# FullStack Spec – Bitcube Internship

## INTRODUCTION

### Purpose

The Client wants to develop a property listing website for property sales. Sellers will be able to list adverts, and buyers will be able to search for properties and contact sellers.

## COLLABORATION REQUIREMENTS

The developer will check work into a GitHub repository at the end of every workday. Work will be checked to make sure the project is progressing on time and to the required specifications. Please make sure you check in working code that compiles. Make sure to invite me to your project after your first chec-in.

## TECHNICAL REQUIREMENTS

- Front-end: Angular with TypeScript
- HTML and CSS – Use Bootstrap. If you can find a free template, even better.
- Backend - ASP.NET Core RESTful API.
- ORM – EF Core
- Persistence – SQL Database

### Browser Support

The website needs to support the following browsers:

- Chrome desktop browser only

## TEMPLATES:

I found two templates online that will work great as a starting point because security is already implemented, one for the API and one for the front-end.

**Back-end:**

The first template is the ASP.NET Core 3.1 API with JWT Authentication:

Tutorial: https://jasonwatmore.com/post/2019/10/11/aspnet-core-3-jwt-authentication-tutorial-with-example-api

Git Repo: https://github.com/cornflourblue/aspnet-core-3-jwt-authentication-api

Clone that repo to your local PC and use it as a starting point for the backend. IMPORTANT: He uses VS Code for the backend, I have no idea why. Use Visual Studio, not VS Code.

**Front-end:**

The front-end is an Angular 9 repo by the same author.
Tutorial: https://jasonwatmore.com/post/2020/04/19/angular-9-jwt-authentication-example-tutorial

Git Repo: https://github.com/cornflourblue/angular-9-jwt-authentication-example

Security is beyond the scope of what we're doing in the internship. These templates will help you get a very basic implementation going, which is enough for what we're doing. As a junior you won't (or shouldn't) be expected to implement security for a project.

# USER PERSONAS

## Site Administrator

The Site Administrator will be able to deal with user account maintenance and will have full control over the listing status of all adverts on the website.

## Unauthenticated Users

An unauthenticated user (UU) is a site user that is not logged on. (Whether the user has an account or not).

## Authenticated Users

An authenticated user (AU) is a user with an account that is logged on. An AU will have access to additional areas of the website not accessible to UA's.

# SOFTWARE REQUIREMENTS

## GENERAL CONSIDERATIONS

These general guidelines must be observed across the system:

- Dates will me in the format: dd-MM-yyyy (23-05-2018)
- Any date input must have the option to select from a calendar. (Note: Dates is difficult, use the Bootstrap date picker and get to know it well).
- Email addresses should always be validated for an @ symbol. No other email validation in necessary.
- All delete actions must have a confirmation message before deletion.
- After each API action (save, delete) an Alert should be shown to the user indicating success or failure.

## OTHER CONSIDERATIONS

The project is broken up into stages. You will complete the stages in the order I set them out (1, then 2 etc). You must complete the front-end, API and SQL persistence of a stage before you continue to the next stage. The project is set up in a way that makes it difficult to complete in 1 month. But you can continue to work on it after you've left the internship program. The goal is to complete as much as you can at the highest quality that you're capable of.

It's a catch 22 that all developers face every day, speed, or quality. You need to find the balance between quality and speed, so you'll deliver as much as possible, but deliver a quality product that works, not a pile of crap. This is your first real test.

How much time are you going to put in? From 9am to 4pm? Or are you going to push hard for a month? What do professional developers do when they're faced with a deadline and more work than they can handle? Weekends? 16 hours days? I'll give you a tip, it's not from 9 to 4.

### GENERAL TIPS:

With each phase, start with the front-end and use the Angular In-Memory API. That way you can create fake data and only work on one layer at a time. Adding extra properties to classes is quick, you only refactor that one layer.

