GET (receive data), PUT (edit data), POST (send data) and DELETE (delete data). These requests will be made and sent to the Mongo Database via the REST Application Interface. Upon submitting a valid request, the database will submit a response with data (also through the REST API) to the client.

The SUR+ Mongo DB currently has been set up to have two collections being “users” and “items”. “users” would contain the list of SUR+’s registered users, each *user (collection item)* with its own unique Object ID, e-mail, username, password, picture URL and recovery question answer [As said before, though this has been set up, it may not be shown in the final report submission]. “items” on the other hands will include each one of the listed product that SUR+ users wish to advertise on the site. Each item will include a title, username assigned to the item (or the person’s name – depending on whether or not user registration is or not supported in the final version), item category it fits into, the X, Y and Z dimensions, the user’s telephone and mobile numbers, the image URL of the item and the list item price.

Depending on whether a user logs in or registers, a product item is being submitted to be advertised on SUR+, a list of items are being searched,.. The REST API will deliver a different response.

Conclusion

Doing this project allowed me to find awesome new languages, packages, etc. to work with. Opening more doors and possibilities in my pursuit of web development. It also allowed me figure out what I'm not doing so well at, so I can brush up as soon as this course is over. It gave me a general feel for independence, which was something I was needing as a young newcoming programmer.

I like the idea of developing very large and complicated websites. And that unfortunately comes with the drawback of having lots of things which can be improved. And like I said in the Design and Implementation section, there's a multitude of featured I wish to add as well as fix numerous vulnerabilities. In my case, the shortcoming are features, and the features are shortcomings, depending on how you look at them.

In hindsight, I'd definitely not have spent so much of my time and energy into a prototype which in the end was a total waste as far as the website goes. Although, on the bright side it allowed me to be a very good prototyper within a short period of time, and now I'll make I won't make these kinds of mistakes again.

References

<https://wireframe.cc> -> Wireframing the website was done using Wireframe.

<https://www.axure.com/> -> Prototyping the website was done using the Windows x64 RP 8 version of Axure

<https://share.axure.com/> -> Hosting the prototyped website.

<https://getbootstrap.com> -> Official Bootstrap website

<http://resteasy.jboss.org/> -> Official RestEasy website

<https://www.mongodb.com/> -> Official MongoDB website

<https://www.ibm.com/cloud-computing/bluemix/> -> IBM's Bluemix web hosting page

<https://wireframe.cc/4iRuh6> -> Desktop view

<https://wireframe.cc/SR7iHz> -> Tablet view

<https://wireframe.cc/ndIV1X> -> Mobile view

<http://ncdkc5.axshare.com/> -> Prototyped website

If asked for a password, input: **antidote123**

Introduction to the site and general information:

I tried to use everything that we learned in the Full Stack Web Development Course provided by HKUST in my project.

Used: HTML, CSS, JavaScript, JQuery, Bootstrap 4.0, AngularJS, NodeJS, MongoDB, RESTful API, Git, NPM, FontAwesome, Bower, AxureRP Prototyping

You may take a look at the early version of SUR+ and its prototype in \SUR+\Prototype SUR+. Since then major changes had been made as I began to realize that an exceedingly complicated procedure to make the front end of the finished product look like that, and I quite frankly hadn't the time to spend so long on it.

In Documentation help you'll find my list of handy commands I made for myself some time ago.

Setting up the website:

Find a suitable location for you to save the website. Once you've downloaded the repository, make sure you have NPM, bower and node globally installed. Then do git bash "bower install" as well as "npm install". This will make sure that any of the plugins I had used will be downloaded for you to use.

Make sure you have MongoDB installed, and add a database (db) by the name of "rest-commerce", and a collection to it named "items".

This will set up the Database. I suggest using Postman or RESTEasy to test and make sure it's working. Next add some random items to the "items" collection.

Hosting the site:

To do so, if not opened already, open mongod.exe followed by mongo.

Then from the /SUR+ folder, git bash the following: node app.

Then both client and server should be running.