If you start with the DB and API, you'll find that if you forget properties on classes (which you will), you have to re-factor all the layers every time you make a change. This takes a lot more time. Get clarity of what's needed on the front-end first, and then move to the API and data-persistence layers. You must complete all layers for each phase before moving to the next phase though. See every phase as a complete project that can be rolled out to production. So, in other words, start with the front-end for phase 1, then the API and DAL layers for phase 1. Then start the front-end for phase 2, etc.

The biggest mistake junior developers make is assuming that the website-user thinks like they do. Users are idiots at the best of times. They will try every trick in the book to bypass input validation. And nothing irritates a senior developer more than input validation that falls apart at the slightest deviance from the 'happy path'. Test your validation, try to break it, over and over. If a user can only enter digits, don't allow anything else. Check on the internet for Angular controls to help you. But, if you decide to use a 3rd party package, make sure they've got lots of downloads and are being actively developed. You don't want to get stuck with a dead control.

Give feedback to the user. If something is loading, don't just display a blank page. If something was added or updated or removed, show a message somewhere to tell the user. Nothing screams 'amateur' like silence after an action took place, or a blank loading screen.

This spec focusses on the front-end, because that's what the users and your client will see. How you deal with the API and data-persistence is up to you. You'll know what's needed once you've completed the front-end sections.

For design, look at other websites that already does what you're tasked to do. For instance, Property24. You'll get a lot of great ideas for layout and display from them. Or any other property listing website. Don't re-invent the wheel, copy and modify it.

# WEB FRONT-END

## General

I'll give general guidance in this document, but the developer is free to use his/her own discretion when it comes to page layout and component use etc.

## Layout Components

The site will have the following shared layout components like menu's and footers.

### Main Menu (1)

A main menu that will change depending on whether the user is authenticated or not.

- Home (**Home Page**)
- Homes for Sale (**Homes for Sale List Page**)
- Sell a house (**Redirects to Register page, you have to be registered to list a property**)
- Login (if not authenticated – will redirect to **Login Page**)
- Register (if not authenticated – will redirect to **Register Page**)

- Logout (if authenticated – will log the user out and return to **Home Page**)

### Sub-Menu (1)

This menu will only display for authenticated users. The menu will change depending on the user role. Don’t overdo this, you can just have a second menu below the top menu. Also note, a lot of these options will point to components that will only be developed in later phases of the project. So just have the menu there, but when you click on the link nothing happens.

Authenticated User

- **My Adverts** (DEFAULT: My Adverts Page)
- **My Profile** (Seller Profile Page)
- **Buy a House** (Homes for Sale List Page)
- **Sell a house** (Create/Edit Advert Page)
- **Manage my Account** (Manage My Account Page)
- **My Seller Profile** (Seller Profile Page)
- **Favourite Houses** (Favourites Page)

Administrator

- **User Management** (User Management Page)
- **Advert Management** (Advert Management Page)

### Footer (1)

This is just a placeholder to round off the page. You can add n list of menu options if you want to but don’t waste time here.

# Web Pages / Components

## Home Page (1)

### *Description*

The main landing page of the website – this can be anything. Don’t spend too much time here, you can update this at a later stage with a banner or whatever AFTER you’ve added the child components in stage 3 to 5.

For now, we just need a place where the user will land and see the menus. Once you’ve completed the ‘Homes for Sale’ page in phase 3 you can add that component on the Home page. And once you’ve completed the ‘Featured Homes List’ component in phase 5 you can replace the Homes for Sale‘ page with the ‘Featured Home List’. Same goes for the ‘Search’ component.

### *Home Page Components (These components are part of Phase 5, so for phase 1, just do the home page and menu’s)*

The home page will consist of the following sections:

- **Search Component (5)**: Search options will be described in detail in the ‘Search Component’ section.

- **Featured Homes list (5)**: This is a list of adverts from sellers who upgraded their adverts to ‘Featured listings’. Adverts should be displayed in a vertical list. For now, display all featured adverts, but later you’ll have to add paging. You won’t have time for that now though. The adverts will be retrieved with an API call. The advert information will be described in more detail in the ‘Featured Homes List Component’.

## Registration Page (1)

### *Description*

User registration page. You should know the deal by now. We use the email address as the username. It’s easier for users to remember their own email addresses that a random username.

### *Page Details*

If the user is already authenticated the user should be redirected to the ‘My Adverts Page’. If not, the registration page should be displayed.

Information to capture:

- **Forenames** string: required: min 1 chars, max 100 chars: Forenames of the user.
- **Surname** string: required: min 3 chars, max 100 chars: Surname of the user.
- **Email Address** string: required: min 6 chars, max 100 chars: Check for @ sign.
- **Password** string: required: min 8 chars, max 100 chars: No special characters needed
- **Password Confirm:** string: should match the first password.

Actions:

- **Register**: Validate the fields and submit the form object to the API.
- **Login**: A link to the login page should also be available

On success, redirect ‘My Adverts Page’.

## Login Page (1)

### *Description*

Basic login page.

### *Page Details*

If the user is NOT authenticated, this page will be displayed. If the user IS authenticated, the user should be redirected to the ‘My Adverts’ Page.

The user will enter an email address and password and click a submit button. Form validation should be performed on the client side. This will post the login details to the API.

- If the login is successful, a UserViewModel with a JWT will be returned from the API can be used in subsequent authorization requests (functionality already built in). The user should be redirected to the ‘My Adverts Page’ after successful login.
- If not successful a message will be returned from the API that needs to be displayed to the user.

Actions:

- **Login**: Validate the fields and submit the form object to the API. The API will return a UserViewModel if successfully authenticated, or an error message if the authentication was unsuccessful.
- **Register Here**: A link to the Registration page.

## My Adverts Page (2)

### *Description*

This is a list of adverts created by this user. The user obviously needs to be authenticated to access this page.

### *Page Details*

This page will have a list of adverts belonging to this user. The adverts should have the same display as any other advert list item, with the following changes:

- A 'Status' label for the advert state. (See Appendices for advert states)
- A button labelled 'Action…' with a sub-list of the following actions:
  - A 'Edit' link that will allow the seller to edit the advert
  - A 'Hide Advert' link that will change the advert status to HIDDEN and remove the advert from the live advert lists.
  - A 'Delete Advert' link that will allow the seller to delete the advert.

Actions:

- **Edit**: Redirect to 'Create/Edit Advert Page'. The current advert status will stay the same.
- **Hide Advert**: Only show this option if the advert does NOT have a status of HIDDEN) Change the advert status to 'HIDDEN' and update the advert.
- **Show Advert**: Only show this option if the advert has a status of HIDDEN. Change the advert status to 'LIVE' and update the advert.
- **Delete Advert**: Ask for confirmation by displaying the following message to the user: "Are you sure you want to delete this advert? This action cannot be undone, are you sure you want to continue?" If the user confirms this action, mark the advert status as 'DELETED' and update the advert and remove the advert from the list. We only shadow-delete adverts, so all that happens is a status change. See appendices for more details.

## Create/Edit Advert Page (2)

### *Description*

The user will be able to create a new advert or edit an existing advert on this page, IF authenticated.

### *Page Details*

If an existing advert is getting updated, populate all the relevant controls with the current advert details.

To be able to create a new advert the following information should be captured and submitted to the API:

- **Advert Headline Text**: string: required: min 10 chars, max 100 chars.
- **Province**: select: required: linked to Provinces. Select control is not editable.
- **City**: select: required: linked to Cities for the selected province. Select control is not editable.
- **Advert Details**: string: required: min 10 chars, max 1000 characters.
- **Price**: number: minimum 10000, max 100000000.

Actions:

- **Publish Advert**: Change the advert status (see the appendix for more info about statuses) to LIVE and post/put the advert to the API. On success, redirect to My Adverts List Page. On failure, display message to user and stay on page.
- **Cancel**: Redirect to My Adverts List Page.

## Homes for Sale List Page (3)

### *Description*

A list of adverts displayed to the buyers that matches the search criteria selected (search criteria is part of Phase 5, so for now, display all adverts). This will be displayed whether the user is authenticated or not, it's basically the shopping page.

### *Page Components*

This page will consist of the following sections:

- The Search Component at the top (5)
- Underneath the search component, on the right a 'Order by' selection of either:
  - Price High to Low
  - Price Low to High
- A list of adverts meeting the search criteria: Each advert item in the list will display the:
  - Advert Headline Text
  - Location (a combination of province and city)
  - Price

Actions:

- If a user clicks on the Advert Headline Text, the user should be redirected to the Advert Details Page for the relevant advert.

Paging (You can add this if you have time after you've completed the project):

- A maximum of 10 adverts should be displayed per page. Pagination navigation should be provided.

## Advert Detail Page (3)

### *Description*

When you click on an advert in any advert list, you get taken to the advert's detail page. (Have a look at the Property24 website to see how they display adverts. This should give you some ideas on how to do this).

### *Page Details*

This page will consist of the following sections:

- Advert Details:
  - The photo placeholder image. We don't have time for photo management now, but later you can update the solution to include photos. Just use 1 standard image for now.
  - Advert Headline Text
  - Location (this is a combination of Province and City)
  - Price

  - Advert description
- The ‘Contact Seller Component’ (5) should be somewhere on the page. (See Contact Seller Component for more details)

## Manage My Account Page (4)

### *Description*

The user can change passwords, email addresses and other account details on this page.

### *Page Details*

The user should be able to change the following account details on this page:

- Password, by entering the existing password and the entering 2 new matching passwords (see Registration Page for password requirements)
- Email Address: Allow the user to change his/her email address (effectively the username).
- Forenames and Surname. (See Registration Page for requirements).

## Seller Profile Page (4)

### *Description*

A seller will be able to update the relevant information needed for buyers to get in contact with the seller on this page.

### *Page Details*

The Seller profile page will display relevant seller information depending on the role of the seller.

The following information will be available for editing. (Note: Always have this info in a form, so when the seller lands on this page, the info is already in input controls, ready to change and save).

| | |
|---|---|
| - **Email Address** | string: required: min 6 chars, max 100 chars: Check for @ sign. |
| - **Phone Number** | string: not required: min 6 chars, max 30 chars: No validation. |

<u>Actions:</u>

| | |
|---|---|
| - **Save Profile**: | Validate and save all the information |
| - **Cancel**: | Cancel all changes and redirects to the ‘My Adverts Page’ |

## Favourites Page (6)

### *Description*

An authorised user can have a list of favourite adverts. Favourites get added from the Contact Seller Component. This component will be displayed on the ‘Advert Details Page’.

### *Page Details*

A list of the user’s favourite adverts should be displayed. A favourite item should look the same as an advert list item. If the user clicks on an advert heading, the user should be redirected to the ‘Adverts Details Page’.

<u>Actions:</u>

**Remove**: Remove the advert from the list of favourite adverts.

## User Management Page (6)

### *Description*

**Site Administrator role only**:

Allows the site administrator to manage user accounts.

### *Page Details*

The page should consist of a list of all the user accounts in a table. The table should be searchable on 'Surname'. On search the table rows should filter to only display the relevant rows searched for. The following columns should be displayed:

- **Forenames**: The user's forenames
- **Surname**: The user's surname
- **Email Address**: The user's email address (username). If the email address is not validated, display an icon next to the email address to show the administrator that the user has not validated the email address.
- **Locked**: Display a 'lock' icon if the account is locked. Display nothing of the account is not locked.
- A dropdown list with the following **actions**:
  - **Unlock account**: (Only display of account is locked) Allows the administrator to unlock the user's account if locked.
  - **Change email address**: Allows the administrator to change the user's email address. When this is clicked, display an input box with the user's email address and a 'Save' button. When the Save button is clicked verify that there is text in the input box and that an '@' sign is present.

## Advert Management Page (6)

### *Description*

**Administrator role only**:

Allows the site administrator to manually change advert statuses.

### *Page Details*

The page will consist of the following sections:

- Searchable user list
- List of the selected user's adverts (in the normal advert list display format). Each advert will have a dropdown list with the advert statuses that will allow the administrator to change the advert status to any other status. Next to the status list should be a save button that will save the status change.

Actions:

- **Save**: Update the advert's status to the selected status and submit the advert to the API.

## Components

Component can be used in multiple pages across the website.

### Search Component (5)

#### *Description*

The search component will be part of the Home Page as well as the Homes for Sale List Page.

#### *Page Details*

A potential buyer should be able to search by the following criteria:

- **Province**: Dropdown list of provinces (see appendices for more information.)
- **City**: Dropdown list of cities, updates dynamically based on the selected province. (See appendices for more information.
- **Minimum Price**: Dropdown list of prices. See appendices for more information. (On max price change, should update to not be more than max price)
- **Maximum Price**: Dropdown list of prices. See appendices for more information. A list will be provided. (On min price change, should not be less than min price)
- **Keyword(s):** One or more keywords can be entered

Note 1: The select dropdowns should be read-only.

Validation:

A search can be submitted to the API with only a Province selected, which will return all the adverts from the API for the selected province.

Actions:

- **Search**: Submit a search object to the API. The API will return a list of relevant adverts and order it by Highest Price first. The result should always be displayed on the Advert List Page, even if the search was made from the Home Page.
- **Reset Search**: Reset all the controls to their default values.
-

### Contact Seller Component (5)

#### *Description*

This component should be displayed on the Advert Details Page. It will give both authorized and non-authorized users an opportunity to contact the Seller. A user does NOT have to have an account to contact a Seller.

#### *Page Details*

The following info should be displayed and captured in this component:

- Display the seller's first and last name
- Display the seller's phone number
- Display the seller's email address.
- Display a 'Add to Favourites' icon. If clicked, add this advert to the user's list of favourite adverts.
- Also display a contact form underneath the above detail with the following information:
  - **Your Name** string: required: min 5 chars, max 100 chars: Name of the user.
  - **Your Email** string: required: min 6 chars, max 100 chars: Check for @ only, no other validation necessary

- **Your Number** string: not required: min 0 chars, max 100 chars: Free text, no validation
- **Message** string: required: min 10 chars, max 2000 chars: Free text

Actions:

- **Send Message**: Validate the fields and just display a message 'Email Sent'. We won't integrate any email sending as it's beyond the scope of this project.
  - On validation failure, highlight failed input's and display failure alert.
  - On success, clear all inputs and display success alert.

### Featured Homes Advert Component (5)

*Description*

Seller's will have an option to upgrade an advert to a 'Featured' advert. Featured adverts will display on the home page and will be top of the list in searches. For now, everybody will obviously want to upgrade their adverts to 'Featured'. But in future we may implement a payment system for featured adverts.

This component will only display on the Home page though. Featured adverts in searches will display the same way as regular adverts but should be top of the list and should be marked as 'Featured'. Use a combination of features and normal adverts for test data.

*Page Details*

Adverts should display in a horizontal slider with a left and right arrow allowing the user to navigate the slider. NO auto navigation. The number of featured adverts on the screen will depend on the screen width of the browsing device. All featured adverts will be returned by the API, newest first. All these adverts should be in the slider.

The information to display in the advert should be the same as on the 'Homes for Sale List Page'.

Actions:

When a user clicks on the advert, redirect to the 'Advert Detail Page'.

# APPENDICES

## APPENDIX A: ADVERT STATES

- **HIDE** The advert is hidden, only the advert owner can see the advert.
- **LIVE** The advert is live and can be seen by potential buyers.
- **DELETED** The owner marked the advert as DELETED. A lot of times you'll 'shadow delete' data. You don't really delete it (for a lot of reasons), but from the system's point of view the data is deleted. Tricky thing to get right.

## APPENDIX B: PROVINCE AND CITY DATA

*These values should be returned by the API, NOT hard coded in the front end. That way you can easily update the data without changing the front-end. In the beginning you'll hard code it until your API is up and running, but then I want it in the API.*

**Provinces and Cities**

*Have a look on Property24 and select 5 provinces and 2 cities for each province. You can easily update this later with more provinces and cities.*